



И. А. Землякова

Нелинейное программирование в примерах и задачах

учебное пособие



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики, механики и компьютерных наук
им. И. И. Воровича

И. А. Землякова

**НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ
В ПРИМЕРАХ И ЗАДАЧАХ**

Учебное пособие

Ростов-на-Дону – Таганрог
Издательство Южного федерального университета
2024

УДК 519.853(075.8)

ББК 22.185.42я73

3-53

Печатается по решению кафедры методов оптимизации и машинного обучения Института математики, механики и компьютерных наук Южного федерального университета (протокол № 6 от 11 июня 2024 г.)

Рецензенты:

д-р техн. наук, профессор кафедры методов оптимизации и машинного обучения Института математики, механики и компьютерных наук Южного федерального университета *Г. И. Белявский*;

д-р физ.-мат. наук, доцент, заведующий кафедрой информатики и информационных таможенных технологий
Ростовского филиала Российской таможенной академии *О. Е. Кудрявцев*

Землякова, И. А.

3-53 Нелинейное программирование в примерах и задачах : учебное пособие / И. А. Землякова ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2024. – 126 с.

ISBN 978-5-9275-4738-8

Учебное пособие содержит систематическое изложение раздела «Нелинейное программирование», снабжённое теоретическим материалом, множеством примеров и заданий для самостоятельного закрепления материала. Целью данного пособия является подробное изучение такого раздела математического программирования, как нелинейное программирование.

Предназначено для студентов бакалавриата, обучающихся по направлению 01.03.02 Прикладная математика и информатика, студентов других специальностей, изучающих курс «Методы оптимизации и исследование операций».

УДК 519.853(075.8)

ББК 22.185.42я73

ISBN 978-5-9275-4738-8

© Южный федеральный университет, 2024

© Землякова И. А., 2024

© Оформление. Макет. Издательство

Южного федерального университета, 2024

Оглавление

1. Постановка задачи нелинейного программирования.....	4
2. Задача безусловной минимизации	5
3. Задача условной оптимизации. Графический метод решения.....	12
4. Выпуклые функции. Проверка функции на выпуклость	22
5. Задача выпуклого программирования	27
6. Задача нелинейного программирования с ограничениями- равенствами. Метод множителей Лагранжа	14
7. Задача нелинейного программирования с ограничениями- равенствами и неравенствами. Метод множителей Лагранжа.....	48
8. Численные методы оптимизации функции одной и нескольких переменных	62
8.1. Безусловная нелинейная оптимизация функции одной переменной	62
8.1.1. Метод перебора	62
8.1.2. Метод дихотомии (деления отрезка пополам)	64
8.1.3. Метод золотого сечения.....	66
8.2. Градиентные методы решения оптимизационных задач	70
8.2.1. Метод градиентного спуска	70
8.2.2. Метод Ньютона	76
8.2.3. Метод субградиентного спуска	79
8.2.4. Метод проксимального градиента.....	80
8.2.5. Метод стохастического градиентного спуска (SGD)	81
8.3. Численные методы решения оптимизационных задач, позволяющие не использовать градиент.....	83
8.3.1. Метод покоординатного спуска.....	84
8.3.2. Метод штрафных и барьерных функций.....	91
8.4. Эвристические алгоритмы решения оптимизационных задач	93
8.4.1. Генетические алгоритмы	94
8.4.2. Метод муравьиных колоний.....	99
8.4.3. Алгоритм имитации отжига	105
Список литературы.....	114
Приложение	117

1. Постановка задачи нелинейного программирования

Нелинейное программирование (НЛП) представляет собой раздел математического программирования. **Задача математического программирования** заключается в отыскании экстремальных значений функции (целевая функция, или функция цели) среди множества её допустимых значений, которое задаётся рядом ограничений.

Например:

$$\begin{aligned} f(x, y) &\rightarrow \max, \\ g(x, y) &\leq C, \\ x, y &\geq 0. \end{aligned}$$

В задаче нелинейного программирования (ЗНЛП) нелинейными являются целевая функция и (или) ограничения.

Задачи нелинейного программирования можно разделить на два больших класса: **задачи безусловной оптимизации** и **задачи условной оптимизации**. В задачах безусловной оптимизации происходит поиск минимума или максимума нелинейной целевой функции при отсутствии дополнительных ограничений. Задача условной оптимизации предполагает наличие дополнительных ограничений на переменные.

Решение задачи нелинейного программирования, в отличие от задачи линейного программирования, является более сложным процессом. Это связано с тем, что область допустимых решений задачи имеет сложную структуру и не обязана быть выпуклым множеством с конечным циклом угловых точек, как в задаче линейного программирования; оптимальное значение целевой функции может достигаться не только в угловой точке области допустимых решений, но и внутри данной области; возможно наличие нескольких локальных решений.

Определение 1. Вектор $x = (x_1, x_2, \dots, x_n)$, удовлетворяющий ограничениям задачи, называется **допустимым решением**.

Определение 2. Множество допустимых решений задачи называется **областью допустимых решений (ОДР)**.

Определение 3. Допустимое решение, доставляющее оптимальное значение целевой функции, называется **оптимальным решением** и обозначается x^* .

Решить задачу нелинейного программирования – это означает найти оптимальное решение x^* и оптимальное значение целевой функции, т. е. $f(x^*)$.

2. Задача безусловной минимизации

Если в задаче нелинейного программирования ограничения отсутствуют, то данная задача является задачей безусловной минимизации и может быть сформулирована в виде [8]:

$$f(x) \rightarrow \min, x \in S, S \subset \mathbb{R}^n,$$

где $f(x)$ – функция многих переменных.

Определение 4 [16]. Точка $x^* \in S$ называется **точкой глобального минимума** функции $f(x)$ на множестве S , если $f(x^*) \leq f(x)$ при всех $x \in S$.

Определение 5 [16]. Точка $x^* \in S$ называется **точкой строгого глобального минимума** функции $f(x)$ на множестве S , если $f(x^*) < f(x)$ при всех $x \in S, x \neq x^*$.

Определение 6 [16]. Точка $x^* \in S$ называется **точкой локального минимума** функции $f(x)$ на множестве S , если $f(x^*) \leq f(x)$ при всех $x \in S \cap V_\varepsilon(x^*)$, где $V_\varepsilon(x^*) = \{x \in \mathbb{R}^n: \|x - x^*\| \leq \varepsilon\}$ – шар радиуса $\varepsilon > 0$ с центром в точке x^* (ε – окрестность точки x^*).

Определение 7 [16]. Точка $x^* \in S$ называется **точкой строгого локального минимума** функции $f(x)$ на множестве S , если $f(x^*) < f(x)$ при всех $x \in S \cap V_\varepsilon(x^*)$, где $V_\varepsilon(x^*) = \{x \in \mathbb{R}^n: \|x - x^*\| \leq \varepsilon\}$ – шар радиуса $\varepsilon > 0$ с центром в точке x^* (ε – окрестность точки x^*).

Определение 8 [17]. Градиентом функции $f(x)$ называется вектор $f'(x)$, своим направлением указывающий направление наибольшего возрастания функции.

Определение 9 [17]. Матрицей Гессе (гессианом) $H(x)$ дважды непрерывно дифференцируемой в точке x функции $f(x)$ называется симметричная квадратная матрица, описывающая поведение функции во втором порядке:

$$H(x) = \left\{ \frac{\partial^2 f}{\partial x_i \partial x_j}(x) \right\}, i, j = 1, \dots, n;$$

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

Далее сформулируем некоторые теоремы, необходимые для решения задач безусловной оптимизации.

Теорема 1 (необходимое условие локального минимума) [16].

Пусть $f(x)$ — дифференцируемая в точке $x^* \in \mathbb{R}^n$ функция, если x^* — точка локального минимума, то $f'(x^*) = 0$.

Теорема 2 (достаточное условие локального минимума) [16].

Пусть $f(x)$ — дважды дифференцируемая в точке $x^* \in \mathbb{R}^n$ функция, если $f'(x^*) = 0$ и матрица Гессе $H(x)$ положительно определена, то x^* — точка локального минимума.

Напомним критерий положительной определённости матрицы.

Теорема 3 (критерий Сильвестра) [20]. Для того чтобы матрица была положительно определённой, необходимо и достаточно, чтобы все угловые миноры матрицы были положительны.

Определение 10 [43]. Угловым минором k -го порядка является определитель, составленный из элементов матрицы квадратичной формы, находящихся на пересечении первых k строк и k столбцов данной матрицы.

Таким образом, сформулируем алгоритм нахождения безусловного минимума.

Алгоритм нахождения безусловного минимума

1. Применить необходимое условие локального минимума: составить вектор $f'(x)$ первых производных функции $f(x)$ и решить систему $f'(x) = 0$. В результате получим вектор стационарных точек x^* .

2. Применить достаточное условие локального минимума: проверить, является ли матрица $H(x_i^*), i = 1, \dots, k$ положительно определённой, используя критерий Сильвестра. В случае положительного ответа сделать вывод о том, что x_i^* — точка локального минимума.

3. Если x_i^* не является точкой локального минимума, проверить, является ли эта точка точкой локального максимума. Для этого следует определить, является ли матрица $-H(x_i^*), i = 1, \dots, k$ положительно определённой, используя критерий Сильвестра. В случае положительного ответа сделать вывод о том, что x_i^* — точка локального максимума.

Рассмотрим примеры.

Пример 1. Найти локальный минимум функции

$$f(x) = x_1^3 - 2x_1x_2 + x_2^2 - 3x_1 - 2x_2.$$

Воспользуемся алгоритмом, описанным выше. Найдём частные производные первого порядка функции $f(x)$ по всем переменным:

$$\frac{\partial f(x)}{\partial x_1} = 3x_1^2 - 2x_2 - 3; \quad \frac{\partial f(x)}{\partial x_2} = -2x_1 + 2x_2 - 2.$$

Составим систему, приравняв найденные производные к нулю:

$$\begin{cases} 3x_1^2 - 2x_2 - 3 = 0, \\ -2x_1 + 2x_2 - 2 = 0. \end{cases}$$

Для решения данной системы прибавим к первому уравнению второе и решим полученное квадратное уравнение:

$$\begin{aligned} 3x_1^2 - 2x_1 - 5 &= 0, \\ x_{1,1} &= -1, \\ x_{1,2} &= \frac{5}{3}. \end{aligned}$$

Продолжим решение системы и для $x_{1,1} = -1$ получим $x_{2,1} = 0$. Для $x_{1,2} = \frac{5}{3}$ получим $x_{2,2} = \frac{8}{3}$.

Таким образом, получили две стационарные точки: $A(-1; 0)$; $B(\frac{5}{3}; \frac{8}{3})$.

Далее находим частные производные функции $f(x)$ второго порядка и составляем матрицу Гессе:

$$\begin{aligned} \frac{\partial^2 f(x)}{\partial x_1^2} &= 6x_1; \quad \frac{\partial^2 f(x)}{\partial x_1 x_2} = -2; \quad \frac{\partial^2 f(x)}{\partial x_2^2} = 2; \quad \frac{\partial^2 f(x)}{\partial x_2 x_1} = -2; \\ H(x) &= \begin{pmatrix} 6x_1 & -2 \\ -2 & 2 \end{pmatrix}. \end{aligned}$$

Рассмотрим стационарную точку $A(-1; 0)$ и составим $H(A) = \begin{pmatrix} -6 & -2 \\ -2 & 2 \end{pmatrix}$. Вычислим угловые миноры матрицы $H(A)$: $\Delta_1 = |a_{11}| = -6 < 0$; $\Delta_2 = |H(A)| = -16 < 0$. Положительной определённости матрицы нет, значит, точка A не является точкой локального минимума.

Проверим эту точку на локальный максимум. Составим $-H(A) = \begin{pmatrix} 6 & 2 \\ 2 & -2 \end{pmatrix}$. Вычислим угловые миноры матрицы $H(A)$: $\Delta_1 = |a_{11}| = 6 > 0$; $\Delta_2 = |H(A)| = -16 < 0$. Положительной определённости матрицы нет, значит, точка A не является точкой локального максимума. Следовательно, точка A — седловая точка.

Рассмотрим стационарную точку $B\left(\frac{5}{3}; \frac{8}{3}\right)$ и составим $H(B) = \begin{pmatrix} 10 & -2 \\ -2 & 2 \end{pmatrix}$. Вычислим угловые миноры матрицы $H(B)$: $\Delta_1 = |a_{11}| = 10 > 0$; $\Delta_2 = |H(A)| = 16 > 0$. Матрица положительно определённая, значит, точка B – точка локального минимума.

Ответ: $B\left(\frac{5}{3}; \frac{8}{3}\right)$ – точка локального минимума.

Проиллюстрируем полученный результат. Построим двумерные линии уровня функции $f(x)$ и отметим точки A и B . Построение будем выполнять в системе компьютерной алгебры Maple [11; 39]. Для этого сначала зададим функцию:

```
> f := (x1, x2) -> x1^3 - 2*x1*x2 + x2^2 - 3*x1 - 2*x2
```

Подключим пакет расширения *plots* и построим контурный график с помощью команды *contourplot*:

```
> with(plots):
```

```
> contourplot(f(x, y), x1 = -2.5 .. 2.5, x2 = -2.5 .. 5.0, axes = frame, filled = true, contours = 25, coloring = [yellow, green]);
```

Здесь опция *contours* задаёт количество линий уровня на поверхности. Первое значение цвета (*yellow*) соответствует наименьшему значению функции, второе (*green*) – наибольшему (рис. 1).

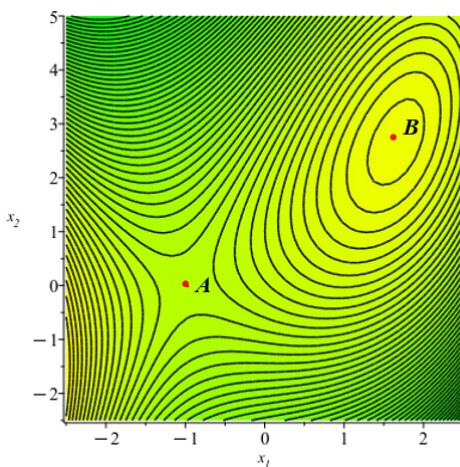


Рис. 1. Двумерные линии уровня функции $f(x)$

Построенные линии уровня подтверждают результат, полученный аналитически.

Пример 2. Найти локальный минимум функции

$$f(x) = x_1^4 + x_2^4 - (x_1 + x_2)^2.$$

Найдём частные производные первого порядка функции $f(x)$ по всем переменным:

$$\frac{\partial f(x)}{\partial x_1} = 4x_1^3 - 2(x_1 + x_2); \quad \frac{\partial f(x)}{\partial x_2} = 4x_2^3 - 2(x_1 + x_2).$$

Составим систему, приравняв найденные производные к нулю:

$$\begin{cases} 4x_1^3 - 2(x_1 + x_2) = 0, \\ 4x_2^3 - 2(x_1 + x_2) = 0; \\ 2x_1^3 - (x_1 + x_2) = 0, \\ 2x_2^3 - (x_1 + x_2) = 0. \end{cases}$$

Решим данную систему и получим стационарные точки $A(0; 0); B(1; 1); C(-1; -1)$.

Далее найдём частные производные функции $f(x)$ второго порядка и составим матрицу Гессе:

$$\frac{\partial^2 f(x)}{\partial x_1^2} = 12x_1^2 - 2; \quad \frac{\partial^2 f(x)}{\partial x_1 x_2} = -2; \quad \frac{\partial^2 f(x)}{\partial x_2^2} = 12x_2^2 - 2; \quad \frac{\partial^2 f(x)}{\partial x_2 x_1} = -2;$$

$$H(x) = \begin{pmatrix} 12x_1^2 - 2 & -2 \\ -2 & 12x_2^2 - 2 \end{pmatrix}.$$

Рассмотрим стационарную точку $A(0; 0)$ и составим $H(A) = \begin{pmatrix} -2 & -2 \\ -2 & -2 \end{pmatrix}$. Вычислим угловые миноры матрицы $H(A)$: $\Delta_1 = |a_{11}| = -2 < 0$; $\Delta_2 = |H(A)| = 0$. Положительной определённости матрицы нет, значит, точка A не является точкой локального минимума.

Проверим эту точку на локальный максимум. Составим $-H(A) = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$. Вычислим угловые миноры матрицы $H(A)$: $\Delta_1 = |a_{11}| = 2 > 0$; $\Delta_2 = |H(A)| = 0$. Положительной определённости матрицы нет, значит, точка A не является точкой локального максимума. Следовательно, точка A — седловая точка.

Рассмотрим стационарную точку $B(1; 1)$ и составим $H(B) = \begin{pmatrix} 10 & -2 \\ -2 & 10 \end{pmatrix}$. Вычислим угловые миноры матрицы $H(B)$: $\Delta_1 = |a_{11}| = 10 > 0$; $\Delta_2 = |H(A)| = 96 > 0$. Матрица положительно определённая, значит, точка B — точка локального минимума.

Рассмотрим стационарную точку $C(-1; -1)$ и составим $H(C) = \begin{pmatrix} 10 & -2 \\ -2 & 10 \end{pmatrix}$. Матрица положительно определённая, значит, точка C – точка локального минимума.

Ответ: $B(1; 1), C(-1; -1)$ – точки локального минимума.

Проиллюстрируем полученный результат. Построим двумерные линии уровня функции $f(x)$ и отметим точки A, B и C . Построение будем выполнять в системе компьютерной алгебры Maple. Для этого сначала зададим функцию:

```
> f := (x1, x2) -> x1^4 + x2^4 - (x1 + x2)^2
```

Подключим пакет расширения *plots* и построим контурный график с помощью команды *contourplot*:

```
> with(plots):
```

```
> contourplot(f(x1, x2), x1 = -2.0 .. 2.0, x2 = -2.0 .. 2.0, axes = frame, filled = true, contours = 25, coloring = [yellow, green]);
```

Здесь опция *contours* задаёт количество линий уровня на поверхности. Первое значение цвета (*yellow*) соответствует наименьшему значению функции, второе (*green*) – наибольшему (рис. 2).

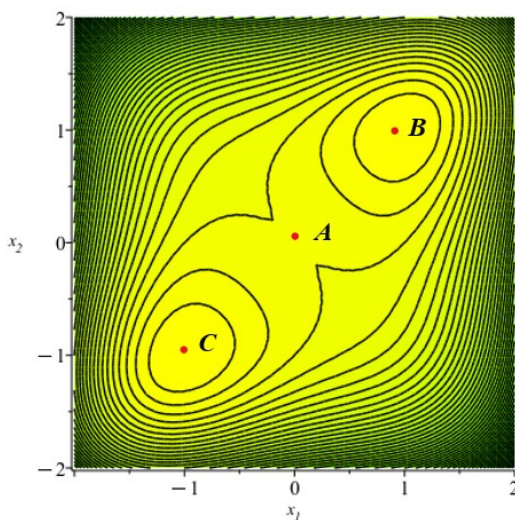


Рис. 2. Двумерные линии уровня функции $f(x)$

Построенные линии уровня подтверждают результат, полученный аналитически.

Задания для самостоятельного решения

Найти локальный минимум функции:

1. $f(x) = 4x_1^2 + 4x_2^2 - 4x_1x_2 - 12x_2$.

2. $f(x) = 2x_1^4 + x_2^4 - x_1^2 - 2x_2^2$.

3. $f(x) = x_1^2 - x_2^2 + ax_1 + bx_2$.

4. $f(x) = (x_1 - 4)^2 + (x_2 - 1)^2$.

5. $f(x) = x_1^2 - 2x_2^2$.

6. $f(x) = x_1^2 - 2x_2^2 + x_3^2$.

7. $f(x) = (x_1 - 4)^2 + (x_2 - 1)^2 + (x_3 - 1)^2$.

3. Задача условной оптимизации. Графический метод решения

Рассмотрим задачу нелинейного программирования в общем виде:

$$\begin{aligned} f(x) &\rightarrow \min (\max), \\ g_i(x) &\leq 0, i = 1, \dots, m, \\ h_i(x) &= 0, i = 1, \dots, k. \end{aligned}$$

Если вектор переменных $x = (x_1, x_2)$ состоит всего из двух компонент, то данная задача может быть решена графически.

При этом рассмотрим два типа задач, разрешимых графически: задача нелинейного программирования с линейной целевой функцией и задача с нелинейной целевой функцией.

При решении задач первого типа будем использовать **градиент** целевой функции. Пусть целевая функция имеет вид $f(x) = cx_1 + cx_2 \rightarrow \min(\max)$. Частные производные целевой функции $c_1 = \frac{\partial f(x)}{\partial x_1}$ и $c_2 = \frac{\partial f(x)}{\partial x_2}$ указывают скорость возрастания функции вдоль координатных осей. Соответственно, вектор, координаты которого представляют собой данные частные производные, будет указывать направление возрастания функции. Этот вектор назовём градиентом целевой функции и обозначим его $\bar{g} = \{c_1; c_2\}$. Таким образом, для нахождения максимума целевой функции следует идти в направлении градиента; для нахождения минимума – в направлении, обратном градиенту, т. е. в направлении антиградиента $-\bar{g} = \{-c_1; -c_2\}$. Строим градиент следующим образом: начало вектора совпадает с началом координат, конец вектора находится в точке с координатой $(c_1; c_2)$. Строим линии уровня – прямые, перпендикулярные градиенту. Если идти в направлении градиента, перемещая линию уровня параллельно самой себе, значение целевой функции будет увеличиваться. Перемещение линии уровня в направлении антиградиента приведёт к уменьшению значения целевой функции. Оптимальное значение целевой функции будет достигаться в крайних точках касания линии уровня и ОДР.

При решении задач второго типа будем использовать **линии уровня** нелинейной функции двух переменных.

Определение 11. Линией уровня функции двух переменных $f(x_1, x_2)$ называется множество точек на плоскости, таких, что во всех этих точках значение функции постоянно и равно некоторой константе C .

Таким образом, минимальное значение целевой функции достигается в первой точке касания линий уровня целевой функции $f(x)$ и области допустимых решений.

Рассмотрим примеры.

Пример 3. Найти наибольшее и наименьшее значения функции $f(x) = x_1^2 + x_2^2 - 4x_2$ при ограничениях:

$$\begin{cases} x_1^2 + x_2^2 \leq 9, \\ x_1^2 + 4x_1 + x_2^2 + 4x_2 \leq 5, \\ x_1 - x_2 = 1. \end{cases}$$

Построим область допустимых решений. Рассмотрим первое ограничение $x_1^2 + x_2^2 \leq 9$, которое задаёт внутреннюю область окружности с центром в точке $(0; 0)$ радиуса 3. Построим данную область в системе компьютерной алгебры Maple. Для этого сначала зададим функцию:

```
> f := x1^2 + x2^2
```

Подключим пакет расширения *plots* и построим область с помощью команды *implicitplot*:

```
> with(plots):
```

```
> q1 := implicitplot(f(x1, x2) <= 9, x1 = -7 .. 7, x2 = -7 .. 7, axes = frame,
coloring = ["SteelBlue"], filled = true, scaling = constrained);
```

Выбрали цвет *SteelBlue* для заливки области (рис. 3).

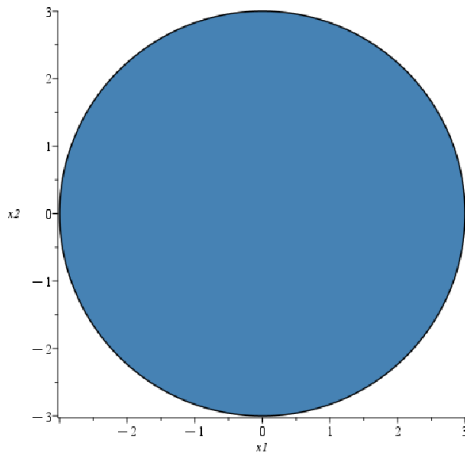


Рис. 3. Графическое изображение первого ограничения системы

Для графического изображения второго ограничения системы преобразуем его, выделив полные квадраты, и сведём к уравнению окружности.

$$(x_1^2 + 4x_1 + 4) + (x_2^2 + 4x_2 + 4) \leq 5 + 4 + 4,$$

$$(x_1^2 + 2)^2 + (x_2^2 + 2)^2 \leq 13.$$

Получили внутреннюю область окружности с центром в точке $(-2; -2)$ радиуса $\sqrt{13}$. Построим данную область в системе компьютерной алгебры Maple.

Первая функция уже задана. Добавим к ней вторую функцию g :

```
> f := x1^2 + x2^2
```

```
> g := x1^2 + x2^2 + 4*x1 + 4*x2
```

Подключим пакет расширения *plots* и зададим области с помощью команды *implicitplot*:

```
> with(plots):
```

```
> q1 := implicitplot(f(x1, x2) <= 9, x1 = -7 .. 7, x2 = -7 .. 7, axes = frame,
coloring = ["SteelBlue"], filled = true, scaling = constrained);
```

```
> q2 := implicitplot(g(x1, x2) <= 5, x1 = -7 .. 7, x2 = -7 .. 7, coloring =
["DarkGreen"], filled = true, scaling = constrained);
```

Выбрали цвет *SteelBlue* для заливки первой области и *DarkGreen* для второй (рис. 4).

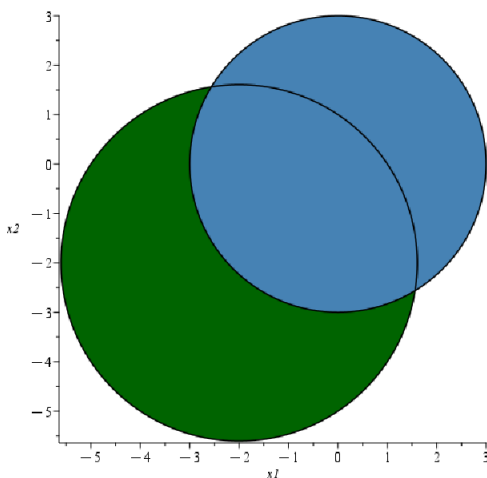


Рис. 4. Графическое изображение первых двух ограничений системы

Выводим две области в одной системе координат с помощью *display*:

```
> display(q1, q2)
```

Третье ограничение системы задаёт прямую $x_1 - x_2 = 1$. Построим её в системе компьютерной алгебры Maple.

Добавим третью функцию z к уже имеющимся:

```
> f := x1^2 + x2^2
```

```
> g := x1^2 + x2^2 + 4*x1 + 4*x2
```

```
> z := x1 - x2
```

Подключим пакет расширения *plots* и зададим области с помощью команды *implicitplot*:

```
> with(plots);
```

```
> q1 := implicitplot(f(x1, x2) <= 9, x1 = -7 .. 7, x2 = -7 .. 7, axes = frame,  
coloring = ["SteelBlue"], filled = true, scaling = constrained);
```

```
> q2 := implicitplot(g(x1, x2) <= 5, x1 = -7 .. 7, x2 = -7 .. 7, coloring =  
["DarkGreen"], filled = true, scaling = constrained);
```

```
> q3 := implicitplot(z(x1, x2) = 1, x1 = -7 .. 7, x2 = -7 .. 7, scaling = con-  
strained);
```

```
> display(q1, q2, q3);
```

Таким образом, область допустимых решений представляет собой множество точек, лежащих на выделенном красным отрезке прямой (рис. 5).

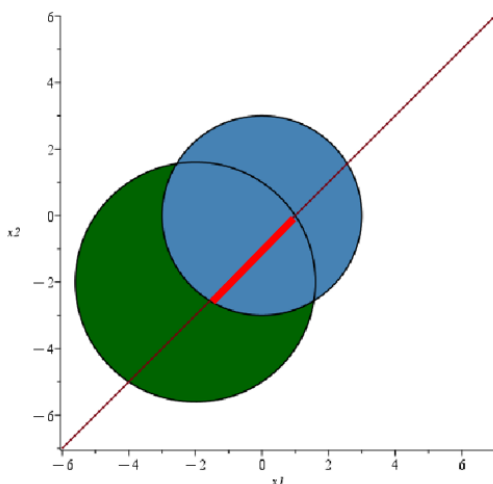


Рис. 5. Область допустимых решений

Перейдём к нахождению наибольшего и наименьшего значения целевой функции на построенной области. Так как целевая функция нелинейная, то изобразим её линии уровня, приравняв $f(x)$ некоторой константе C :

$$f(x) = x_1^2 + x_2^2 - 4x_2 = C.$$

Преобразуем целевую функцию, выделив полные квадраты:

$$x_1^2 + (x_2^2 - 4x_2 + 4) = C,$$

$$x_1^2 + (x_2 - 2)^2 = C.$$

Получили множество концентрических окружностей с центром в точке $(0; 2)$. Изобразим данное множество пунктиром. Как было отмечено ранее, точка минимума – это точка первого касания линий уровня и ОДР, т. е. точка A ; точка максимума – это точка последнего касания линий уровня и ОДР, т. е. точка B (рис. 6).

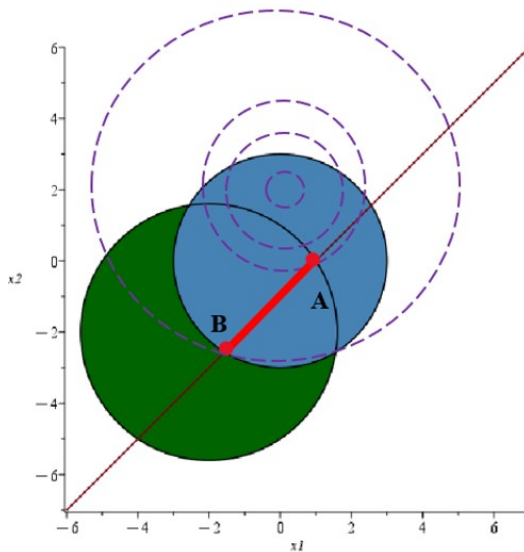


Рис. 6. Оптимальное решение

Определим координаты точки A как точку пересечения прямой $x_1 - x_2 = 1$ и окружности $(x_1^2 + 2)^2 + (x_2^2 + 2)^2 = 13$, решив систему:

$$\begin{cases} x_1 - x_2 = 1, \\ (x_1^2 + 2)^2 + (x_2^2 + 2)^2 = 13. \end{cases}$$

Получим $A(1; 0)$, $f(A) = 1$ – наименьшее значение функции при указанных ограничениях.

Определим координаты точки B как точку пересечения прямой $x_1 - x_2 = 1$ и окружности $x_1^2 + x_2^2 = 9$, решив систему:

$$\begin{cases} x_1 - x_2 = 1, \\ x_1^2 + x_2^2 = 9. \end{cases}$$

Получим $B\left(\frac{\sqrt{17}}{2} + \frac{1}{2}; \frac{\sqrt{17}}{2} - \frac{1}{2}\right)$, $f(B) = 11 - 2\sqrt{17}$ – наибольшее значение функции при указанных ограничениях.

Ответ: наименьшее значение функции при заданных ограничениях достигается в точке $A(1; 0)$ и равно 1; наибольшее значение функции достигается в точке $B\left(\frac{\sqrt{17}}{2} + \frac{1}{2}; \frac{\sqrt{17}}{2} - \frac{1}{2}\right)$ и равно $11 - 2\sqrt{17}$.

Пример 4. Найти наибольшее и наименьшее значения функции $f(x) = x_1^2 + x_2^2$ при ограничениях:

$$\begin{cases} x_1 x_2 \leq 4, \\ x_1 + x_2 \geq 5, \\ x_1 \leq 7, \\ x_2 \leq 6, \\ x \geq 0. \end{cases}$$

Ограничение на вектор $x \geq 0$ означает, что $x_1 \geq 0$; $x_2 \geq 0$.

Построим область допустимых решений (рис. 7). Построим данную область в системе компьютерной алгебры Maple.

Зададим функции:

```
> f := x1*x2;
> g := x1 + x2;
> z := x1;
> w := x2;
> q1:=implicitplot(f(x1, x2)<=4, x1 = 0 .. 8, x2 = 0 .. 7, scaling = constrained);
> q2:=implicitplot(5<=g(x1, x2),x1 = 0 .. 8, x2 = 0 .. 7, scaling = constrained);
> q3:=implicitplot(z(x1, x2)<=7, x1 = 0 .. 8, x2 = 0 .. 7, scaling = constrained);
> q3:=implicitplot(w(x1, x2)<=6,x1 = 0 .. 8, x2 = 0 .. 7, scaling = constrained);
> display(q1, q2, q3);
```

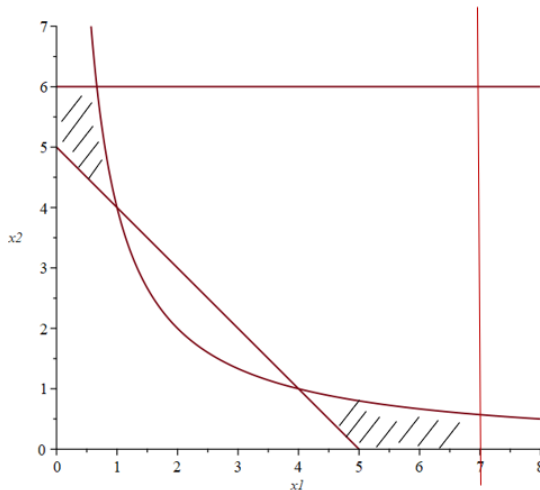


Рис. 7. Область допустимых решений

Изобразим линии уровня целевой функции, приравняв $f(x)$ некоторой константе C :

$$f(x) = x_1^2 + x_2^2 = C.$$

Получили множество концентрических окружностей с центром в точке $(0; 0)$. Изобразим данное множество пунктиром. Как было отмечено ранее, точка минимума – это точка первого касания линий уровня и ОДР, т. е. точки A и B ; точка максимума – это точка последнего касания линий уровня и ОДР, т. е. точка D или C (рис. 8).

Определим координаты данных точек. Получили $A(1; 4), B(4; 1); C\left(7; \frac{4}{7}\right); D\left(\frac{2}{3}; 6\right)$. $f(A) = f(B) = 17$ – наименьшее значение функции при указанных ограничениях. $f(C) = 49\frac{16}{49}; f(D) = 36\frac{4}{9}$. Таким образом, $f(C) = 49\frac{16}{49}$ – наибольшее значение функции при указанных ограничениях.

Ответ: наименьшее значение функции при заданных ограничениях достигается в точках $A(1; 4), B(4; 1)$ и равно 17; наибольшее значение функции достигается в точке $C\left(7; \frac{4}{7}\right)$ и равно $49\frac{16}{49}$.

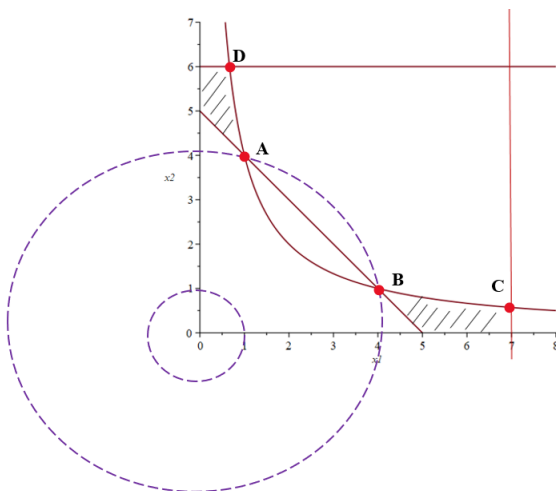


Рис. 8. Оптимальное решение

Пример 5. Найти наибольшее и наименьшее значения функции $f(x) = 2x_1 + x_2$ при ограничениях:

$$\begin{cases} x_1^2 + x_2^2 \leq 16, \\ x \geq 0. \end{cases}$$

Построим область допустимых решений (рис. 9). Построим данную область в системе компьютерной алгебры Maple.

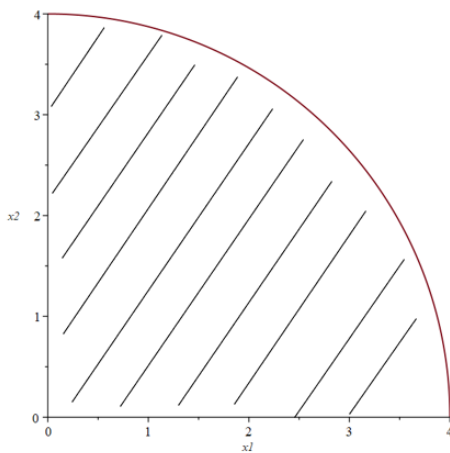


Рис. 9. Область допустимых решений

Зададим функцию:

$$> f := x_1^2 + x_2^2;$$

$> q1 := \text{implicitplot}(f(x_1, x_2) \leq 16, x_1 = 0 \dots 8, x_2 = 0 \dots 7, \text{scaling} = \text{constrained});$

Так как целевая функция линейна, то построим вектор-градиент целевой функции $\vec{g}\{2; 1\}$. Он указывает направление возрастания целевой функции. Крайнее касание градиента и области допустимых решений и задаёт точку, в которой функция принимает наибольшее значение. Тогда вектор, противоположный градиенту (антиградиент), $-\vec{g}\{-2; -1\}$ укажет направление убывания функции и позволит найти точку, в которой функция минимальна.

Таким образом, своё минимальное значение функция достигает в точке B , максимальное – в точке A (рис. 10).

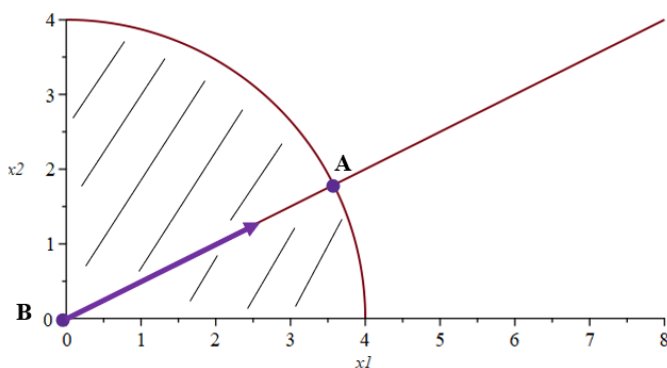


Рис. 10. Оптимальное решение

Определим координаты данных точек. Очевидно, что $B(0; 0)$ и наименьшее значение функции при указанных ограничениях $f(B) = 0$.

Определим координаты точки A как координаты точки пересечения окружности и прямой, на которой лежит градиент. Данная прямая проходит через точки с координатами $(0; 0)$ и $(2; 1)$. Составим её уравнение:

$$\frac{x_1 - 0}{2 - 0} = \frac{x_2 - 0}{1 - 0},$$

$$x_2 = \frac{1}{2}x_1.$$

Для нахождения координат точки A решим систему:

$$\begin{cases} x_1^2 + x_2^2 = 16, \\ x_2 = \frac{1}{2}x_1. \end{cases}$$

Получили $A\left(\frac{8\sqrt{5}}{5}; \frac{4\sqrt{5}}{5}\right)$. $f(A) = 4\sqrt{5}$ – наибольшее значение функции при указанных ограничениях.

Ответ: наименьшее значение функции при заданных ограничениях достигается в точке $B(0; 0)$ и равно 0; наибольшее значение функции достигается в точке $A\left(\frac{8\sqrt{5}}{5}; \frac{4\sqrt{5}}{5}\right)$ и равно $4\sqrt{5}$.

Задания для самостоятельного решения

Найти наибольшее и наименьшее значения функции $f(x)$ при указанных ограничениях:

1. $f(x) = (x_1 - 2)^2 + x_2^2$,

$$\begin{cases} x_1^2 - x_2 \leq 0, \\ -x_1 + x_2 \leq 1, \\ x \geq 0. \end{cases}$$
2. $f(x) = (x_1 - 4)^2 + (x_2 - 6)^2$,

$$\begin{cases} x_1 + x_2 \geq 1, \\ 2x_1 + 3x_2 \leq 12, \\ x \geq 0. \end{cases}$$
3. $f(x) = (x_1 + 2)^2 + (x_2 - 1)^2$,

$$\begin{cases} x_1 + x_2 \leq 1, \\ x_1 + x_2 \leq 7, \\ x \geq 0. \end{cases}$$
4. $f(x) = x_1^2 + x_2^2$,

$$\begin{cases} x_1x_2 \geq 4, \\ x_1 + x_2 \leq 7, \\ x \geq 0. \end{cases}$$
5. $f(x) = -2x_1 + x_2$,

$$\begin{cases} (x_1 - 4)^2 + x_2^2 \leq 4, \\ (x_1 - 4)^2 + (x_2 - 1)^2 \geq 1, \\ x \geq 0. \end{cases}$$

4. Выпуклые функции. Проверка функции на выпуклость

Задачи нелинейного программирования с выпуклыми функциями хорошо изучены и относятся к разделу выпуклого программирования, поэтому начнём с понятия выпуклой функции.

Определение 12 [6]. Функция $f: [a, b] \rightarrow R$ называется **выпуклой** на $[a, b] \subset R$, если $\forall x_1, x_2 \in [a, b]$ и $\forall \alpha_1, \alpha_2 \geq 0: \alpha_1 + \alpha_2 = 1$, таких, что $\alpha_1 + \alpha_2 = 1$, выполняется неравенство:

$$f(\alpha_1 x_1 + \alpha_2 x_2) \leq \alpha_1 f(x_1) + \alpha_2 f(x_2).$$

Если при $x_1 \neq x_2$ и $\alpha_1, \alpha_2 \neq 0$ это неравенство строгое, то функция $f(x)$ – **строго выпуклая** на $[a, b]$ [6].

Алгоритм проверки функции на выпуклость

1. Построить матрицу Гессе для рассматриваемой функции.
2. Вычислить угловые миноры матрицы Гессе. Если все угловые миноры строго больше нуля, то сделать вывод о строгой выпуклости функции.
3. Вычислить главные миноры матрицы Гессе. Если все главные миноры неотрицательны, то сделать вывод о выпуклости функции.

Определение углового минора было дано ранее (Определение 10), приведём определение главного минора.

Определение 13 [43]. Главным минором k -го порядка называется определитель, составленный из элементов матрицы квадратичной формы, находящихся на пересечении k строк и k столбцов данной матрицы с одинаковыми номерами. Рассмотрим в качестве примера матрицу размерности 3×3 :

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Главные миноры первого порядка: $\Delta_{11} = a_{11}, \Delta_{22} = a_{22}, \Delta_{33} = a_{33}$; второго порядка: $\Delta_{12} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \Delta_{13} = \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}, \Delta_{23} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}$; третьего порядка: $\Delta_{123} = |A|$.

Пример 6. Проверить функцию на выпуклость:

$$f(x) = x_1^2 + 2x_2^2 + x_1x_3 - 2x_2x_3.$$

Найдём частные производные первого порядка по всем переменным:

$$\frac{\partial f}{\partial x_1} = 2x_1 + x_3; \quad \frac{\partial f}{\partial x_2} = 4x_2 - 2x_3; \quad \frac{\partial f}{\partial x_3} = x_1 - 2x_2.$$

Далее определим частные производные второго порядка:

$$\frac{\partial^2 f}{\partial x_1^2} = 2; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_2 \partial x_1} = 0; \quad \frac{\partial^2 f}{\partial x_1 \partial x_3} = \frac{\partial^2 f}{\partial x_3 \partial x_1} = 1;$$

$$\frac{\partial^2 f}{\partial x_2^2} = 4; \quad \frac{\partial^2 f}{\partial x_2 \partial x_3} = \frac{\partial^2 f}{\partial x_3 \partial x_2} = -2; \quad \frac{\partial^2 f}{\partial x_3^2} = 0.$$

Составим матрицу Гессе:

$$H = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 4 & -2 \\ 1 & -2 & 0 \end{pmatrix}.$$

Угловые миноры матрицы Гессе: $\Delta_1 = 2 > 0$; $\Delta_2 = \begin{vmatrix} 2 & 0 \\ 0 & 4 \end{vmatrix} = 8 > 0$;

$$\Delta_3 = \begin{vmatrix} 2 & 0 & 1 \\ 0 & 4 & -2 \\ 1 & -2 & 0 \end{vmatrix} = -12 < 0. \text{ Так как } \Delta_3 < 0, \text{ то приходим к выводу,}$$

что функция не является выпуклой.

Ответ: функция не является выпуклой.

Пример 7. Определить значения параметров a, b, c , при которых функция $f(x) = ax_1^2 + 2bx_1x_2 + cx_2^2$ является выпуклой.

Функция является выпуклой, если, как было сказано ранее, главные миноры её матрицы Гессе неотрицательны.

Найдём частные производные первого порядка по всем переменным:

$$\frac{\partial f}{\partial x_1} = 2ax_1 + 2bx_2; \quad \frac{\partial f}{\partial x_2} = 2bx_1 + 2cx_2.$$

Далее определим частные производные второго порядка:

$$\frac{\partial^2 f}{\partial x_1^2} = 2a; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_2 \partial x_1} = 2b; \quad \frac{\partial^2 f}{\partial x_2^2} = 2c.$$

Составим матрицу Гессе:

$$H = \begin{pmatrix} 2a & 2b \\ 2b & 2c \end{pmatrix}.$$

Главные миноры матрицы Гессе:

$$\Delta_{11} = 2a; \quad \Delta_{22} = 2c; \quad \Delta_{12} = \begin{vmatrix} 2a & 2b \\ 2b & 2c \end{vmatrix} = 4ac - 4b^2.$$

Условие неотрицательности главных миноров приводит к системе неравенств:

$$\begin{cases} 2a \geq 0, \\ 2c \geq 0, \\ 4ac - 4b^2 \geq 0 \end{cases} \Rightarrow \begin{cases} a \geq 0, \\ c \geq 0, \\ ac \geq b^2. \end{cases}$$

Ответ: функция $f(x)$ является выпуклой при $\begin{cases} a \geq 0, \\ c \geq 0, \\ ac \geq b^2. \end{cases}$

Пример 8. Построить область выпуклости функции $f(x_1, x_2) = \frac{1}{3}x_1^3 - 4x_1x_2 + x_2^3$.

Найдём частные производные первого порядка по всем переменным:

$$\frac{\partial f}{\partial x_1} = x_1^2 + 4x_2; \quad \frac{\partial f}{\partial x_2} = 4x_1 + 3x_2^2.$$

Далее определим частные производные второго порядка:

$$\frac{\partial^2 f}{\partial x_1^2} = 2x_1; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_2 \partial x_1} = 4; \quad \frac{\partial^2 f}{\partial x_2^2} = 6x_2.$$

Составим матрицу Гессе:

$$H = \begin{pmatrix} 2x_1 & 4 \\ 4 & 6x_2 \end{pmatrix}.$$

Главные миноры матрицы Гессе:

$$\Delta_{11} = 2x_1; \quad \Delta_{22} = 6x_2; \quad \Delta_{12} = \begin{vmatrix} 2x_1 & 4 \\ 4 & 6x_2 \end{vmatrix} = 12x_1x_2 - 16.$$

Условие неотрицательности главных миноров приводит к системе неравенств:

$$\begin{cases} 2x_1 \geq 0, \\ 6x_2 \geq 0, \\ 12x_1x_2 - 16 \geq 0 \end{cases} \Rightarrow \begin{cases} x_1 \geq 0, \\ x_2 \geq 0, \\ x_2 \geq \frac{4}{3x_1}. \end{cases}$$

Изобразим полученную область (рис. 11).

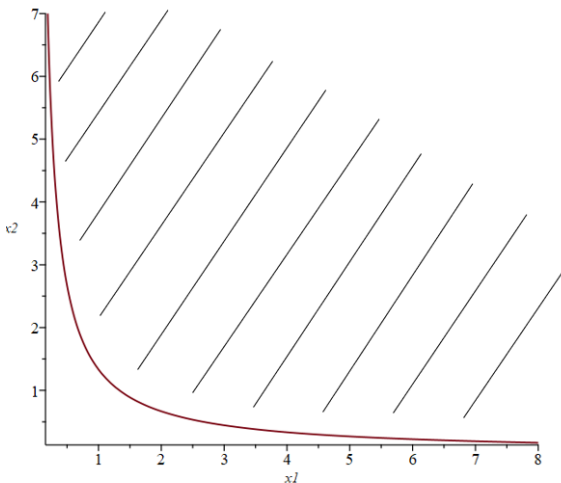


Рис. 11. Область выпуклости функции

Рассмотрим ещё один пример на построение области выпуклости функции.

Пример 9. Построить область выпуклости функции $f = x_1^2 - 2x_1^2x_2 + 3x_2^2$.

Найдём частные производные первого порядка по всем переменным:

$$\frac{\partial f}{\partial x_1} = 2x_1 - 4x_1x_2; \quad \frac{\partial f}{\partial x_2} = -2x_1^2 + 6x_2.$$

Далее определим частные производные второго порядка:

$$\frac{\partial^2 f}{\partial x_1^2} = 2 - 4x_2; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_2 \partial x_1} = -4x_1; \quad \frac{\partial^2 f}{\partial x_2^2} = 6.$$

Составим матрицу Гессе:

$$H = \begin{pmatrix} 2 - 4x_2 & -4x_1 \\ -4x_1 & 6 \end{pmatrix}.$$

Главные миноры матрицы Гессе:

$$\Delta_{11} = 2 - 4x_2; \quad \Delta_{22} = 6; \quad \Delta_{12} = \begin{vmatrix} 2 - 4x_2 & -4x_1 \\ -4x_1 & 6 \end{vmatrix} = 12 - 24x_2 - 16x_1^2.$$

Условие неотрицательности главных миноров приводит к системе неравенств:

$$\begin{cases} 2 - 4x_2 \geq 0, \\ 12 - 24x_2 - 16x_1^2 \geq 0 \end{cases} \Rightarrow \begin{cases} x_2 \leq \frac{1}{2}, \\ x_2 \leq -\frac{2}{3}x_1^2 + \frac{1}{2}. \end{cases}$$

Построим данную область в системе компьютерной алгебры Maple.

Зададим полученные ограничения (рис. 12):

```
> f := x2;
> g := x2 + 2/3*x1^2
> with(plots):
> q1 := implicitplot(f(x1, x2) <= 1/2, x1 = -8 .. 8, x2 = -8 .. 8, scaling =
constrained);
> q2 := implicitplot(g(x1, x2) <= 1/2, x1 = -8 .. 8, x2 = -8 .. 8, filled = true,
scaling = constrained);
> display(q1, q2);
```

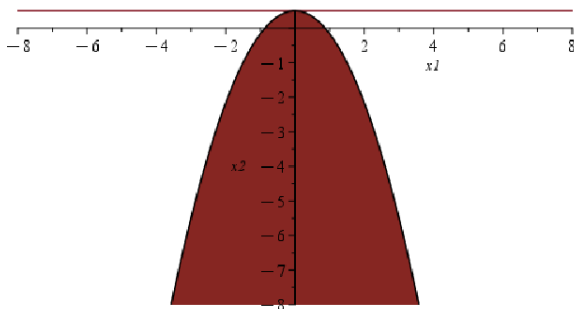


Рис. 12. Область выпуклости функции

Задания для самостоятельного решения

Построить область выпуклости функции:

1. $f(x) = 3x_1^2 + x_2^3 - x_3^3 + x_1x_3$.
2. $f(x) = x_1^2 - x_2^3 + 4$.
3. $x_1^3 - 2x_1^2x_2 + x_3^3$.

Определить значения параметров a, b, c , при которых функция является выпуклой:

1. $f(x) = ax_1^2 + bx_2^3 - cx_1x_2$.
2. $f(x) = ax_1^3 - bx_1x_2 + cx_2^3$.

5. Задача выпуклого программирования

Пусть функции $f(x)$ и $g_i(x)$, $i = 1, \dots, m$ выпуклые. Задачей выпуклого программирования называется задача вида

$$\begin{aligned} f(x) &\rightarrow \min, \\ \begin{cases} g_i(x) \leq 0, i = 1, \dots, m, \\ x \geq 0. \end{cases} \end{aligned}$$

Сформулируем ряд определений и теорем, необходимых для решения данной задачи [9].

Определение 14. Ограничение $g_i(x) \leq 0$, $i = 1, \dots, m$ называется **регулярным**, если найдётся допустимое решение задачи $\tilde{x}$, такое, что $g_i(\tilde{x}) < 0$, $i = 1, \dots, m$.

Определение 15. Задача выпуклого программирования называется **регулярной**, если все её ограничения регулярны.

Определение 16. **Функцией Лагранжа** задачи выпуклого программирования называется функция

$$L(x, \lambda) = f(x) + \sum_{i=1}^m g_i(x) \lambda_i,$$

где λ_i , $i = 1, \dots, m$ – множители Лагранжа.

Определение 17. Точка (x^*, λ^*) , $x^* \geq 0$, $\lambda^* \geq 0$ называется **седловой точкой** функции Лагранжа, если

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*), \forall x, \lambda \geq 0.$$

Теорема 4 (1-я теорема Куна – Таккера). Если (x^*, λ^*) – седловая точка функции Лагранжа, то x^* – оптимальное решение задачи выпуклого программирования.

Теорема 5 (2-я теорема Куна – Таккера). Для любого оптимального решения x^* регулярной задачи выпуклого программирования существует вектор множителей Лагранжа λ^* : (x^*, λ^*) – седловая точка функции Лагранжа.

Теорема 6. Пусть функции, входящие в задачу, выпуклые и непрерывно дифференцируемые. Точка (x^*, λ^*) , $x^* \geq 0$, $\lambda^* \geq 0$ является седловой точкой функции Лагранжа тогда и только тогда, когда в ней выполняются условия Куна – Таккера:

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &\geq 0, \\ x^* &\geq 0, \end{aligned}$$

$$(x^*, \nabla_x L(x^*, \lambda^*)) = 0,$$

$$\nabla_y L(x^*, \lambda^*) \leq 0,$$

$$\lambda^* \geq 0,$$

$$(\lambda^*, \nabla_\lambda L(x^*, \lambda^*)) = 0.$$

Сформулируем условия Куна – Таккера в удобном для решения задач виде.

Рассмотрим задачу выпуклого программирования вида

$$\begin{aligned} & f(x) \rightarrow \min, \\ & \begin{cases} g_i(x) \leq 0, i = 1, \dots, m_1, \\ g_i(x) = 0, i = m_1 + 1, \dots, m, \\ x_j \geq 0, j = 1, \dots, n_1, \\ x_j - \forall, j = n_1 + 1, \dots, n. \end{cases} \end{aligned}$$

Условия Куна – Таккера, соответствующие ограничениям задачи, в седловой точке имеют следующий вид (табл. 1).

Таблица 1

Условия Куна – Таккера в седловой точке

Ограничение задачи	Соответствующее ограничению условие Куна – Таккера
$g_i(x) \leq 0$	$\lambda_i^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_i} \leq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_i} \lambda_i^* = 0$
$g_i(x) = 0$	$\lambda_i^* - \forall; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_i} = 0$
$x_j \geq 0$	$x_j^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial x_j} \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial x_j} x_j^* = 0$
$x_j - \forall$	$x_j^* - \forall; \frac{\partial L(x^*, \lambda^*)}{\partial x_j} = 0$

Применим изложенный выше теоретический материал для проверки на оптимальность некоторой точки x^* в задаче выпуклого программирования.

Алгоритм проверки точки x^* на оптимальность в задаче выпуклого программирования

1. Преобразовать задачу выпуклого программирования к виду

$$f(x) \rightarrow \min,$$

$$\begin{cases} g_i(x) \leq 0, i = 1, \dots, m_1, \\ g_i(x) = 0, i = m_1 + 1, \dots, m, \\ x_j \geq 0, j = 1, \dots, n_1, \\ x_j - \forall, j = n_1 + 1, \dots, n. \end{cases}$$

2. Доказать выпуклость всех нелинейных функций.
3. Проверить регулярность относительно нелинейных ограничений.
4. Сформулировать условия Куна – Таккера в проверяемой точке x^* .
Получили систему уравнений и неравенств относительно λ^* .
5. Если полученная система разрешима и имеет решение λ^* , то из первой теоремы Куна – Таккера следует, что (x^*, λ^*) – седловая точка функции Лагранжа, а x^* – оптимальное решение задачи выпуклого программирования.
6. Если полученная система неразрешима, то из второй теоремы Куна – Таккера следует, что не найдётся вектор λ^* , такой, что (x^*, λ^*) – седловая точка функции Лагранжа, а это означает, что x^* не является оптимальным решением задачи выпуклого программирования.

Пример 10. Определить, при каких значениях параметра k точка $x^*(0,0)$ является оптимальным решением задачи:

$$f(x) = -x_1^2 - 4x_2^2 + 4x_1 + 16x_2 \rightarrow \max,$$

$$\begin{cases} x_1^2 + x_2^2 \leq k, \\ -x_1 + x_2 \geq 0, \\ x_2 \geq 0. \end{cases}$$

Воспользуемся описанным выше алгоритмом. Преобразуем задачу к необходимому виду. Для этого умножим целевую функцию на -1 для получения задачи минимизации. В первом ограничении перенесём параметр k в левую часть. Во втором ограничении умножим обе части на -1. Заметим, что по условию ограничение на переменную x_1 не задано, значит, полагаем, что $x_1 - \forall$. Получим задачу:

$$f(x) = x_1^2 + 4x_2^2 - 4x_1 - 16x_2 \rightarrow \min,$$

$$\begin{cases} x_1^2 + x_2^2 - k \leq 0, \\ x_1 - x_2 \leq 0, \\ x_1 - \forall, \\ x_2 \geq 0. \end{cases}$$

Докажем, что все нелинейные функции являются выпуклыми. Начнём с целевой функции $f(x) = x_1^2 + 4x_2^2 - 4x_1 - 16x_2$.

Найдём частные производные первого порядка по всем переменным:

$$\frac{\partial f}{\partial x_1} = 2x_1 - 4; \quad \frac{\partial f}{\partial x_2} = 8x_2 - 16.$$

Далее определим частные производные второго порядка:

$$\frac{\partial^2 f}{\partial x_1^2} = 2; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_2 \partial x_1} = 0; \quad \frac{\partial^2 f}{\partial x_2^2} = 8.$$

Составим матрицу Гессе:

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 8 \end{pmatrix}.$$

Главные миноры матрицы Гессе:

$$\Delta_{11} = 2 \geq 0; \quad \Delta_{22} = 8 \geq 0; \quad \Delta_{12} = 16 \geq 0.$$

Условие неотрицательности главных миноров выполнено, значит, функция $f(x)$ — выпуклая.

Далее рассмотрим первое ограничение системы $g_1(x) = x_1^2 + x_2^2 - k$.

Найдём частные производные первого порядка по всем переменным:

$$\frac{\partial g_1}{\partial x_1} = 2x_1; \quad \frac{\partial g_1}{\partial x_2} = 2x_2.$$

Далее определим частные производные второго порядка:

$$\frac{\partial^2 g_1}{\partial x_1^2} = 2; \quad \frac{\partial^2 g_1}{\partial x_1 \partial x_2} = \frac{\partial^2 g_1}{\partial x_2 \partial x_1} = 0; \quad \frac{\partial^2 g_1}{\partial x_2^2} = 2.$$

Составим матрицу Гессе:

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Главные миноры матрицы Гессе:

$$\Delta_{11} = 2 \geq 0; \quad \Delta_{22} = 2 \geq 0; \quad \Delta_{12} = 4 \geq 0.$$

Условие неотрицательности главных миноров выполнено, значит, функция $g_1(x)$ — выпуклая.

Далее проверим регулярность нелинейных ограничений. В рассматриваемой задаче одно нелинейное ограничение, которое удобно записать в виде $x_1^2 + x_2^2 \leq k$. Проанализируем возможные значения переменной k . Исходя из вида ограничения, делаем вывод, что $k \geq 0$.

При $k = 0$ система имеет единственное решение $(0; 0)$. Соответственно, это решение будет и оптимальным.

Продолжим анализ при $k > 0$. Понятно, что найдётся некоторое допустимое решение, при котором первое нелинейное ограничение выполняется как строгое неравенство. Таким образом, ограничение является регулярным.

Далее составим условия Куна – Таккера. Согласно табл. 1, для этого нам понадобится функция Лагранжа и её частные производные по всем переменным. Функция Лагранжа рассматриваемой задачи имеет вид

$$L(x, \lambda) = x_1^2 + 4x_2^2 - 4x_1 - 16x_2 + \lambda_1(x_1^2 + x_2^2 - k) + \lambda_2(x_1 - x_2),$$

где λ_1, λ_2 – множители Лагранжа.

$$\begin{aligned} \frac{\partial L(x, \lambda)}{\partial x_1} &= 2x_1 - 4 + 2x_1\lambda_1 + \lambda_2; \quad \frac{\partial L(x, \lambda)}{\partial x_2} = 8x_2 - 16 + 2x_2\lambda_1 - \lambda_2; \\ \frac{\partial L(x, \lambda)}{\partial \lambda_1} &= x_1^2 + x_2^2 - k; \quad \frac{\partial L(x, \lambda)}{\partial \lambda_2} = x_1 - x_2. \end{aligned}$$

Условия Куна – Таккера, соответствующие ограничениям задачи, имеют следующий вид (табл. 2).

Таблица 2

Условия Куна – Таккера для рассматриваемой задачи

Ограничение задачи	Соответствующее ограничению условие Куна – Таккера
$x_1^2 + x_2^2 - k \leq 0$	$\lambda_1^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_1} \leq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_1} \lambda_1^* = 0$
$x_1 - x_2 \leq 0$	$\lambda_2^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_2} \leq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_2} \lambda_2^* = 0$
$x_1 - \forall$	$x_1^* - \forall; \frac{\partial L(x^*, \lambda^*)}{\partial x_1} = 0$
$x_2 \geq 0$	$x_2^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial x_2} \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial x_2} x_2^* = 0$

Здесь $x^* = (0; 0)$ – проверяемая точка, т. е. $x_1^* = 0, x_2^* = 0$. Также следует вычислить значение частных производных функции Лагранжа в точке (x^*, λ^*) , где $x^* = (0; 0), \lambda^* = (\lambda_1^*, \lambda_2^*)$:

$$\frac{\partial L(x^*, \lambda^*)}{\partial x_1} = 2 \cdot 0 - 4 + 2 \cdot 0 \cdot \lambda_1^* + \lambda_2^* = -4 + \lambda_2^*;$$

$$\frac{\partial L(x^*, \lambda^*)}{\partial x_2} = 8 \cdot 0 - 16 + 2 \cdot 0 \cdot \lambda_1^* - \lambda_2^* = -16 - \lambda_2^*;$$

$$\frac{\partial L(x^*, \lambda^*)}{\partial \lambda_1} = 0 + 0 - k = -k; \quad \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_2} = 0 - 0 = 0.$$

Таким образом, условия Куна – Таккера, соответствующие ограничениям задачи в точке (x^*, λ^*) , имеют следующий вид (табл. 3).

Таблица 3

Условия Куна – Таккера для рассматриваемой задачи

Ограничение задачи	Соответствующее ограничению условие Куна – Таккера
$x_1^2 + x_2^2 - k \leq 0$	$\lambda_1^* \geq 0; -k \leq 0; -k\lambda_1^* = 0$
$x_1 - x_2 \leq 0$	$\lambda_2^* \geq 0; 0 \leq 0; 0 \cdot \lambda_2^* = 0$
$x_1 - \forall$	$0 - \forall; -4 + \lambda_2^* = 0$
$x_2 \geq 0$	$0 \geq 0; -16 - \lambda_2^* \geq 0; (-16 - \lambda_2^*) \cdot 0 = 0$

Во втором столбце таблицы образовалась система уравнений и неравенств:

$$\left\{ \begin{array}{l} \lambda_1^* \geq 0, \\ -k \leq 0, \\ -k\lambda_1^* = 0, \\ \lambda_2^* \geq 0, \\ -4 + \lambda_2^* = 0, \\ -16 - \lambda_2^* \geq 0. \end{array} \right.$$

Вспомним, что рассматривается случай $k > 0$. Решаем данную систему и приходим к выводу, что она противоречива, т. е. из второй теоремы Куна – Таккера следует, что не найдётся вектор λ^* , такой, что (x^*, λ^*) – седловая точка функции Лагранжа, а это означает, что x^* не является оптимальным решением задачи выпуклого программирования.

Ответ: точка $x^*(0,0)$ является оптимальным решением задачи при $k = 0$.

Рассмотрим ещё один пример.

Пример 11. Проверить, является ли точка $x^*(2,5; 1,5)$ оптимальным решением задачи:

$$f(x) = x_1^2 + x_2^2 - 2x_1 \rightarrow \min,$$

$$\begin{cases} x_1^2 + 4x_2^2 - 4x_1 - 4x_2 \leq 0, \\ x_1 + x_2 \geq 4, \\ x \geq 0. \end{cases}$$

Преобразуем задачу к необходимому виду:

$$\begin{aligned} f(x) &= x_1^2 + x_2^2 - 2x_1 \rightarrow \min, \\ \begin{cases} x_1^2 + 4x_2^2 - 4x_1 - 4x_2 \leq 0, \\ -x_1 - x_2 + 4 \leq 0, \\ x_1 \geq 0, \\ x_2 \geq 0. \end{cases} \end{aligned}$$

Докажем, что все нелинейные функции являются выпуклыми. Начнём с целевой функции $f(x) = x_1^2 + x_2^2 - 2x_1$.

Найдём частные производные первого порядка по всем переменным:

$$\frac{\partial f}{\partial x_1} = 2x_1 - 2; \quad \frac{\partial f}{\partial x_2} = 2x_2.$$

Далее определим частные производные второго порядка:

$$\frac{\partial^2 f}{\partial x_1^2} = 2; \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_2 \partial x_1} = 0; \quad \frac{\partial^2 f}{\partial x_2^2} = 2.$$

Составим матрицу Гессе:

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Главные миноры матрицы Гессе:

$$\Delta_{11} = 2 \geq 0; \quad \Delta_{22} = 2 \geq 0; \quad \Delta_{12} = 4 \geq 0.$$

Условие неотрицательности главных миноров выполнено, значит, функция $f(x)$ — выпуклая.

Далее рассмотрим первое ограничение системы $g_1(x) = x_1^2 + 4x_2^2 - 4x_1 - 4x_2$.

Найдём частные производные первого порядка по всем переменным:

$$\begin{aligned} \frac{\partial g_1}{\partial x_1} &= 2x_1 - 4; \\ \frac{\partial g_1}{\partial x_2} &= 8x_2 - 4. \end{aligned}$$

Далее определим частные производные второго порядка:

$$\frac{\partial^2 g_1}{\partial x_1^2} = 2;$$

$$\frac{\partial^2 g_1}{\partial x_1 \partial x_2} = \frac{\partial^2 g_1}{\partial x_2 \partial x_1} = 0;$$

$$\frac{\partial^2 g_1}{\partial x_2^2} = 8.$$

Составим матрицу Гессе:

$$H = \begin{pmatrix} 2 & 0 \\ 0 & 8 \end{pmatrix}.$$

Главные миноры матрицы Гессе:

$$\Delta_{11} = 2 \geq 0; \Delta_{22} = 8 \geq 0; \Delta_{12} = 16 \geq 0.$$

Условие неотрицательности главных миноров выполнено, значит, функция $g_1(x)$ – выпуклая.

Далее проверяем регулярность нелинейных ограничений. В рассматриваемой задаче первое ограничение нелинейно. Подберём некоторое допустимое решение задачи, которое обращает данное ограничение в строгое неравенство, – например, $\tilde{x}(3; 1)$. Таким образом, ограничение является регулярным.

Далее составим условия Куна – Таккера. Функция Лагранжа рассматриваемой задачи имеет вид

$$L(x, \lambda) = x_1^2 + x_2^2 - 2x_1 + \lambda_1(x_1^2 + 4x_2^2 - 4x_1 - 4x_2) + \lambda_2(-x_1 - x_2 + 4);$$

$$\frac{\partial L(x, \lambda)}{\partial x_1} = 2x_1 - 2 + 2x_1\lambda_1 - 4\lambda_1 - \lambda_2;$$

$$\frac{\partial L(x, \lambda)}{\partial x_2} = 2x_2 + 8x_2\lambda_1 - 4\lambda_1 - \lambda_2;$$

$$\frac{\partial L(x, \lambda)}{\partial \lambda_1} = x_1^2 + 4x_2^2 - 4x_1 - 4x_2; \quad \frac{\partial L(x, \lambda)}{\partial \lambda_2} = -x_1 - x_2 + 4.$$

Условия Куна – Таккера, соответствующие ограничениям задачи, имеют следующий вид (табл. 4).

Таблица 4

Условия Куна – Таккера для рассматриваемой задачи

Ограничение задачи	Соответствующее ограничению условие Куна – Таккера
$x_1^2 + 4x_2^2 - 4x_1 - 4x_2 \leq 0$	$\lambda_1^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_1} \leq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_1} \lambda_1^* = 0$
$-x_1 - x_2 + 4 \leq 0$	$\lambda_2^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_2} \leq 0; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_2} \lambda_2^* = 0$

Ограничение задачи	Соответствующее ограничению условие Куна – Таккера
$x_1 \geq 0$	$x_1^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial x_1} \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial x_1} x_1^* = 0$
$x_2 \geq 0$	$x_2^* \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial x_2} \geq 0; \frac{\partial L(x^*, \lambda^*)}{\partial x_2} x_2^* = 0$

Здесь $x^*(2,5; 1,5)$ – проверяемая точка, т. е. $x_1^* = 2,5, x_2^* = 1,5$. Также следует вычислить значение частных производных функции Лагранжа в точке (x^*, λ^*) , где $x^* = (2,5; 1,5), \lambda^* = (\lambda_1^*; \lambda_2^*)$:

$$\frac{\partial L(x^*, \lambda^*)}{\partial x_1} = 5 - 2 + 5\lambda_1^* - 4\lambda_1^* - \lambda_2^* = 3 + \lambda_1^* - \lambda_2^*;$$

$$\frac{\partial L(x^*, \lambda^*)}{\partial x_2} = 3 + 12\lambda_1^* - 4\lambda_1^* - \lambda_2^* = 3 + 8\lambda_1^* - \lambda_2^*;$$

$$\frac{\partial L(x^*, \lambda^*)}{\partial \lambda_1} = -0,75; \frac{\partial L(x^*, \lambda^*)}{\partial \lambda_2} = 0.$$

Таким образом, условия Куна – Таккера, соответствующие ограничениям задачи в точке (x^*, λ^*) , имеют следующий вид (табл. 5).

Таблица 5

Условия Куна – Таккера для рассматриваемой задачи

Ограничение задачи	Соответствующее ограничению условие Куна – Таккера
$x_1^2 + 4x_2^2 - 4x_1 - 4x_2 \leq 0$	$\lambda_1^* \geq 0; -0,75 \leq 0; -0,75\lambda_1^* = 0$
$-x_1 - x_2 + 4 \leq 0$	$\lambda_2^* \geq 0; 0 \leq 0; 0 \cdot \lambda_2^* = 0$
$x_1 \geq 0$	$2,5 \geq 0; 3 + \lambda_1^* - \lambda_2^* \geq 0; (3 + \lambda_1^* - \lambda_2^*) \cdot 2,5 = 0$
$x_2 \geq 0$	$1,5 \geq 0; 3 + 8\lambda_1^* - \lambda_2^* \geq 0; (3 + 8\lambda_1^* - \lambda_2^*) \cdot 1,5 = 0$

Во втором столбце таблицы образовалась система уравнений и неравенств:

$$\begin{cases} \lambda_1^* \geq 0, \\ -0,75\lambda_1^* = 0, \\ \lambda_2^* \geq 0, \\ 3 + \lambda_1^* - \lambda_2^* \geq 0, \\ (3 + \lambda_1^* - \lambda_2^*) \cdot 2,5 = 0, \\ 3 + 8\lambda_1^* - \lambda_2^* \geq 0, \\ (3 + 8\lambda_1^* - \lambda_2^*) \cdot 1,5 = 0. \end{cases}$$

Данная система имеет решение $\lambda_1^* = 0, \lambda_2^* = 3$. Значит, из первой теоремы Куна – Таккера следует, что (x^*, λ^*) – седловая точка функции Лагранжа, а $x^*(2,5; 1,5)$ – оптимальное решение задачи выпуклого программирования.

Ответ: точка $x^*(2,5; 1,5)$ является оптимальным решением задачи.

Задания для самостоятельного решения

1. Проверить, является ли точка $x^*(4; 0)$ оптимальным решением задачи:

$$\begin{aligned} f(x) &= 3x_1^2 + 4x_1x_2 + 5x_2^2 \rightarrow \min, \\ &\begin{cases} x_1 + x_2 \geq 4, \\ x_1 \geq 0, \\ x_2 \geq 0. \end{cases} \end{aligned}$$

2. Проверить, является ли точка $x^*(5; 1)$ оптимальным решением задачи:

$$\begin{aligned} f(x) &= 3x_1^2 + 4x_1x_2 + 5x_2^2 \rightarrow \min, \\ &\begin{cases} x_1 + x_2 \geq 4, \\ x_1 \geq 0, \\ x_2 \geq 0. \end{cases} \end{aligned}$$

3. Проверить, является ли точка $x^*\left(\frac{1}{\sqrt{2}}; 0; -\frac{1}{\sqrt{2}}\right)$ оптимальным решением задачи:

$$\begin{aligned} f(x) &= x_1 + x_3 \rightarrow \min, \\ &\begin{cases} x_1^2 + x_2^2 + x_3^2 - 1 \leq 0, \\ x_1 + x_2 - \frac{1}{\sqrt{2}} = 0, \\ x_2 \geq 0. \end{cases} \end{aligned}$$

4. Проверить, является ли точка $x^* \left(\frac{1}{\sqrt{2}}; 0; 0 \right)$ оптимальным решением задачи:

$$\begin{aligned} f(x) &= x_1 + x_3 \rightarrow \min, \\ \begin{cases} x_1^2 + x_2^2 + x_3^2 - 1 \leq 0, \\ x_1 + x_2 - \frac{1}{\sqrt{2}} = 0, \\ x_2 \geq 0. \end{cases} \end{aligned}$$

6. Задача нелинейного программирования с ограничениями-равенствами. Метод множителей Лагранжа

Рассмотрим задачу условной оптимизации с ограничениями-равенствами вида

$$\begin{cases} f(x) \rightarrow \max(\min), \\ g_i(x) = 0, i = 1, \dots, m, \end{cases} \quad (1)$$

где $f(x), g_i(x), i = 1, \dots, m$ – непрерывно дифференцируемые функции.

Здесь основная идея решения заключается в отыскании безусловного экстремума специально построенной функции Лагранжа. Далее используется классический метод решения задачи безусловной минимизации.

Функция Лагранжа имеет вид

$$L(x, \lambda_0, \dots, \lambda_m) = \lambda_0 f(x) + \sum_{i=1}^m g_i(x) \lambda_i,$$

где $\lambda_i, i = 0, \dots, m$ – множители Лагранжа.

Теорема 7 (необходимое условие локального экстремума первого порядка в задаче с ограничениями в виде равенств) [3; 12]. Пусть $f(x), g_i(x), i = 1, \dots, m$ – непрерывно дифференцируемые в окрестности точки x^* функции. Если точка x^* является локальным экстремумом задачи условной оптимизации с ограничениями-равенствами (1), то найдётся ненулевой набор множителей Лагранжа λ , такой, что для функции Лагранжа выполняется условие стационарности по x :

$$L'_x(x^*, \lambda) = 0.$$

Далее сформулируем достаточное условие экстремума.

Теорема 8 (достаточное условие локального экстремума первого порядка в задаче с ограничениями в виде равенств) [3; 12]. Пусть $x \in \Omega$ и найдётся такой вектор λ , что $\lambda_0 \geq 0$ и $L'_x(x, \lambda) = 0$. Если для любого вектора $dx \neq 0$, удовлетворяющего условиям $g'_i(x)dx = 0, i = 1, \dots, m$, выполняется неравенство $L''_{xx}(x, \lambda)dx^2 > 0$, то в точке x функция $f(x)$ достигает своего строгого локального минимума. Если же $L''_{xx}(x, \lambda)dx^2 < 0$, то в точке x функция $f(x)$ достигает своего строгого локального максимума.

Согласно сформулированным выше теоремам составим алгоритм решения задачи нелинейного программирования с ограничениями-равенствами методом множителей Лагранжа.

Алгоритм решения задачи нелинейного программирования с ограничениями-равенствами методом множителей Лагранжа [19]:

1. Составить функцию Лагранжа $L(x, \lambda_0, \dots, \lambda_m) = \lambda_0 f(x) + \sum_{i=1}^m g_i(x) \lambda_i$.
2. Выписать необходимые условия экстремума:

$$\begin{cases} \frac{\partial L(x, \lambda)}{\partial x_j} = \lambda_0 \frac{\partial f(x)}{\partial x_j} + \lambda_i \sum_{i=1}^m \frac{\partial g_i(x)}{\partial x_j} = 0, j = 1, \dots, n; \\ g_i(x) = 0, i = 1, \dots, m. \end{cases} \quad (2)$$

3. Решить полученную систему. Если x является решением задачи (1), то, согласно теореме о необходимом условии экстремума, существует вектор $\lambda \neq 0$, такой, что пара (x, λ) удовлетворяет рассматриваемой системе (2). Это означает, что экстремальные точки задачи (1) будем разыскивать среди решений системы (2), у которых $\lambda \neq 0$.
4. Решения системы (2) разбиваем на две группы: нерегулярные ($\lambda_0 = 0$) и регулярные ($\lambda \neq 0$). Для регулярных решений полагаем, что λ_0 равно некоторой ненулевой константе.
5. Проверить, являются ли найденные точки экстремальными решениями задачи (1). Это можно сделать или по теореме Вейерштрасса, или используя теорему о достаточном условии экстремума (в случае дважды непрерывно дифференцируемых функций $f(x)$ и $g_i(x), i = 1, \dots, m$).

Теорема 9 (теорема Вейерштрасса) [16] Если функция $z = f(x, y)$, непрерывна на компакте, то она достигает на нем своего наибольшего и наименьшего значений.

Рассмотрим ряд примеров на использование метода множителей Лагранжа.

Пример 12. Решить задачу условной оптимизации:

$$\begin{cases} f(x, y) = 4x + 3y \rightarrow \max(\min), \\ x^2 + y^2 - 1 = 0. \end{cases}$$

Составим функцию Лагранжа рассматриваемой задачи:

$$L(x, y, \lambda_0, \lambda_1) = \lambda_0(4x + 3y) + \lambda_1(x^2 + y^2 - 1).$$

Найдём частные производные от функции Лагранжа по всем переменным:

$$\frac{\partial L(x, y, \lambda)}{\partial x} = 4\lambda_0 + 2\lambda_1 x; \quad \frac{\partial L(x, y, \lambda)}{\partial y} = 3\lambda_0 + 2\lambda_1 y.$$

Необходимое условие экстремума имеет вид

$$\begin{cases} \frac{\partial L(x, y, \lambda)}{\partial x} = 0, \\ \frac{\partial L(x, y, \lambda)}{\partial y} = 0, \\ x^2 + y^2 - 1 = 0. \end{cases}$$

То есть

$$\begin{cases} 4\lambda_0 + 2\lambda_1 x = 0, \\ 3\lambda_0 + 2\lambda_1 y = 0, \\ x^2 + y^2 - 1 = 0. \end{cases}$$

Займёмся поиском нерегулярных решений системы. Для этого положим $\lambda_0 = 0$ и получим следующую систему:

$$\begin{cases} 2\lambda_1 x = 0, \\ 2\lambda_1 y = 0, \\ x^2 + y^2 - 1 = 0. \end{cases}$$

Система имеет решение при $\lambda_1 = 0$, т. е. $\lambda = (0, 0)$, следовательно, точки, подозрительные на экстремум, отсутствуют.

Переходим к поиску регулярных решений системы. Полагаем $\lambda_0 = 1$:

$$\begin{cases} 4 + 2\lambda_1 x = 0, \\ 3 + 2\lambda_1 y = 0, \\ x^2 + y^2 - 1 = 0. \end{cases}$$

Решение данной системы $\lambda_1 = \pm \frac{5}{2}$ приводит к двум точкам, подозрительным на экстремум: $A\left(\frac{4}{5}; \frac{3}{5}\right); B\left(-\frac{4}{5}; -\frac{3}{5}\right)$.

В данном примере удобно воспользоваться теоремой Вейерштрасса. Она применима, так как экстремум ищем на окружности, которая является компактным множеством, а целевая функция непрерывна. Значит, $f(x, y)$ достигает своих максимума и минимума на окружности.

Определим, какая точка является максимумом, а какая минимумом. Для этого найдём значения $f(x, y)$ в данных точках. $f(A) = 4 \cdot \frac{4}{5} + 3 \cdot \frac{3}{5} = 5$; $f(B) = -4 \cdot \frac{4}{5} - 3 \cdot \frac{3}{5} = -5$. То есть $A\left(\frac{4}{5}; \frac{3}{5}\right)$ — глобальный максимум, $B\left(-\frac{4}{5}; -\frac{3}{5}\right)$ — глобальный минимум. Других локальных экстремумов функция не имеет.

Проиллюстрируем полученный результат. Построим линии уровня целевой функции и ограничение $x^2 + y^2 - 1 = 0$. Построение будем выполнять в системе компьютерной алгебры Maple. Для этого сначала зададим функцию:

```
> f := 4*x + 3*y;
```

И ограничение:

```
> g := x^2 + y^2 - 1;
```

Изобразим линии уровня функции f и множество $g(x, y) = 0$.

```
> with(plots):
```

```
> q1 := implicitplot(g(x,y)=0, x=-1..1, y=-1..1,
```

```
scaling=constrained):
```

```
> q2 := contourplot(f(x,y), x=-1.2..1.2, y=-1.2..1.2, scaling=constrained,
filled=false, coloring=[red,green], contours=30):
```

```
> display(q1,q2);
```

Подключение опции *scaling=constrained* необходимо для задания одинакового масштаба по координатным осям. Точки минимума и максимума соответствуют точкам касания окружности $g(x, y) = 0$ и линий уровня функции $f(x, y)$ (рис. 13).

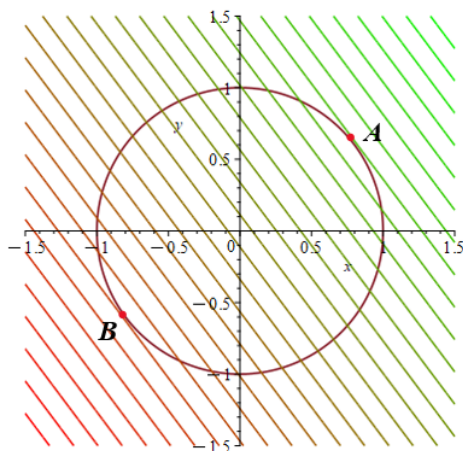


Рис. 13. Линии уровня целевой функции $f(x, y)$ и ограничение $g(x, y) = 0$

Ответ: $\left(\frac{4}{5}; \frac{3}{5}\right)$ — глобальный максимум, $B\left(-\frac{4}{5}; -\frac{3}{5}\right)$ — глобальный минимум.

Пример 13. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = 7x_1x_2 + x_2x_3 \rightarrow \max(\min), \\ 3x_1 + x_2 = 21, \\ x_2 + x_3 = 10. \end{cases}$$

Составим функцию Лагранжа рассматриваемой задачи:

$$L(x, \lambda) = \lambda_0(7x_1x_2 + x_2x_3) + \lambda_1(3x_1 + x_2 - 21) + \lambda_2(x_2 + x_3 - 10).$$

Найдём частные производные от функции Лагранжа по всем переменным:

$$\begin{aligned} \frac{\partial L(x, \lambda)}{\partial x_1} &= 7\lambda_0x_2 + 3\lambda_1; \quad \frac{\partial L(x, \lambda)}{\partial x_2} = 7\lambda_0x_1 + \lambda_0x_3 + \lambda_1 + \lambda_2; \\ \frac{\partial L(x, \lambda)}{\partial x_3} &= \lambda_0x_2 + \lambda_2. \end{aligned}$$

Необходимое условие экстремума имеет вид

$$\begin{cases} 7\lambda_0x_2 + 3\lambda_1 = 0, \\ 7\lambda_0x_1 + \lambda_0x_3 + \lambda_1 + \lambda_2 = 0, \\ \lambda_0x_2 + \lambda_2 = 0, \\ 3x_1 + x_2 = 21, \\ x_2 + x_3 = 10. \end{cases}$$

Займёмся поиском нерегулярных решений системы.

Для этого положим $\lambda_0 = 0$ и получим следующую систему:

$$\begin{cases} 3\lambda_1 = 0, \\ \lambda_1 + \lambda_2 = 0, \\ \lambda_2 = 0, \\ 3x_1 + x_2 = 21, \\ x_2 + x_3 = 10. \end{cases}$$

Система имеет решение при $\lambda_1, \lambda_2 = 0$, т. е. $\lambda = (0, 0, 0)$, следовательно, точки, подозрительные на экстремум, отсутствуют.

Переходим к поиску регулярных решений системы. Полагаем $\lambda_0 = 1$:

$$\begin{cases} 7x_2 + 3\lambda_1 = 0, \\ 7x_1 + x_3 + \lambda_1 + \lambda_2 = 0, \\ x_2 + \lambda_2 = 0, \\ 3x_1 + x_2 = 21, \\ x_2 + x_3 = 10. \end{cases}$$

Решение данной системы $\lambda_1 = -20\frac{13}{20}$; $\lambda_2 = -8\frac{17}{20}$ приводит к точке, подозрительной на экстремум: $A\left(4\frac{1}{20}; 8\frac{17}{20}; 1\frac{3}{20}\right)$.

Воспользуемся теоремой 8 о достаточном условии локального экстремума для проверки точки A на локальный минимум или локальный максимум.

Из условия $g'_i(x)dx = 0, i = 1, \dots, m$ теоремы вытекает система уравнений:

$$\begin{cases} d(3x_1 + x_2 - 21) = 0, \\ d(x_2 + x_3 - 10) = 0. \end{cases}$$

Откуда по свойствам дифференциала:

$$\begin{cases} 3dx_1 + dx_2 = 0, \\ dx_2 + dx_3 = 0 \end{cases} \Rightarrow \begin{cases} dx_1 = -\frac{1}{3}dx_2, \\ dx_3 = -dx_2. \end{cases}$$

Далее определим все частные производные второго порядка от функции Лагранжа:

$$\begin{aligned} \frac{\partial^2 L(x, \lambda)}{\partial x_1^2} &= 0; \quad \frac{\partial^2 L(x, \lambda)}{\partial x_2^2} = 0; \quad \frac{\partial^2 L(x, \lambda)}{\partial x_3^2} = 0; \quad \frac{\partial^2 L(x, \lambda)}{\partial x_1 \partial x_2} = \frac{\partial^2 L(x, \lambda)}{\partial x_2 \partial x_1} = 7; \\ \frac{\partial^2 L(x, \lambda)}{\partial x_1 \partial x_3} &= \frac{\partial^2 L(x, \lambda)}{\partial x_3 \partial x_1} = 0; \quad \frac{\partial^2 L(x, \lambda)}{\partial x_2 \partial x_3} = \frac{\partial^2 L(x, \lambda)}{\partial x_3 \partial x_2} = 1. \end{aligned}$$

Составим второй дифференциал функции Лагранжа. Для этого напомним, что второй дифференциал функции трёх переменных имеет следующий вид (здесь $f = f(x, y, z)$):

$$\begin{aligned} d^2 f &= \frac{\partial^2 f(x, y, z)}{\partial x^2} dx^2 + \frac{\partial^2 f(x, y, z)}{\partial y^2} dy^2 + \frac{\partial^2 f(x, y, z)}{\partial z^2} dz^2 + \\ &+ 2 \frac{\partial^2 f(x, y, z)}{\partial x \partial y} dx dy + 2 \frac{\partial^2 f(x, y, z)}{\partial x \partial z} dx dz + 2 \frac{\partial^2 f(x, y, z)}{\partial y \partial z} dy dz. \end{aligned}$$

Второй дифференциал функции Лагранжа в точке A имеет вид

$$\begin{aligned} d^2 L(A, \lambda) &= 0dx_1^2 + 0dx_2^2 + 0dx_3^2 + 14dx_1 dx_2 + 0dx_1 dx_3 + 2dx_2 dx_3 = \\ &= 14dx_1 dx_2 + 2dx_2 dx_3. \end{aligned}$$

Воспользуемся полученным ранее выражением $dx_1 = -\frac{1}{3}dx_2$, $dx_3 = -dx_2$ и подставим это выражение во второй дифференциал функции Лагранжа:

$$\begin{aligned} d^2 L(A, \lambda) &= 14dx_1 dx_2 + 2dx_2 dx_3 = -\frac{14}{3}(dx_2)^2 - 2(dx_2)^2 = \\ &= -\frac{20}{3}(dx_2)^2 < 0. \end{aligned}$$

Так как второй дифференциал функции Лагранжа в точке A меньше нуля, значит, точка A – строгий локальный максимум.

Ответ: $A\left(4\frac{1}{20}; 8\frac{17}{20}; 1\frac{3}{20}\right)$ – строгий локальный максимум.

Пример 14. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = x_1 x_2 x_3 \rightarrow \max(\min), \\ 2x_1 x_2 + x_2 x_3 = 12, \\ 2x_1 - x_2 = 8. \end{cases}$$

Составим функцию Лагранжа рассматриваемой задачи:

$$L(x, \lambda) = \lambda_0(x_1 x_2 x_3) + \lambda_1(2x_1 x_2 + x_2 x_3 - 12) + \lambda_2(2x_1 - x_2 - 8).$$

Найдём частные производные от функции Лагранжа по всем переменным:

$$\begin{aligned} \frac{\partial L(x, \lambda)}{\partial x_1} &= \lambda_0 x_2 x_3 + 2\lambda_1 x_2 + 2\lambda_2; \quad \frac{\partial L(x, \lambda)}{\partial x_2} = \\ &= \lambda_0 x_1 x_3 + 2\lambda_1 x_1 + \lambda_1 x_3 - \lambda_2; \\ \frac{\partial L(x, \lambda)}{\partial x_3} &= \lambda_0 x_1 x_2 + \lambda_1 x_2. \end{aligned}$$

Необходимое условие экстремума имеет вид

$$\begin{cases} \lambda_0 x_2 x_3 + 2\lambda_1 x_2 + 2\lambda_2 = 0, \\ \lambda_0 x_1 x_3 + 2\lambda_1 x_1 + \lambda_1 x_3 - \lambda_2 = 0, \\ \lambda_0 x_1 x_2 + \lambda_1 x_2 = 0, \\ 2x_1 x_2 + x_2 x_3 = 12, \\ 2x_1 - x_2 = 8. \end{cases}$$

Займёмся поиском нерегулярных решений системы. Для этого положим $\lambda_0 = 0$ и получим следующую систему:

$$\begin{cases} 2\lambda_1 x_2 + 2\lambda_2 = 0, \\ 2\lambda_1 x_1 + \lambda_1 x_3 - \lambda_2 = 0, \\ \lambda_1 x_2 = 0, \\ 2x_1 x_2 + x_2 x_3 = 12, \\ 2x_1 - x_2 = 8. \end{cases}$$

Система имеет решение при $\lambda_1, \lambda_2 = 0$, т. е. $\lambda = (0, 0, 0)$, следовательно, точки, подозрительные на экстремум, отсутствуют.

Переходим к поиску регулярных решений системы. Полагаем $\lambda_0 = 1$:

$$\begin{cases} x_2x_3 + 2\lambda_1x_2 + 2\lambda_2 = 0, \\ x_1x_3 + 2\lambda_1x_1 + \lambda_1x_3 - \lambda_2 = 0, \\ x_1x_2 + \lambda_1x_2 = 0, \\ 2x_1x_2 + x_2x_3 = 12, \\ 2x_1 - x_2 = 8. \end{cases}$$

Решение данной системы $\lambda_{11} = -3$; $\lambda_{12} = \frac{1}{2}$; $\lambda_{21} = -18$; $\lambda_{22} = -\frac{2}{9}$ приводит к точкам, подозрительным на экстремум: $A(3; -2; -12)$; $B\left(-\frac{1}{3}; -\frac{26}{3}; -\frac{28}{39}\right)$.

Воспользуемся теоремой о достаточном условии экстремума для проверки точек на локальный минимум или локальный максимум.

Из условия $g'_i(x)dx = 0, i = 1, \dots, m$ теоремы вытекает система уравнений:

$$\begin{cases} d(2x_1x_2 + x_2x_3 - 12) = 0, \\ d(2x_1 - x_2 - 8) = 0. \end{cases}$$

Откуда по свойствам дифференциала:

$$\begin{cases} 2d(x_1x_2) + d(x_2x_3) = 0, \\ 2dx_1 - dx_2 = 0 \end{cases} \Rightarrow$$

$$\begin{cases} 2dx_1x_2 + 2dx_2x_1 + dx_2x_3 + dx_3x_2 = 0, \\ dx_2 = 2dx_1. \end{cases}$$

Далее определим все частные производные второго порядка от функции Лагранжа:

$$\begin{aligned} \frac{\partial^2 L(x, \lambda)}{\partial x_1^2} &= 0; \frac{\partial^2 L(x, \lambda)}{\partial x_2^2} = 0; \frac{\partial^2 L(x, \lambda)}{\partial x_3^2} = 0; \frac{\partial^2 L(x, \lambda)}{\partial x_1 \partial x_2} = \\ &= \frac{\partial^2 L(x, \lambda)}{\partial x_2 \partial x_1} = x_3 + 2\lambda_1; \\ \frac{\partial^2 L(x, \lambda)}{\partial x_1 \partial x_3} &= \frac{\partial^2 L(x, \lambda)}{\partial x_3 \partial x_1} = x_2; \frac{\partial^2 L(x, \lambda)}{\partial x_2 \partial x_3} = \frac{\partial^2 L(x, \lambda)}{\partial x_3 \partial x_2} = x_1 + \lambda_1. \end{aligned}$$

Второй дифференциал функции Лагранжа имеет вид

$$\begin{aligned} d^2L(x, \lambda) &= 0dx_1^2 + 0dx_2^2 + 0dx_3^2 + 2(x_3 + 2\lambda_1)dx_1dx_2 + 2x_2dx_1dx_3 + \\ &\quad + 2(x_1 + \lambda_1)dx_2dx_3 = \\ &= 2(x_3 + 2\lambda_1)dx_1dx_2 + 2x_2dx_1dx_3 + 2(x_1 + \lambda_1)dx_2dx_3. \end{aligned}$$

Второй дифференциал функции Лагранжа в точке $A(3; -2; -12)$ имеет вид

$$d^2L(x, \lambda) = -36dx_1dx_2 - 4dx_1dx_3.$$

Воспользуемся полученным ранее выражением $dx_2 = 2dx_1$. Также упростим первое уравнение системы $2dx_1x_2 + 2dx_2x_1 + dx_2x_3 + dx_3x_2 = 0$ для рассматриваемой точки:

$$-4dx_1 + 6dx_2 - 12dx_2 - 2dx_3 = 0,$$

$$-4dx_1 - 6dx_2 - 2dx_3 = 0,$$

$$-4dx_1 - 12dx_1 - 2dx_3 = 0,$$

$$-16dx_1 - 2dx_3 = 0,$$

$$dx_3 = -8dx_1.$$

Подставим эти выражения во второй дифференциал функции Лагранжа:

$$d^2L(A, \lambda) = -72(dx_1)^2 + 32(dx_1)^2 = -40(dx_1)^2 < 0.$$

Так как второй дифференциал функции Лагранжа в точке A меньше нуля, значит, точка A – строгий локальный максимум.

Второй дифференциал функции Лагранжа имеет вид

$$d^2L(x, \lambda) = 0dx_1^2 + 0dx_2^2 + 0dx_3^2 + 2(x_3 + 2\lambda_1)dx_1dx_2 + 2x_2dx_1dx_3 + + 2(x_1 + \lambda_1)dx_2dx_3 =$$

$$= 2(x_3 + 2\lambda_1)dx_1dx_2 + 2x_2dx_1dx_3 + 2(x_1 + \lambda_1)dx_2dx_3.$$

Аналогично для точки $B\left(-\frac{1}{3}; -\frac{26}{3}; -\frac{28}{39}\right)$ имеем $d^2L(B, \lambda) > 0$.

Так как второй дифференциал функции Лагранжа в точке B больше нуля, значит, точка B – строгий локальный минимум.

Ответ: $A(3; -2; -12)$ – строгий локальный максимум;
 $B\left(-\frac{1}{3}; -\frac{26}{3}; -\frac{28}{39}\right)$ – строгий локальный минимум.

Задания для самостоятельного решения

1. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = e^{x_1x_2} \rightarrow \max(\min), \\ x_1 + x_2 - 1 = 0. \end{cases}$$

2. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = x_1^2 x_2 (4 - x_1 - x_2) \rightarrow \max(\min), \\ x_1 + 2x_2 = 4. \end{cases}$$

3. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = (x_1 - 2)^2 + x_2^2 \rightarrow \max(\min), \\ x_1^2 - x_2^3 = 0. \end{cases}$$

4. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = \frac{1}{2}x_1^2 + x_2^2 \rightarrow \max(\min), \\ x_1^3 + x_2^3 = 1. \end{cases}$$

5. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = -x_1 x_2 - x_2 x_3 \rightarrow \max(\min), \\ x_1 + x_2 = 2, \\ x_2 + x_3 = 2. \end{cases}$$

7. Задача нелинейного программирования с ограничениями-равенствами и неравенствами. Метод множителей Лагранжа

Рассмотрим задачу условной оптимизации с ограничениями-равенствами и неравенствами вида

$$\begin{cases} f(x) \rightarrow \max(\min), \\ g_i(x) = 0, i = 1, \dots, m_1, \\ g_i(x) \leq 0, i = m_1 + 1, \dots, m, \end{cases} \quad (3)$$

где $f(x)$, $g_i(x)$, $i = 1, \dots, m$ — непрерывно дифференцируемые функции.

Функция Лагранжа имеет вид

$$L(x, \lambda_0, \dots, \lambda_m) = \lambda_0 f(x) + \sum_{i=1}^m g_i(x) \lambda_i,$$

где λ_i , $i = 0, \dots, m$ — множители Лагранжа.

Теорема 10 (необходимое условие локального экстремума первого порядка в задаче с ограничениями в виде равенств и неравенств) [20]. Пусть допустимая точка x — точка локального минимума в задаче (3). Функции $f(x)$, $g_i(x)$, $i = 1, \dots, m$ — непрерывно дифференцируемые функции в окрестности точки x . Тогда найдётся ненулевой набор множителей Лагранжа λ , такой, что для функции Лагранжа выполняются следующие условия:

1. $L'_x(x, \lambda) = 0$ (условие стационарности по x).
2. $\lambda_i g_i(x) = 0$, $i = m_1 + 1, \dots, m$ (условие дополняющей нежёсткости).
3. $\lambda_i \geq 0$, $i = 0, i = m_1 + 1, \dots, m$.

Переходим к формулировке достаточного условия. Для этого введём некоторые обозначения.

Пусть $A(x)$ — множество активных индексов, т. е. номера тех неравенств, которые выполняются в точке x как равенства, $P(x)$ — множество пассивных индексов, т. е. номера тех неравенств, которые выполняются в точке x как строгие неравенства.

Получаем, что точка x лежит на границе области, соответствующей ограничению-неравенству с активным индексом и строго внутри области, соответствующей ограничению-неравенству с пассивным индексом.

Тогда в точке x множество допустимых направлений примет вид

$$H(x) = \{h \in \mathbb{R}^n: (f'(x), h) \leq 0, (g'_i(x), h) \leq 0, i \in A(x), \\ (g'_i(x), h) = 0, i = 1, \dots, m_1\}.$$

Теорема 11 (достаточное условие локального экстремума первого порядка в задаче с ограничениями в виде равенств и неравенств) [20]. Пусть x – допустимая точка, функции $f(x)$, $g_i(x)$, $i = 1, \dots, m$ – дважды непрерывно дифференцируемые функции в окрестности точки x . Пусть существует набор множителей Лагранжа λ , такой, что:

1. Выполняются условия теоремы о необходимых условиях локального минимума.
2. $(L''_{xx}(x, \lambda)h, h) > 0 \quad \forall h \in H(x), h \neq 0$.

Согласно сформулированным выше теоремам составим алгоритм решения задачи нелинейного программирования с ограничениями – равенствами и неравенствами методом множителей Лагранжа.

Алгоритм решения задачи нелинейного программирования с ограничениями – равенствами и неравенствами методом множителей Лагранжа [20]:

1. Составить функцию Лагранжа $L(x, \lambda_0, \dots, \lambda_m) = \lambda_0 f(x) + \sum_{i=1}^m g_i(x) \lambda_i$.
2. Выписать необходимые условия локального экстремума:

$$\begin{cases} \frac{\partial L(x, \lambda)}{\partial x_j} = \lambda_0 \frac{\partial f(x)}{\partial x_j} + \lambda_i \sum_{i=1}^m \frac{\partial g_i(x)}{\partial x_j} = 0, j = 1, \dots, n, \\ \lambda_i g_i(x) = 0, \lambda_i \geq 0, i = m_1 + 1, \dots, m. \end{cases}$$

3. Решить полученную систему, т. е. найти стационарные точки, для которых не все множители Лагранжа обращаются в ноль. Рассмотреть отдельно случаи $\lambda_0 = 0$ и $\lambda_0 \neq 0$. В случае $\lambda_0 \neq 0$ полагаем λ_0 равным некоторой константе (удобно положить $\lambda_0 = 1$).
4. Определить, какая из найденных стационарных точек является минимумом, а какая максимумом. Для этого следует воспользоваться достаточным условием оптимальности.
5. Проверить, есть ли в рассматриваемой задаче глобальные максимум и минимум.

Рассмотрим ряд примеров.

Пример 15. Решить задачу условной оптимизации:

$$f(x) = 29x_1^2 - 54x_1x_2 + 26x_2^2 \rightarrow \max(\min),$$

$$\begin{cases} 3x_1 - 4x_2 = 1, \\ 2x_1 - x_2 \leq 0. \end{cases}$$

Составим функцию Лагранжа рассматриваемой задачи:

$$L(x, \lambda) = \lambda_0(29x_1^2 - 54x_1x_2 + 26x_2^2) + \lambda_1(3x_1 - 4x_2 - 1) + \lambda_2(2x_1 - x_2).$$

Найдём частные производные от функции Лагранжа по всем переменным:

$$\frac{\partial L(x, \lambda)}{\partial x_1} = 58\lambda_0x_1 - 54\lambda_0x_2 + 3\lambda_1 + 2\lambda_2;$$

$$\frac{\partial L(x, \lambda)}{\partial x_2} = -54\lambda_0x_1 + 52\lambda_0x_2 - 4\lambda_1 - \lambda_2.$$

Условие стационарности имеет вид

$$\begin{cases} 58\lambda_0x_1 - 54\lambda_0x_2 + 3\lambda_1 + 2\lambda_2 = 0, \\ -54\lambda_0x_1 + 52\lambda_0x_2 - 4\lambda_1 - \lambda_2 = 0. \end{cases}$$

Условие дополняющей нежёсткости:

$$\lambda_2(2x_1 - x_2) = 0.$$

Условия неотрицательности (в случае задачи на минимум):

$$\begin{cases} \lambda_0 \geq 0, \\ \lambda_2 \geq 0. \end{cases}$$

Условия неотрицательности (в случае задачи на максимум):

$$\begin{cases} \lambda_0 \leq 0, \\ \lambda_2 \geq 0. \end{cases}$$

Пусть $\lambda_0 = 0$, тогда из условия стационарности получим:

$$\begin{cases} 3\lambda_1 + 2\lambda_2 = 0, \\ -4\lambda_1 - \lambda_2 = 0. \end{cases}$$

Значит, $\lambda_1 = \lambda_2 = 0$, т. е. экстремумов нет.

а) Пусть $\lambda_0 = 1$. То есть рассматриваем решение задачи минимизации. Имеем систему:

$$\begin{cases} 58x_1 - 54x_2 + 3\lambda_1 + 2\lambda_2 = 0, \\ -54x_1 + 52x_2 - 4\lambda_1 - \lambda_2 = 0, \\ \lambda_2(2x_1 - x_2) = 0, \\ \lambda_2 \geq 0, \\ 3x_1 - 4x_2 = 1, \\ 2x_1 - x_2 \leq 0. \end{cases}$$

Решаем данную систему. Складываем первое и второе уравнения системы:

$$4x_1 - 2x_2 - \lambda_1 + \lambda_2 = 0.$$

Откуда

$$x_2 = 2x_1 - \frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2.$$

Подставим полученное выражение для x_2 во второе уравнение системы:

$$-54x_1 + 52\left(2x_1 - \frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2\right) - 4\lambda_1 - \lambda_2 = 0,$$

$$-54x_1 + 104x_1 - 26\lambda_1 + 26\lambda_2 - 4\lambda_1 - \lambda_2 = 0,$$

$$50x_1 - 30\lambda_1 + 25\lambda_2 = 0.$$

$$10x_1 - 6\lambda_1 + 5\lambda_2 = 0.$$

Выразим отсюда x_1 : $x_1 = \frac{3}{5}\lambda_1 - \frac{1}{2}\lambda_2$ и подставим в выражение для x_2 :

$$\begin{aligned} x_2 &= 2x_1 - \frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2 = 2\left(\frac{3}{5}\lambda_1 - \frac{1}{2}\lambda_2\right) - \frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2 = \\ &= \frac{6}{5}\lambda_1 - \lambda_2 - \frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2 = \frac{7}{10}\lambda_1 - \frac{1}{2}\lambda_2. \end{aligned}$$

Из третьего уравнения системы следует, что или $\lambda_2 = 0$, или $2x_1 - x_2 = 0$.

1) Рассмотрим случай, когда $\lambda_2 = 0$. Получим, что $x_1 = \frac{3}{5}\lambda_1$,

$x_2 = \frac{7}{10}\lambda_1$. Подставим в пятое уравнение системы:

$$3x_1 - 4x_2 = 1,$$

$$\frac{9}{5}\lambda_1 - \frac{14}{5}\lambda_1 = 1,$$

$$-\lambda_1 = 1,$$

$$\lambda_1 = -1.$$

Отсюда следует, что $x_1 = -\frac{3}{5}$, $x_2 = -\frac{7}{10}$. Это решение удовлетворяет последнему неравенству системы, т. е. получена точка $A\left(-\frac{3}{5}; -\frac{7}{10}\right)$, подозрительная на экстремум и удовлетворяющая необходимым условиям минимума.

2) Рассмотрим случай, когда $2x_1 - x_2 = 0$. Перейдём к системе:

$$\begin{cases} 3x_1 - 4x_2 = 1, \\ 2x_1 - x_2 = 0. \end{cases}$$

Откуда $x_1 = -\frac{1}{5}$, $x_2 = -\frac{2}{5}$. Продолжим решение системы. Подставим найденные значения x_1 и x_2 в первое и второе уравнения системы:

$$\begin{cases} 58 \cdot \left(-\frac{1}{5}\right) - 54 \cdot \left(-\frac{2}{5}\right) + 3\lambda_1 + 2\lambda_2 = 0, \\ -54 \cdot \left(-\frac{1}{5}\right) + 52 \cdot \left(-\frac{2}{5}\right) - 4\lambda_1 - \lambda_2 = 0, \end{cases}$$

$$\begin{cases} -\frac{58}{5} + \frac{108}{5} + 3\lambda_1 + 2\lambda_2 = 0, \\ \frac{54}{5} - \frac{104}{5} - 4\lambda_1 - \lambda_2 = 0, \end{cases}$$

$$\begin{cases} 3\lambda_1 + 2\lambda_2 = -10, \\ -8\lambda_1 - 2\lambda_2 = 20, \end{cases}$$

$$-5\lambda_1 = 10 \Rightarrow \lambda_1 = -2,$$

$$16 - 2\lambda_2 = 20 \Rightarrow \lambda_2 = -2.$$

Так как $\lambda_1 = -2$ и $\lambda_2 = -2$, то условие неотрицательности не выполняется и рассматриваемая точка с координатами $x_1 = -\frac{1}{5}$, $x_2 = -\frac{2}{5}$ не является точкой, подозрительной на экстремум.

б) Пусть $\lambda_0 = -1$. То есть рассматриваем решение задачи максимизации. Имеем систему:

$$\begin{cases} -58x_1 + 54x_2 + 3\lambda_1 + 2\lambda_2 = 0, \\ 54x_1 - 52x_2 - 4\lambda_1 - \lambda_2 = 0, \\ \lambda_2(2x_1 - x_2) = 0, \\ \lambda_2 \geq 0, \\ 3x_1 - 4x_2 = 1, \\ 2x_1 - x_2 \leq 0. \end{cases}$$

Решаем данную систему. Складываем первое и второе уравнения системы:

$$-4x_1 + 2x_2 - \lambda_1 + \lambda_2 = 0.$$

Откуда

$$x_2 = 2x_1 + \frac{1}{2}\lambda_1 - \frac{1}{2}\lambda_2.$$

Подставим полученное выражение для x_2 во второе уравнение системы:

$$54x_1 - 52\left(2x_1 + \frac{1}{2}\lambda_1 - \frac{1}{2}\lambda_2\right) - 4\lambda_1 - \lambda_2 = 0,$$

$$54x_1 - 104x_1 - 26\lambda_1 + 26\lambda_2 - 4\lambda_1 - \lambda_2 = 0,$$

$$-50x_1 - 30\lambda_1 + 25\lambda_2 = 0,$$

$$-10x_1 - 6\lambda_1 + 5\lambda_2 = 0.$$

Выразим отсюда x_1 : $x_1 = -\frac{3}{5}\lambda_1 + \frac{1}{2}\lambda_2$ и подставим в выражение для x_2 :

$$\begin{aligned} x_2 &= 2x_1 + \frac{1}{2}\lambda_1 - \frac{1}{2}\lambda_2 = 2\left(-\frac{3}{5}\lambda_1 + \frac{1}{2}\lambda_2\right) + \frac{1}{2}\lambda_1 - \frac{1}{2}\lambda_2 = \\ &= -\frac{6}{5}\lambda_1 + \lambda_2 + \frac{1}{2}\lambda_1 - \frac{1}{2}\lambda_2 = -\frac{7}{10}\lambda_1 + \frac{1}{2}\lambda_2. \end{aligned}$$

Из третьего уравнения системы следует, что или $\lambda_2 = 0$, или $2x_1 - x_2 = 0$.

- 1) Рассмотрим случай, когда $\lambda_2 = 0$. Получим, что $x_1 = -\frac{3}{5}\lambda_1$, $x_2 = -\frac{7}{10}\lambda_1$. Подставим в пятое уравнение системы:

$$3x_1 - 4x_2 = 1,$$

$$-\frac{9}{5}\lambda_1 + \frac{14}{5}\lambda_1 = 1,$$

$$\lambda_1 = 1.$$

Отсюда следует, что $x_1 = -\frac{3}{5}$, $x_2 = -\frac{7}{10}$. Это решение удовлетворяет последнему неравенству системы, т. е. получена точка $A\left(-\frac{3}{5}; -\frac{7}{10}\right)$, подозрительная на экстремум и удовлетворяющая необходимым условиям максимума.

- 2) Рассмотрим случай, когда $2x_1 - x_2 = 0$. Перейдём к системе:

$$\begin{cases} 3x_1 - 4x_2 = 1, \\ 2x_1 - x_2 = 0. \end{cases}$$

Откуда $x_1 = -\frac{1}{5}$, $x_2 = -\frac{2}{5}$. Продолжим решение системы. Подставим найденные значения x_1 и x_2 в первое и второе уравнения системы:

$$\begin{cases} -58 \cdot \left(-\frac{1}{5}\right) + 54 \cdot \left(-\frac{2}{5}\right) + 3\lambda_1 + 2\lambda_2 = 0, \\ 54 \cdot \left(-\frac{1}{5}\right) - 52 \cdot \left(-\frac{2}{5}\right) - 4\lambda_1 - \lambda_2 = 0, \end{cases}$$

$$\begin{cases} \frac{58}{5} - \frac{108}{5} + 3\lambda_1 + 2\lambda_2 = 0, \\ -\frac{54}{5} + \frac{104}{5} - 4\lambda_1 - \lambda_2 = 0, \\ \begin{cases} 3\lambda_1 + 2\lambda_2 = 10, \\ -8\lambda_1 - 2\lambda_2 = -20, \end{cases} \\ 5\lambda_1 = 10 \Rightarrow \lambda_1 = 2, \\ -16 - 2\lambda_2 = -20 \Rightarrow \lambda_2 = 2. \end{cases}$$

Так как $\lambda_1 = 2$ и $\lambda_2 = 2$, то условие неотрицательности выполняется и рассматриваемая точка с координатами $B\left(-\frac{1}{5}; -\frac{2}{5}\right)$ является точкой, подозрительной на экстремум. Она удовлетворяет необходимым условиям локального максимума.

Переходим к проверке найденных точек с помощью достаточных условий экстремума.

1. Проверим точку $A\left(-\frac{3}{5}; -\frac{7}{10}\right)$, подозрительную на минимум, с помощью теоремы о достаточном условии локального минимума. Для этого следует определить знак выражения $(L''_{xx}(x, \lambda)h, h)$, $\forall h \in H(x), h \neq 0$, $H(x) = \{h \in \mathbb{R}^n: (f'(x), h) \leq 0, (g'_i(x), h) \leq 0, i \in A(x), (g'_i(x), h) = 0, i = 1, \dots, m_1\}$. Здесь напомним, что $A(x)$ — множество активных индексов, т. е. номера тех неравенств, которые выполняются в точке x как равенства, $P(x)$ — множество пассивных индексов, т. е. номера тех неравенств, которые выполняются в точке x как строгие неравенства.

Целевая функция $f_0(x) = 29x_1^2 - 54x_1x_2 + 26x_2^2$ и ограничения задачи $f_1(x) = 3x_1 - 4x_2, f_2(x) = 2x_1 - x_2$.

Векторы частных производных первого порядка данных функций имеют вид

$$f'_0(x) = (58x_1 - 54x_2, -54x_1 + 52x_2), f'_1(x) = (3, -4), f'_2(x) = (2, -1).$$

Для точки A множество активных индексов пусто, следовательно,

$$H(A) = \{h \in \mathbb{R}^2: (f'_0(x), h) \leq 0, (g'_1(x), h) = 0\} = \{h = (h_1, h_2): 3h_1 - 4h_2 \leq 0, 3h_1 - 4h_2 = 0\} = \{h = \left(h_1, \frac{3h_1}{4}\right), h_1 \in \mathbb{R}\}.$$

Матрица вторых производных имеет вид

$$\frac{\partial^2 L(x, \lambda)}{\partial x^2} = \begin{pmatrix} 58\lambda_0 & -54\lambda_0 \\ -54\lambda_0 & 52\lambda_0 \end{pmatrix}.$$

При $\lambda_0 = 1$ матрица вторых производных примет вид

$$\frac{\partial^2 L(x, \lambda)}{\partial x^2} = \begin{pmatrix} 58 & -54 \\ -54 & 52 \end{pmatrix}.$$

Матрица является положительно определённой, следовательно,

$$\left(L''_{xx} \left(\left(-\frac{3}{5}; -\frac{7}{10} \right), \lambda \right) h, h \right) > 0.$$

Это означает, что точка $A \left(-\frac{3}{5}; -\frac{7}{10} \right)$ является строгим локальным минимумом (по теореме о достаточном условии локального минимума).

2. Проверим точку $B \left(-\frac{1}{5}; -\frac{2}{5} \right)$, подозрительную на максимум, с помощью теоремы о достаточном условии локального минимума.

Для точки B множество активных индексов непусто, следовательно,

$$\begin{aligned} H(B) &= \{h \in \mathbb{R}^2: (f'_0(x), h) \leq 0, (f'_1(x), h) = 0, (f'_2(x), h) \leq 0\} = \\ &= \{h = (h_1, h_2): 3h_1 - 4h_2 = 0, 10h_1 - 10h_2 \leq 0, 2h_1 - h_2 \leq 0\} = \\ &= \{h = \left(h_1, \frac{3h_1}{4} \right), h_1 \leq 0\}. \end{aligned}$$

При $\lambda_0 = -1$ матрица вторых производных примет вид

$$\frac{\partial^2 L(x, \lambda)}{\partial x^2} = \begin{pmatrix} -58 & 54 \\ 54 & -52 \end{pmatrix}.$$

Матрица является отрицательно определённой, следовательно,

$$\left(L''_{xx} \left(\left(-\frac{3}{5}; -\frac{7}{10} \right), \lambda \right) h, h \right) < 0.$$

Это означает, что точка $B \left(-\frac{1}{5}; -\frac{2}{5} \right)$ является строгим локальным максимумом (по теореме о достаточном условии локального максимума).

Проиллюстрируем полученный результат. Построим линии уровня целевой функции и ограничения. Область допустимых решений выделена красным цветом (рис. 14).

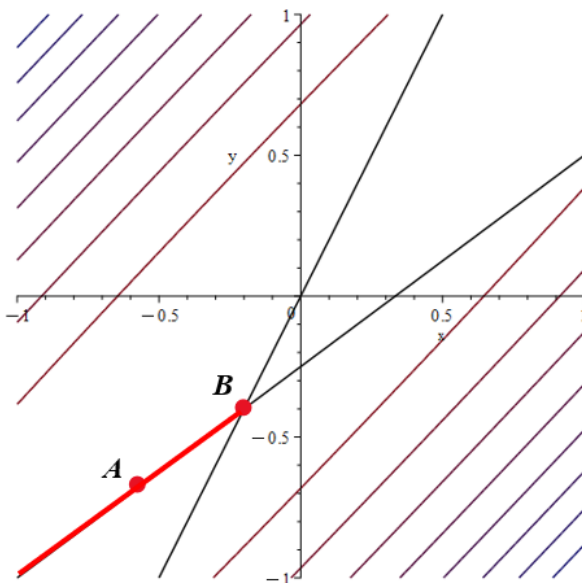


Рис. 14. Линии уровня целевой функции $f(x, y)$ и ОДР задачи

Ответ: $A\left(-\frac{3}{5}; -\frac{7}{10}\right)$ – строгий локальный минимум; $B\left(-\frac{1}{5}; -\frac{2}{5}\right)$ – строгий локальный максимум.

Пример 16. Решить задачу условной оптимизации:

$$f(x) = x_1 x_2 x_3 \rightarrow \max(\min),$$

$$x_1^2 + x_2^2 + x_3^2 \leq 1.$$

Функция Лагранжа имеет вид

$$L(x, \lambda) = \lambda_0 x_1 x_2 x_3 + \lambda_1 (x_1^2 + x_2^2 + x_3^2 - 1).$$

Найдём частные производные от функции Лагранжа по всем переменным:

$$\frac{\partial L(x, \lambda)}{\partial x_1} = \lambda_0 x_2 x_3 + 2\lambda_1 x_1;$$

$$\frac{\partial L(x, \lambda)}{\partial x_2} = \lambda_0 x_1 x_3 + 2\lambda_1 x_2;$$

$$\frac{\partial L(x, \lambda)}{\partial x_3} = \lambda_0 x_1 x_2 + 2\lambda_1 x_3.$$

Условие стационарности имеет вид

$$\begin{cases} \lambda_0 x_2 x_3 + 2\lambda_1 x_1 = 0, \\ \lambda_0 x_1 x_3 + 2\lambda_1 x_2 = 0, \\ \lambda_0 x_1 x_2 + 2\lambda_1 x_3 = 0. \end{cases}$$

Условие дополняющей нежёсткости

$$\lambda_1(x_1^2 + x_2^2 + x_3^2 - 1) = 0.$$

Условия неотрицательности (в случае задачи на минимум):

$$\begin{cases} \lambda_0 \geq 0, \\ \lambda_1 \geq 0. \end{cases}$$

Условия неотрицательности (в случае задачи на максимум):

$$\begin{cases} \lambda_0 \leq 0, \\ \lambda_1 \geq 0. \end{cases}$$

Пусть $\lambda_0 = 0$, тогда система имеет вид

$$\begin{cases} 2\lambda_1 x_1 = 0, \\ 2\lambda_1 x_2 = 0, \\ 2\lambda_1 x_3 = 0, \\ \lambda_1(x_1^2 + x_2^2 + x_3^2 - 1) = 0. \end{cases}$$

Значит, $\lambda_1 = 0$, и получаем нулевой вектор λ , это означает, что нет точек, подозрительных на экстремум.

с) Пусть $\lambda_0 \neq 0$. Положим $\lambda_0 = 2$. То есть рассматриваем решение задачи минимизации. Имеем систему:

$$\begin{cases} x_2 x_3 + \lambda_1 x_1 = 0, \\ x_1 x_3 + \lambda_1 x_2 = 0, \\ x_1 x_2 + \lambda_1 x_3 = 0, \\ \lambda_1(x_1^2 + x_2^2 + x_3^2 - 1) = 0. \end{cases}$$

Решаем данную систему. Из последнего уравнения системы получим, что или $\lambda_1 = 0$, или $x_1^2 + x_2^2 + x_3^2 - 1 = 0$.

Если $\lambda_1 = 0$, то $x_2 x_3 = x_1 x_3 = x_1 x_2 = 0$. Решение системы представляет собой множество точек в трёхмерном пространстве, у которых или первая, или вторая, или третья координата ненулевые, а остальные равны нулю. Причём модуль ненулевой координаты не превосходит единицу.

Если $x_1^2 + x_2^2 + x_3^2 - 1 = 0$, то для решения системы умножим первое уравнение на x_1 , второе на x_2 , третье на x_3 :

$$\begin{cases} x_1 x_2 x_3 + \lambda_1 x_1^2 = 0, \\ x_1 x_2 x_3 + \lambda_1 x_2^2 = 0, \\ x_1 x_2 x_3 + \lambda_1 x_3^2 = 0, \\ \lambda_1 \neq 0. \end{cases}$$

Из первых трёх уравнений системы получим $\lambda_1 x_1^2 = \lambda_1 x_2^2 = \lambda_1 x_3^2 = -x_1 x_2 x_3$, т. е. $x_1^2 = x_2^2 = x_3^2$.

Подставим это выражение в $x_1^2 + x_2^2 + x_3^2 - 1 = 0$, получим $3x_1^2 = 1$.

Тогда $x_1 = \pm \frac{\sqrt{3}}{3}$. Решениями системы являются точки с координатами $(\pm \frac{\sqrt{3}}{3}; \pm \frac{\sqrt{3}}{3}; \pm \frac{\sqrt{3}}{3})$ со всевозможными комбинациями знаков.

Вспомним, что задача условной оптимизации решается на множестве $x_1^2 + x_2^2 + x_3^2 \leq 1$, которое является компактным. В данном случае применима теорема Вейерштрасса, из которой следует, что рассматриваемая функция достигает своего глобального минимума и глобального максимума на рассматриваемой области.

В точках, являющихся решением в случае, если $\lambda_1 = 0$, целевая функция принимает значение, равное 0. А в точках вида $(\pm \frac{\sqrt{3}}{3}; \pm \frac{\sqrt{3}}{3}; \pm \frac{\sqrt{3}}{3})$ целевая функция принимает значение, равное $\pm \frac{\sqrt{3}}{9}$. Значит, точки, в которых функция принимает значение $\frac{\sqrt{3}}{9}$, являются точками глобального максимума, а точки, в которых функция принимает значение $-\frac{\sqrt{3}}{9}$, являются точками глобального минимума.

Ответ: точки вида $(\pm \frac{\sqrt{3}}{3}; \pm \frac{\sqrt{3}}{3}; \pm \frac{\sqrt{3}}{3})$ являются точками экстремума, причём точки, доставляющие целевой функции значение $\frac{\sqrt{3}}{9}$, являются точками глобального максимума, а точки, в которых функция принимает значение $-\frac{\sqrt{3}}{9}$, являются точками глобального минимума.

Проиллюстрируем полученный результат. Построим область, заданную ограничением $x_1^2 + x_2^2 + x_3^2 \leq 1$. Это область внутри единичного шара. Построение будем выполнять в системе компьютерной алгебры Maple. Для этого сначала зададим функцию:

$$> g := x1^2 + x2^2 + x3^2 ;$$

Изобразим графически область, удовлетворяющую неравенству $x_1^2 + x_2^2 + x_3^2 \leq 1$ (рис. 15):

```
> with(plots):
> plots:-implicitplot3d(f <= 1, x1 = -2 .. 2, x2 = -2 .. 2, x3 = -2 .. 2, style =
surface, grid = [50, 50, 50], color = blue, view = [-2 .. 2, -2 .. 2, -2 .. 2])
```

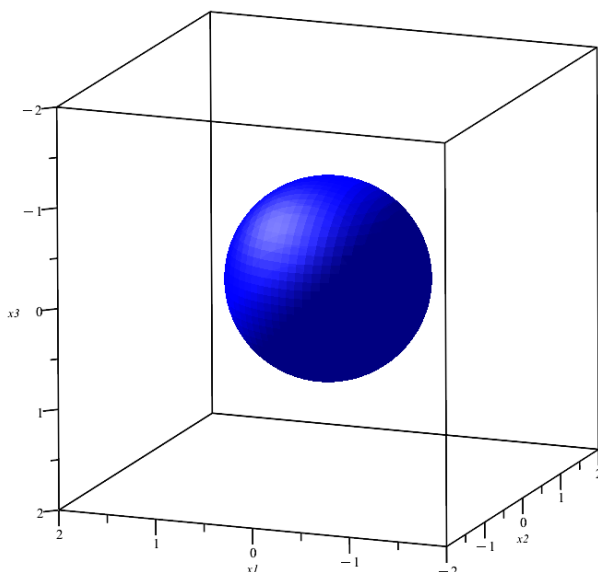


Рис. 15. Область, удовлетворяющая неравенству $x_1^2 + x_2^2 + x_3^2 \leq 1$

Добавим линии уровня целевой функции $f(x) = x_1 x_2 x_3$ (рис. 16). Зададим целевую функцию:

```
> f := x1*x2*x3;
```

Построим линии уровня целевой функции:

```
> q2 := implicitplot3d(f, x1 = -1.2 .. 1.2, x2 = -1.2 .. 1.2, x3 = -1.2 .. 1.2,
scaling = constrained, filled = false, contours = 30)
```

Программный код имеет следующий вид:

```
with(plots);
```

```
> g := x1^2 + x2^2 + x3^2;
```

```
> f := x1*x2*x3;
```

```
> q1 := plots:-implicitplot3d(g <= 1, x1 = -2 .. 2, x2 = -2 .. 2, x3 = -2 .. 2,
style = surface, grid = [50, 50, 50], color = blue, view = [-2 .. 2, -2 .. 2, -2 .. 2]);
```

```
> q2 := implicitplot3d(f, x1 = -1.2 .. 1.2, x2 = -1.2 .. 1.2, x3 = -1.2 .. 1.2,
scaling = constrained, filled = false, contours = 30);
```

```
> display(q1, q2)
```

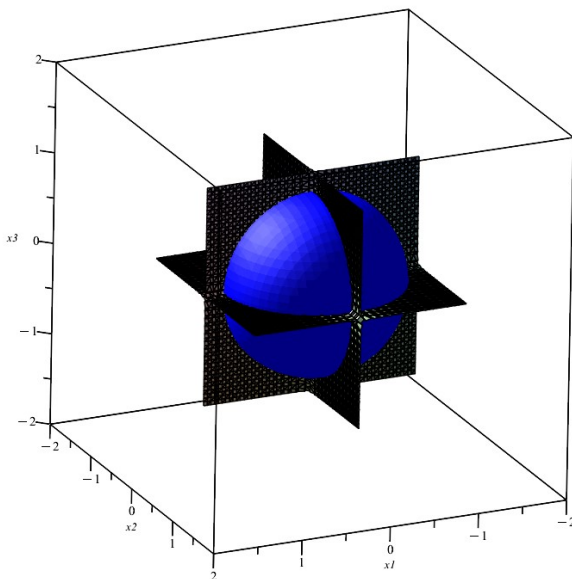


Рис. 16. Область, удовлетворяющая неравенству $x_1^2 + x_2^2 + x_3^2 \leq 1$, и линии уровня функции $f(x) = x_1x_2x_3$

Задания для самостоятельного решения

1. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = (x_1 - 2)^2 + x_2^2 \rightarrow \max(\min), \\ x_1^2 - x_2^3 = 0, \\ x_1 - x_2 \geq 0. \end{cases}$$

2. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = \frac{1}{2}x_1^2 + x_2^2 \rightarrow \max(\min), \\ x_1^3 + x_2^3 = 1, \\ 2x_1 + x_2 \geq 0. \end{cases}$$

3. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = -x_1x_2 - x_2x_3 \rightarrow \max(\min), \\ x_1 + x_2 = 0, \\ x_1^3 + x_2^2 \leq 0. \end{cases}$$

4. Решить задачу условной оптимизации

$$\begin{cases} f(x) = -x_1 + x_2^2 \rightarrow \max(\min), \\ 2x_1 + x_2^2 = 0, \\ x_1 + x_2^3 \leq 0. \end{cases}$$

5. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = x_1^2 + x_2^2 \rightarrow \max(\min), \\ x_1 - 3x_2 = 0, \\ 4x_1^3 + x_2^2 \leq 0. \end{cases}$$

6. Решить задачу условной оптимизации:

$$\begin{cases} f(x) = x_1 x_2 x_3 \rightarrow \max(\min), \\ x_1^2 + x_2^2 + x_3^2 \leq 1. \end{cases}$$

8. Численные методы оптимизации функции одной и нескольких переменных

Рассмотрим применение численных методов [40; 41] для решения оптимизационных задач.

8.1. Безусловная нелинейная оптимизация функции одной переменной

Пусть функция $f(x)$ – числовая функция вещественной переменной x . Тогда задача одномерной оптимизации функции $f(x)$ на отрезке $[a, b]$ имеет вид

$$f(x) \rightarrow \min_{x \in [a, b]}.$$

Задача одномерной оптимизации решается путём локализации точек минимума и дальнейшего поиска оптимального решения на полученном отрезке.

Рассмотрим различные методы решения сформулированной выше задачи [32].

8.1.1. Метод перебора

Метод перебора является простейшим из прямых методов минимизации [1].

Пусть задана одноэкстремальная функция $f(x)$ на отрезке $[a, b]$. Необходимо найти какую-либо из её точек минимума на данном отрезке с заданной точностью $\varepsilon > 0$.

Разобьём отрезок $[a, b]$ на n равных частей так, что

$$x_j = a + jh \quad (j = 0, 1, \dots, n),$$

где $h = \frac{b-a}{n}$ – шаг и n – целое число, $n \geq \frac{b-a}{\varepsilon}$. То есть длина каждого подынтервала не превосходит ε .

Среди всех точек x_j , $(j = 0, 1, \dots, n)$ выбираем точку x_m :

$$f(x_m) = \min_{0 \leq j \leq n} f(x_j).$$

Получаем решение $x^* = x_m$, $f(x^*) = f(x_m)$.

Сформулируем алгоритм метода перебора:

1. Вводим границы отрезка a, b и точность ε .
2. Вычисляем $n = \left\lceil \frac{b-a}{\varepsilon} \right\rceil + 1$; $h = \frac{b-a}{n}$. Полагаем $x_m = a$ и $\min = f(a)$.

3. Для каждого значения $i, i = 1, \dots, n$:
 - 3.1. Вычисляем $x = a + i \cdot h$.
 - 3.2. Вычисляем $y = f(x)$.
 - 3.3. Сравниваем y и $\min$. Если $y \leq \min$, то $\min = y, x_m = x$ и переходим к следующей итерации. Если данное условие не выполняется, то полагаем $i = n$ и заканчиваем вычисления.
4. В результате минимальное значение x находится в переменной x_m , минимальное значение функции $f(x)$ в переменной $\min$.

Пример 17. Найти наименьшее значение функции $f(x) = x^2 - 4x + 3$ на отрезке $[0,1]$ с точностью $\varepsilon = 0,1$.

Определим число отрезков разбиения $n \geq \frac{b-a}{\varepsilon}$, т. е. $n \geq 10$. Разобьём отрезок $[0,1]$ на 10 равных частей. $h = \frac{b-a}{n} = 0,1$.

Полагаем $x_m = 0$ и $\min = f(0) = 3$.

- 1) Пусть $i = 1$, тогда $x = 0 + 1 \cdot 0,1 = 0,1$; $f(x) = 0,01 - 0,4 + 3 = 2,61$. $2,61 < \min$, значит, полагаем $\min = 2,61$; $x_m = 0,1$.
- 2) $i = 2$, тогда $x = 0,2$; $f(x) = 0,04 - 0,8 + 3 = 2,24$. $2,24 < \min$, значит, полагаем $\min = 2,24$; $x_m = 0,2$.
- 3) $i = 3$, тогда $x = 0,3$; $f(x) = 0,09 - 1,2 + 3 = 1,89$. $1,89 < \min$, значит, полагаем $\min = 1,89$; $x_m = 0,3$.
- 4) $i = 4$, тогда $x = 0,4$; $f(x) = 0,16 - 1,6 + 3 = 1,56$. $1,56 < \min$, значит, полагаем $\min = 1,56$; $x_m = 0,4$.
- 5) $i = 5$, тогда $x = 0,5$; $f(x) = 0,25 - 2 + 3 = 1,25$. $1,25 < \min$, значит, полагаем $\min = 1,25$; $x_m = 0,5$.
- 6) $i = 6$, тогда $x = 0,6$; $f(x) = 0,36 - 2,4 + 3 = 0,96$. $0,96 < \min$, значит, полагаем $\min = 0,96$; $x_m = 0,6$.
- 7) $i = 7$, тогда $x = 0,7$; $f(x) = 0,49 - 2,8 + 3 = 0,69$. $0,69 < \min$, значит, полагаем $\min = 0,69$; $x_m = 0,7$.
- 8) $i = 8$, тогда $x = 0,8$; $f(x) = 0,64 - 3,2 + 3 = 0,44$. $0,44 < \min$, значит, полагаем $\min = 0,44$; $x_m = 0,8$.
- 9) $i = 9$, тогда $x = 0,9$; $f(x) = 0,81 - 3,6 + 3 = 0,21$. $0,21 < \min$, значит, полагаем $\min = 0,21$; $x_m = 0,9$.
- 10) $i = 10$, тогда $x = 1$; $f(x) = 1 - 4 + 3 = 0$. $0 < \min$, значит, полагаем $\min = 0$; $x_m = 1$.

Таким образом, получили точку минимума $x_m = 1$ и наименьшее значение функции на рассматриваемом отрезке равно 0.

Ответ: $x_m = 1, f(x_m) = 0$.

Данный метод предполагает большое число вычислений значений функции. Улучшение данного метода будет связано с уменьшением длины отрезка, на котором находится точка минимума.

8.1.2. Метод дихотомии (деления отрезка пополам)

Данный метод более эффективен по сравнению с методом перебора, так как для определения очередного приближения к точке минимума использует уже имеющиеся значения оптимизируемой функции. Метод деления отрезка пополам широко описан в литературе, например [1; 2; 13; 28].

В методе дихотомии строится последовательность вложенных подотрезков, в каждом из которых содержится хотя бы одна из точек минимума.

Рассмотрим задачу минимизации функции $f(x)$ на отрезке $[a, b]$ с точностью ε в предположении, что функция $f(x)$ имеет единственный локальный минимум.

Пусть x^* — точное решение рассматриваемой задачи, $\tilde{x}$ — приближённое решение. Тогда $|\tilde{x} - x^*| < \varepsilon$.

Метод дихотомии основан на делении текущего отрезка $[a, b]$, содержащего экстремум, на две равные части. Далее из этих двух частей выбирается одна, содержащая минимум. Экстремум локализуется путём сравнения двух значений критерия оптимальности в точках, отстоящих от середины отрезка на величину $\delta = \frac{\varepsilon}{2}$ где ε — описанная выше погрешность решения.

Перейдём от описательной части рассматриваемого метода к его применению. Рассмотрим отрезок $[a, b]$ и разделим его точкой c пополам. Получим, что $f(c) < f(a)$ и $f(c) < f(b)$. Точка минимума может находиться как в левой, так и в правой части отрезка $[a, b]$. То есть середина отрезка не способствует сужению отрезка неопределённости.

Поэтому из отрезка $[a, b]$ выберем две точки $x_1 = \frac{a+b}{2} - \frac{\delta}{2}$, $x_2 = \frac{a+b}{2} + \frac{\delta}{2}$, где параметр δ был введён ранее и не превосходит ε .

Таким образом, точка x^* попадёт или в отрезок $[a, x_2]$, или в отрезок $[x_1, b]$. Если $f(x_1) < f(x_2)$, то $x^* \in [a, x_2]$ не может попасть в отрезок

$[x_2, b]$. Если $f(x_1) > f(x_2)$, то $x^* \in [x_1, b]$. Если $f(x_1) = f(x_2)$, то $x^* \in [x_1, x_2]$.

После первого шага задача остаётся прежней, но отрезок неопределённости стал меньше. Многократное применение описанной выше процедуры приведет к тому, что отрезок неопределённости сузится до желаемого.

Сформулируем алгоритм метода дихотомии:

1. Вводим границы отрезка a, b , точность ε , вычисляем параметр $\delta = \frac{\varepsilon}{2}$.
2. Полагаем $x_1 = \frac{a+b-\delta}{2}$; $x_2 = \frac{a+b+\delta}{2}$. Если $f(x_1) < f(x_2)$, полагаем $b = x_2$, иначе полагаем $a = x_1$.
3. Проверяем условие окончания итераций: $|b - a| < \varepsilon$. Если условие окончания итераций выполняется, то задача решена, полагаем $x^* = a$. Если нет – переходим к шагу 2.

Пример 18. Найти наименьшее значение функции $f(x) = x^2 - 4x + 3$ на отрезке $[0; 2]$ с точностью $\varepsilon = 0,1$.

1. Границы отрезка $a = 0, b = 2$, точность $\varepsilon = 0,1$, параметр $\delta = \frac{\varepsilon}{2} = 0,05$.
2. Полагаем $x_1 = \frac{0+2-0,05}{2} = 0,975$; $x_2 = \frac{0+2+0,05}{2} = 1,025$. $f(x_1) = f(0,975) = 0,050625$; $f(x_2) = f(1,025) = -0,049375$. Получили, что $f(x_1) > f(x_2)$, значит, полагаем $a = x_1 = 0,975$. Проверяем условие окончания итераций: $|b - a| = |2 - 0,975| = 1,025 > \varepsilon$. Условие окончания итераций не выполнялось, значит, продолжаем решение, сужаем отрезок до $[0,975; 2]$ и переходим к шагу 2 алгоритма.
3. Полагаем $x_1 = \frac{0,975+2-0,05}{2} = 1,4625$; $x_2 = \frac{0,975+2+0,05}{2} = 1,5125$. $f(x_1) = f(1,4625) \approx -0,711$; $f(x_2) = f(1,5125) \approx -0,762$. Получили, что $f(x_1) > f(x_2)$, значит, полагаем $a = x_1 = 1,4625$. Проверяем условие окончания итераций: $|b - a| = |2 - 1,4625| = 0,5375 > \varepsilon$. Условие окончания итераций не выполнялось, значит, продолжаем решение, сужаем отрезок до $[1,4625; 2]$ и переходим к шагу 2 алгоритма.
4. Полагаем $x_1 = \frac{1,4625+2-0,05}{2} \approx 1,706$; $x_2 = \frac{1,4625+2+0,05}{2} \approx 1,756$. $f(x_1) = f(1,706) \approx -0,914$; $f(x_2) = f(1,756) \approx -0,941$. Получи-

ли, что $f(x_1) > f(x_2)$, значит, полагаем $a = x_1 = 1,70625$. Проверяем условие окончания итераций: $|b - a| = |2 - 1,706| = 0,293 > \varepsilon$. Условие окончания итераций не выполнялось, значит, продолжаем решение, сужаем отрезок до $[1,706; 2]$ и переходим к шагу 2 алгоритма.

5. Полагаем $x_1 = \frac{1,706+2-0,05}{2} \approx 1,828$; $x_2 = \frac{1,706+2+0,05}{2} \approx 1,878$.
 $f(x_1) = f(1,828) \approx -0,97$; $f(x_2) = f(1,878) \approx -0,985$. $f(x_1) > f(x_2)$, значит, полагаем $a = x_1 = 1,828$. Проверяем условие окончания итераций: $|b - a| = |2 - 1,828| = 0,172 > \varepsilon$. Условие окончания итераций не выполнялось, значит, продолжаем решение, сужаем отрезок до $[1,828; 2]$ и переходим к шагу 2 алгоритма.
 6. Полагаем $x_1 = \frac{1,828+2-0,05}{2} \approx 1,889$; $x_2 = \frac{1,828+2+0,05}{2} \approx 1,939$.
 $f(x_1) = f(1,889) \approx -0,988$; $f(x_2) = f(1,939) \approx -0,996$. $f(x_1) > f(x_2)$, значит, полагаем $a = x_1 = 1,889$. Проверяем условие окончания итераций: $|b - a| = |2 - 1,889| = 0,111 > \varepsilon$. Условие окончания итераций не выполнялось, значит, продолжаем решение, сужаем отрезок до $[1,889; 2]$ и переходим к шагу 2 алгоритма.
 7. Полагаем $x_1 = \frac{1,889+2-0,05}{2} \approx 1,919$; $x_2 = \frac{1,889+2+0,05}{2} \approx 1,969$.
 $f(x_1) = f(1,919) \approx -0,994$; $f(x_2) = f(1,969) \approx -0,999$. $f(x_1) > f(x_2)$, значит, полагаем $a = x_1 = 1,919$. Проверяем условие окончания итераций: $|b - a| = |2 - 1,919| = 0,08 < \varepsilon$. Условие окончания итераций выполнялось, значит, полагаем $x^* = a \approx 1,919$ и завершаем решение.
 8. Таким образом, получили точку минимума $x_{\min} \approx 1,919$ и наименьшее значение функции на рассматриваемом отрезке равно $-0,993$.
- Ответ: $x_{\min} \approx 1,919453125$, $f(x_{\min}) \approx -0,993512201$.

8.1.3. Метод золотого сечения

Метод золотого сечения улучшает метод дихотомии, так как в нём более рационально используются вычисленные значения функции. Переход к новому подотрезку осуществляется после вычисления только одного, а не двух, как в методе дихотомии, значения минимизируемой функции.

Золотым сечением отрезка называется такое его деление на две неравные части, что отношение длины отрезка к длине большего подотрезка равно отношению длины большего подотрезка к длине меньшего.

В данном методе на каждой итерации будем вычислять значение функции на отрезке и отбрасывать подотрезок, не содержащий точку минимума. Данный процесс повторяем до тех пор, пока не будет достигнута заданная точность ε .

Таким образом, метод сечения должен руководствоваться следующими принципами [13; 28]:

- для каждого итерационного шага вычисленное значение функции $f(x)$ может быть использовано повторно;
- для получения нового подотрезка меньшей длины необходимо использовать лишь одно новое значение $f(x)$;
- подотрезок, отбрасываемый на каждой итерации, должен иметь как можно меньшую длину.

Пусть рассматривается задача минимизации функции $f(x)$ на отрезке $[a, b]$. Введём новую переменную:

$$c = \frac{x - a}{b - a}.$$

Подставим в выражение для новой переменной вместо x переменные, соответствующие началу и концу отрезка (a и b). Получим отрезок $[0, 1]$, на котором расположены значения переменной c .

Расположим на этом отрезке две пробные точки c_1 и c_2 . Пусть точка c_2 имеет координату p , а точка c_1 имеет координату $1 - p$ (рис. 17).

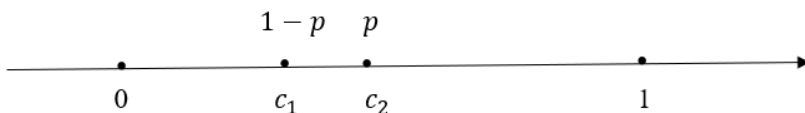


Рис. 17. Расположение пробных точек

Полагаем, что после сравнения значений функции в пробных точках отбрасываем правый конец отрезка и продолжаем работать с отрезком $[0, p]$. Для того чтобы пробная точка c_1 , расположенная на нём, также являлась пробной точкой c_2^1 для нового промежутка, необходимо, чтобы отношение длин отрезков $[0, 1]$ и $[0, c_2]$ совпадало с отношением длин новых отрезков $[0, p]$ и $[0, c_2^1]$. То есть должно выполняться следующее соотношение:

$$\frac{1}{p} = \frac{p}{1 - p}.$$

Из данной пропорции имеем, что $p = \frac{\sqrt{5}-1}{2} \approx 0,618034$. Получим, что $c_1 = 1 - p = \frac{3-\sqrt{5}}{2}$ и $c_2 = p = \frac{\sqrt{5}-1}{2}$.

Так как $c = \frac{x-a}{b-a}$, то получим выражение для x : $x = a + (b - a)u$. Таким образом, пробные точки для отрезка $[a, b]$ имеют вид

$$x_1 = a + (b - a) \frac{3-\sqrt{5}}{2} \text{ и } x_2 = a + (b - a) \frac{\sqrt{5}-1}{2}.$$

Сложив эти равенства, получим:

$$x_1 + x_2 = a + b.$$

На каждом последующем шаге одна пробная точка переходит из предыдущего промежутка, а вторая вычисляется из условия $x_1 + x_2 = a + b$.

Таким образом, с помощью пробных точек происходит последовательное уменьшение промежутка до тех пор, пока он не станет равен или меньше удвоенной точности ε . По достижении заданной точности полагаем точку минимума равной середине рассматриваемого промежутка.

Сформулируем алгоритм метода золотого сечения:

1. Вводим границы отрезка a, b и точность ε .
2. Полагаем $x_1 = a + (b - a) \frac{3-\sqrt{5}}{2}$ и $x_2 = a + b - x_1$; $y_1 = f(x_1)$ и $y_2 = f(x_2)$.
3. Проверяем выполнение условия окончания итераций – является ли длина очередного промежутка меньше или равной ε . Если нет, т. е. если $b - a > \varepsilon$, переходим к пункту 4. Иначе – к пункту 5.
4. Если $y_1 \leq y_2$, полагаем $b = x_2$, $x_2 = x_1$, $x_1 = a + b - x_2$, $y_2 = y_1$, $y_1 = f(x_1)$. Если $y_1 > y_2$, полагаем $a = x_1$, $x_1 = x_2$, $x_2 = a + b - x_1$, $y_1 = y_2$, $y_2 = f(x_2)$. Возвращаемся к пункту 3.
5. Полагаем $x_{\min} = \frac{a+b}{2}$, минимальное значение функции $f(x) = f(x_{\min})$.

Пример 19. Найти наименьшее значение функции $f(x) = x^2 - 4x + 3$ на отрезке $[0; 2]$ с точностью $\varepsilon = 0,1$.

1. $x_1 = 2 \cdot \frac{3-\sqrt{5}}{2} = 3 - \sqrt{5}$ и $x_2 = 2 - (3 - \sqrt{5}) = \sqrt{5} - 1$;
 $y_1 = f(x_1) = f(3 - \sqrt{5}) \approx 0,5376$ и $y_2 = f(x_2) \approx -0,4224$.

2. Проверим условие окончания итераций $(b - a > \varepsilon)$: $b - a = 2 > 0,1$. Так как $y_1 > y_2$, полагаем $a = 3 - \sqrt{5}$, $x_1 = \sqrt{5} - 1$, $x_2 = 3 - \sqrt{5} + 2 - (\sqrt{5} - 1) = 6 - 2\sqrt{5}$, $y_1 = -0,4224$, $y_2 = -0,82$.
 3. Проверим условие окончания итераций $b - a = 2 - (3 - \sqrt{5}) > 0,1$. Так как $y_1 > y_2$, полагаем $a = \sqrt{5} - 1$, $x_1 = 6 - 2\sqrt{5} \approx 1,53$, $x_2 = \sqrt{5} - 1 + 2 - (6 - 2\sqrt{5}) = -5 + 3\sqrt{5} \approx 1,71$, $y_1 = -0,82$, $y_2 = -0,92$.
 4. Проверим условие окончания итераций $b - a = 2 - (\sqrt{5} - 1) > 0,1$. Так как $y_1 > y_2$, полагаем $a = 1,53$, $x_1 = 1,71$, $x_2 = 1,53 + 2 - 1,71 = 1,82$, $y_1 = -0,92$, $y_2 = -0,97$.
 5. Проверим условие окончания итераций $b - a = 2 - 1,53 > 0,1$. Так как $y_1 > y_2$, полагаем $a = 1,71$, $x_1 = 1,82$, $x_2 = 1,71 + 2 - 1,82 = 1,89$, $y_1 = -0,97$, $y_2 = -0,99$.
 6. Проверим условие окончания итераций $b - a = 2 - 1,71 > 0,1$. Так как $y_1 > y_2$, полагаем $a = 1,82$, $x_1 = 1,89$, $x_2 = 1,82 + 2 - 1,89 = 1,93$, $y_1 = -0,99$, $y_2 = -0,995$.
 7. Проверим условие окончания итераций $b - a = 2 - 1,82 > 0,1$. Так как $y_1 > y_2$, полагаем $a = 1,89$, $x_1 = 1,93$, $x_2 = 1,89 + 2 - 1,93 = 1,96$, $y_1 = -0,995$, $y_2 = -0,998$.
 8. Проверим условие окончания итераций $b - a = 2 - 1,89 > 0,1$. Так как $y_1 > y_2$, полагаем $a = 1,93$, $x_1 = 1,96$, $x_2 = 1,93 + 2 - 1,96 = 1,97$, $y_1 = -0,998$, $y_2 = -0,999$.
 9. Проверим условие окончания итераций $b - a = 2 - 1,93 < 0,1$. Полагаем $x_{\min} = \frac{a+b}{2} = \frac{1,93+2}{2} = 1,965$, минимальное значение функции $f(x) = f(x_{\min}) = -1$.
- Ответ: $x_{\min} = 1,965$, $f(x_{\min}) = -1$.

Задания для самостоятельного решения

1. Найти наименьшее значение функции $f(x) = x^3 - 4x^2 + 6$ на отрезке $[0; 2]$ с точностью $\varepsilon = 0,1$ с помощью трёх изложенных выше методов.
2. Найти наименьшее значение функции $f(x) = 5x^3 - 4x + 1$ на отрезке $[0; 1]$ с точностью $\varepsilon = 0,1$ с помощью трёх изложенных выше методов.
3. Найти наименьшее значение функции $f(x) = x^2 - 2x + 1$ на отрезке $[0; 2]$ с точностью $\varepsilon = 0,1$ с помощью трёх изложенных выше методов.

4. Найти наименьшее значение функции $f(x) = 3\sin x - 2x + 1$ на отрезке $[0; 1]$ с точностью $\varepsilon = 0,1$ с помощью трёх изложенных выше методов.
5. Найти наименьшее значение функции $f(x) = \ln x - 4x^2 + 1$ на отрезке $[1; 2]$ с точностью $\varepsilon = 0,1$ с помощью трёх изложенных выше методов.

8.2. Градиентные методы решения оптимизационных задач

Градиентные методы решения оптимизационных задач относятся к поисковым методам, т. е. предполагают последовательное перемещение от точки к точке, которое заканчивается в точке, являющейся оптимальным решением задачи. Причём данное движение осуществляется в направлении градиента при решении задачи максимизации функции и в направлении, противоположном направлению градиента, при решении задачи минимизации.

Таким образом, общий план всех градиентных методов можно представить в виде следующего алгоритма [7; 10]:

- 1) Выбрать некоторую начальную точку, принадлежащую области определения функции.
- 2) Найти направление градиента в выбранной точке.
- 3) Переместиться из начальной точки в следующую, совершив шаг длиной α в направлении градиента, если решается задача максимизации целевой функции, или в направлении антиградиента, если решается задача минимизации целевой функции.
- 4) Минимум (максимум) целевой функции будет достигнут, когда градиент обратится в ноль. При решении практических задач условие остановки обычно задаётся в виде некоторого критерия, причём каждый подвид градиентных методов полагает свой критерий остановки.

Данный алгоритм применим ко всем подвидам градиентных методов. Отличие заключается лишь в выборе оптимального шага α .

Перейдём к рассмотрению различных типов градиентных методов.

8.2.1. Метод градиентного спуска

Данный метод относится к методам спуска, в которых основная идея заключается в последовательном выполнении итераций, каждая из которых приводит к уменьшению значения целевой функции:

$$f(x^{k+1}) < f(x^k), \forall k \geq 0.$$

Общий план методов спуска может быть представлен в виде следующего алгоритма [7; 10]:

- 1) Выбрать начальное приближение x^0 .
- 2) Пусть найдено приближение x^k в точке минимума, выбираем вектор $g^k \neq 0: \forall \alpha > 0$
- 3) $F_k(\alpha) = f(x^k + \alpha g^k) < f(x^k)$,
- 4) где α — достаточно малое число, а вектор g^k — направление спуска.
- 5) Определить длину шага $\alpha_k > 0$ по выбранному направлению g^k . Взять такое α_k , что $F_k(\alpha_k) = f(x^k + \alpha_k g^k) < f(x^k)$.
- 6) В качестве очередного приближения взять $x^{k+1} = x^k + \alpha_k g^k$.
- 7) Проверить выполнение критерия окончания итераций. Если критерий выполнен, то положить $x^* \approx x^{k+1}$. В противном случае вернуться к пункту 2.

В результате применения метода спуска формируется последовательность $\{x_k\}_{k=0}^{\infty}$, которую называют траекторией спуска.

В методах спуска одним из главных действий является выбор направления спуска. Для этого разложим функцию в ряд Тейлора в точке x^k : $f(x^k + \alpha g^k) = f(x^k) + \alpha(f'(x^k), g^k) + o(\alpha)$.

Если $(f'(x^k), g^k) < 0$, то $f(x^k + \alpha g^k) - f(x^k) < 0$. Значит, чтобы направление g^k было направлением спуска, необходимо и достаточно, чтобы g^k составляло острый угол с антиградиентом.

Методы, в которых в качестве направления спуска выбирается антиградиент, называются **градиентными**.

Таким образом, суть градиентного метода заключается в построении последовательности $\{x_k\}_{k=0}^{\infty}$ по правилу $x^{k+1} = x^k - \alpha_k f'(x^k)$, $\alpha_k > 0, k = 0, 1, \dots, N$.

Рассмотрим выбор шага спуска в методе градиентного спуска.

Определим на луче, направленном по антиградиенту, функцию $F_k(\alpha) = f(x^k - \alpha_k f'(x^k))$, α_k найдём из задачи одномерной минимизации $F_k(\alpha) \rightarrow \min_{\alpha \geq 0}$.

Квадратичная функция имеет вид $f(x) = \frac{1}{2} x^T A x + b^T x + c$. Рассмотрим применение метода наискорейшего градиентного спуска для квадратичных функций.

1. Вычислить градиентное направление спуска $g^k = -(Ax^k + b)$.
2. Определить шаг α_k из условия $F_k(\alpha) \rightarrow \min_{\alpha \geq 0}$:

$$F_k(\alpha) = f(x^k + \alpha g^k) = f(x^k) - \alpha \|g^k\|^2 + \frac{1}{2} \alpha^2 (Ag^k, g^k).$$

Единственная точка минимума может быть найдена из условия экстремума $F'_k(\alpha) = -\|g^k\|^2 + \alpha (Ag^k, g^k) = 0$. Получаем $\alpha_k = \frac{\|g^k\|^2}{(Ag^k, g^k)} > 0$.

3. Переходим к следующей итерации $x^{k+1} = x^k + \alpha_k g^k$. Проверяем выполнение критерия окончания итерации. Если критерий выполняется, то полагаем $x^* \approx x^{k+1}$, иначе переходим к пункту 1. В результате получили последовательность $\{x^k\}$ убывания функции $f(x)$.

В качестве критерия окончания итераций можно выбрать близость градиента к нулю, т. е. выполнение условия $\|f'(x^j)\| < \varepsilon$, $\|x^i - x^{i-1}\| < \varepsilon_1$, $\|f(x^i) - f(x^{i-1})\| < \varepsilon_2$. То есть если $\|Ax^k + b\| \rightarrow 0$ ($k \rightarrow \infty$), то $x^k \rightarrow x^*$: $f(x^k) \rightarrow f(x^*)$.

Рассмотрим примеры применения метода градиентного спуска для квадратичных функций.

Пример 20. Найти минимум функции $f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 4x_2$.

Для удобства запишем функцию в виде $f(x) = \frac{1}{2} x^T A x + b^T x + c$, откуда имеем $A = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$, $b = (-4; -4)$.

Преобразуем функцию $f(x_1, x_2) = (x_1^2 - 4x_1 + 4) + 2(x_2^2 - 2x_2 + 1) - 6$ и выберем начальное приближение. Пусть это будет точка $x^0(0; 0)$.

1. Градиентное направление спуска $g^0 = -(Ax^0 + b) = (4; 4)^T$.

2. Вычисляем шаг $\alpha_0 = \frac{\|g^0\|^2}{(Ag^0, g^0)} = \frac{32}{96} = \frac{1}{3}$.

3. Переходим к следующей итерации $x^1 = x^0 + \alpha_0 g^0 = (0; 0) + \frac{1}{3}(4; 4)^T = \left(\frac{4}{3}; \frac{4}{3}\right)$. Проверяем выполнение критерия окончания итерации: $\|Ax^1 + b\| = \left\| \left(-\frac{4}{3}; \frac{4}{3}\right) \right\| = \frac{\sqrt{32}}{3}$. Критерий окончания

итерации не выполнен, данная норма не стремится к нулю, значит, продолжаем решение и переходим к пункту 1 алгоритма.

$$4. \quad g^1 = -(Ax^1 + b) = \left(\frac{4}{3}; -\frac{4}{3}\right)^T.$$

$$5. \quad \alpha_1 = \frac{\|g^1\|^2}{(Ag^1, g^1)} = \frac{1}{3}.$$

$$6. \quad x^2 = x^1 + \alpha_1 g^1 = \left(\frac{4}{3}; \frac{4}{3}\right) - \left(\frac{4}{3}; -\frac{4}{3}\right)^T = \left(\frac{16}{9}; \frac{8}{9}\right)^T.$$

Проверяем выполнение критерия окончания итерации:

$\|Ax^2 + b\| = \frac{\sqrt{32}}{3^2}$. Критерий окончания итерации не выполнен, данная норма не стремится к нулю, значит, продолжаем решение и переходим к пункту 1 алгоритма.

$$7. \quad g^2 = -(Ax^2 + b) = \left(\frac{4}{9}; \frac{4}{9}\right)^T.$$

$$8. \quad \alpha_2 = \frac{\|g^2\|^2}{(Ag^2, g^2)} = \frac{1}{3}.$$

$$9. \quad x^3 = x^2 + \alpha_2 g^2 = \left(\frac{52}{27}; \frac{28}{27}\right)^T.$$

Проверяем выполнение критерия окончания итерации: $\|Ax^3 + b\| = \frac{\sqrt{32}}{3^3}$. Критерий окончания итерации не выполнен, данная норма не стремится к нулю, значит, продолжаем решение и переходим к пункту 1 алгоритма.

Продолжаем решение до тех пор, пока $\|Ax^k + b\|$ не будет близко к нулю. Заметим, что $\|Ax^k + b\| = \frac{\sqrt{32}}{3^k} \rightarrow 0, k \rightarrow \infty$.

В ходе трёх проделанных шагов данного метода была получена следующая траектория спуска: $\left\{(0; 0); \left(\frac{4}{3}; \frac{4}{3}\right); \left(\frac{16}{9}; \frac{8}{9}\right); \left(\frac{52}{27}; \frac{28}{27}\right); \dots\right\}$.

Изобразим графически полученный результат (рис. 18). Для построения линий уровня функции $f(x_1, x_2)$ воспользуемся системой компьютерной алгебры Maple.

Для этого сначала зададим функцию:

$$> f := x1^2 + 2*x2^2 - 4*x1 - 4*x2;$$

Изобразим линии уровня функции $f(x_1, x_2)$ с помощью команды *contourplot*:

$$> with(plots);$$

$$> contourplot(f(x1, x2), x1 = -5 .. 5, x2 = -5 .. 5, scaling = constrained, filled = false, coloring = [red, green], contours = 30);$$

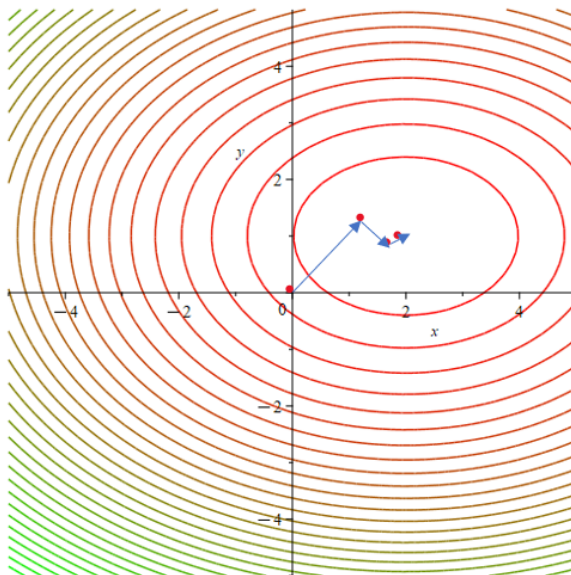


Рис. 18. Метод градиентного спуска в рассматриваемой задаче

Вспомним следующее представление функции $f(x_1, x_2) = (x_1^2 - 4x_1 + 4) + 2(x_2^2 - 2x_2 + 1) - 6 = (x - 2)^2 + 2(x - 1)^2 - 6$. Очевидно, что своё наименьшее значение функция принимает в точке $x^*(2; 1)$. Полученное направление как раз подтверждает этот очевидный результат.

Рассмотрим применение метода градиентного спуска для функций, не обязательно являющихся квадратичными, но являющихся выпуклыми.

Решение данных задач происходит по алгоритму, который выше был назван общим планом методов спуска. Остановимся на вопросе, связанном с выбором шага α . Так как при перемещении из точки x^0 в точку x^1 происходит изменение функции на Δf , то логично потребовать, чтобы значение Δf было наибольшим. Для этого применим необходимый признак экстремума функции Δf :

$$\frac{d\Delta f}{d\alpha} = 0 \Leftrightarrow \nabla f(x^0) \cdot \nabla f(x^1) = 0.$$

Пример 21. Найти максимум функции $f(x_1, x_2) = 2x_1 - 0,2x_1^2 + 2,4x_2 - 0,2x_2^2$ при ограничениях:

$$\begin{cases} x_1^2 + x_2^2 \leq 100, \\ x_1 + 2x_2 \geq 10, \\ x_2 \leq 8, \\ x \geq 0. \end{cases}$$

Определим начальное приближение. Возьмём, например, $x^0 = (5; 5)$. Эта точка является допустимой (т. е. удовлетворяет всем ограничениям системы), соответственно, возможен выбор данного начального приближения.

Определим координаты градиента $\frac{\partial f}{\partial x_1} = 2 - 0,4x_1$; $\frac{\partial f}{\partial x_2} = 2,4 - 0,4x_2$. Таким образом, градиент имеет вид $\nabla f(x^0) = (2 - 0,4x_1; 2,4 - 0,4x_2)$. В рассматриваемой точке $x^0 = (5; 5)$ координаты вектора градиента примут значения $\nabla f(x^0) = (0; 0,4)$. Градиент не является нулевым вектором, значит, рассматриваемая точка $x^0 = (5; 5)$ не является оптимальным решением.

Переместимся из точки $x^0 = (5; 5)$ в следующую точку $x^1 = x^0 + \alpha \cdot \nabla f(x^0) = (5; 5) + \alpha(0; 0,4) = (5; 5 + 0,4\alpha)$, т. е. $x^1 = (5; 5 + 0,4\alpha)$.

Определим значение шага α из условия $\frac{d\Delta f}{d\alpha} = 0 \Leftrightarrow \nabla f(x^0) \times \nabla f(x^1) = 0$. $\nabla f(x^1) = (2 - 0,4 \cdot 5; 2,4 - 0,4 \cdot (5 + 0,4\alpha)) = (0; 0,4 - 0,16\alpha)$.

$$\frac{d\Delta f}{d\alpha} = \nabla f(x^0) \cdot \nabla f(x^1) = (0; 0,4) \cdot (0; 0,4 - 0,16\alpha) = 0,4 \cdot (0,4 - 0,16\alpha).$$

Приравняем производную к нулю и получим $0,4 - 0,16\alpha = 0$, значит, $\alpha = 2,5$.

Определим вторую производную $\frac{d^2\Delta f}{d\alpha^2} = (0,4 \cdot (0,4 - 0,16\alpha))' = -0,064$. Видим, что $\frac{d^2\Delta f}{d\alpha^2} < 0$, значит, $\alpha = 2,5$ — точка максимума функции Δf .

При $\alpha = 2,5$ имеем $x^1 = (5; 5 + 0,4 \cdot 2,5) = (5; 6)$. Проверяем, что данная точка действительно удовлетворяет всем ограничениям исходной задачи, т. е. не выходит из области допустимых решений.

В рассматриваемой точке $x^1 = (5; 6)$ координаты вектора градиента примут значения $\nabla f(x^1) = (0; 0)$. Это означает, что функция $f(x)$ больше не может быть увеличена. Соответственно, точка $x^* = (5; 6)$

является оптимальным решением задачи (точкой максимума) и доставляет оптимальное значение целевой функции, равное $f(x^*) = 10 - 5 + 14,4 - 7,2 = 12,2$.

Ответ: $x^* = (5; 6)$ – точка максимума; $f(x^*) = 12,2$.

8.2.2. Метод Ньютона

Метод Ньютона основан на квадратичной аппроксимации оптимизируемой функции и использует её первые и вторые производные. То есть рассматриваемая функция должна быть дважды дифференцируема. Матрица вторых производных должна быть невырожденной.

Пусть x^k , k -е приближение к точке минимума. Разложим функцию в ряд Тейлора в рассматриваемой точке:

$$f(x) = f(x^k) + (f'(x^k) \cdot (x - x^k)) + \frac{1}{2}(f''(x^k) \cdot (x - x^k), (x - x^k)) + o(\|x - x^k\|^2),$$

где $f''(x^k)$ – матрица вторых производных.

Данное разложение используется для нахождения следующего приближения x^{k+1} . Это приближение находим как точку минимума функции $f(x)$, такую, что $f'(x^{k+1}) = 0$, или $f'(x^k) + f''(x^k) \cdot (x^{k+1} - x^k) = 0$, отсюда получим **формулу Ньютона**:

$$x^{k+1} = x^k - (f''(x^k))^{-1} \cdot f'(x^k).$$

В случае, если $f(x)$ – сильно выпуклая функция, а начальное приближение выбрано достаточно близко к точке минимума, то метод Ньютона будет сходиться с квадратичной скоростью сходимости, т. е. $\|x^{k+1} - x^k\| \leq C \cdot \|x^k - x^*\|^2$.

Таким образом, сформулируем алгоритм метода Ньютона [26; 38]:

1. Задаём начальное приближение x^0 , погрешности расчёта – $\varepsilon_1, \varepsilon_2, M$ – число итераций. Находим градиент функции в произвольной точке $\nabla f(x)$ и матрицу Гессе $H(x)$.
2. Полагаем $k = 0$.
3. Вычисляем $\nabla f(x^k)$.
4. Проверяем выполнение критерия окончания $\|\nabla f(x^k)\| < \varepsilon_1$:
 - 4.1. Если критерий выполнен, то полагаем $x^* = x^k$.
 - 4.2. Если критерий не выполнен, то переходим к шагу 5.
5. Проверяем выполнение неравенства $k \geq M$:
 - 5.1. Если критерий выполнен, то полагаем $x^* = x^k$.
 - 5.2. Если критерий не выполнен, то переходим к шагу 6.

6. Находим значение матрицы Гессе $H(x^k)$.
7. Находим значение матрицы Гессе $H^{-1}(x^k)$.
8. Проверяем выполнение условия $H^{-1}(x^k) > 0$:
 - 8.1. Если $H^{-1}(x^k) > 0$, то переходим к шагу 9.
 - 8.2. Если $H^{-1}(x^k) \leq 0$, то переходим к шагу 10, положив $d^k = -\nabla f(x^k)$.
9. Находим $d^k = -H^{-1}(x^k)\nabla f(x^k)$.
10. Вычисляем $x^{k+1} = x^k + t_k d^k$, положив $t_k = 1$, если $d^k = -H^{-1}(x^k)\nabla f(x^k)$, или выбираем t_k из условия $f(x^{k+1}) < f(x^k)$, если $d^k = -\nabla f(x^k)$.
11. Проверяем выполнение условий $\|x^{k+1} - x^k\| \leq \varepsilon_2, \|f(x^{k+1}) - f(x^k)\| \leq \varepsilon_2$:
 - 11.1. Если оба условия выполнены, то полагаем $x^* = x^{k+1}$.
 - 11.2. Если хотя бы одно из условий не выполнено, то полагаем $k = k + 1$ и переходим к шагу 3.

Рассмотрим следующий пример.

Пример 22. С помощью метода Ньютона найти точку минимума функции $f(x) = 9x_1^2 + x_2^2 - 18x_1 + 6x_2 + 18$ с точностью $\varepsilon = 10^{-3}$.

Начнём с вычисления вектора первых производных:

$$f'(x) = (18x_1 - 18; 2x_2 + 6).$$

Матрица вторых производных имеет вид

$$f''(x) = \begin{pmatrix} 18 & 0 \\ 0 & 2 \end{pmatrix}.$$

Эта матрица является положительно определённой по критерию Сильвестра.

$$\text{Значение матрицы Гессе } H^{-1}(x^k) = \begin{pmatrix} \frac{1}{18} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}.$$

Формула Ньютона для расчёта последующего приближения имеет вид

$$x^{k+1} = x^k - (f''(x^k))^{-1} \cdot f'(x^k),$$

$$x^{k+1} = x^k - \begin{pmatrix} \frac{1}{18} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} 18x_1^k - 18 \\ 2x_2^k + 6 \end{pmatrix}.$$

Решение задачи закончится в случае выполнения следующих условий:

$$\begin{aligned} |x_1^{k+1} - x_1^k| &\leq 10^{-3}, \\ |x_2^{k+1} - x_2^k| &\leq 10^{-3}, \\ \|f(x^{k+1}) - f(x^k)\| &\leq 10^{-3}, \\ |18x_1^{k+1} - 18| &\leq 10^{-4}, \\ |2x_2^{k+1} + 6| &\leq 10^{-4}. \end{aligned}$$

Выберем начальное приближение $x^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Рассчитаем первую итерацию по формуле Ньютона:

$$x^1 = x^0 - \begin{pmatrix} \frac{1}{18} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} 18 \cdot 0 - 18 \\ 2 \cdot 0 + 6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} \frac{1}{18} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} -18 \\ 6 \end{pmatrix} = \begin{pmatrix} 1 \\ -3 \end{pmatrix}.$$

Критерий остановки: $|x_1^1 - x_1^0| = |0 - 1| = 1 \geq 10^{-3}$. Условие окончания итераций не выполнено, значит, продолжаем и переходим ко второй итерации метода Ньютона:

$$x^2 = \begin{pmatrix} 1 \\ -3 \end{pmatrix} - \begin{pmatrix} \frac{1}{18} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ -3 \end{pmatrix}.$$

Получили, что второе приближение совпало с первым и $\begin{vmatrix} 18 \cdot 1 - 18 \\ 2 \cdot (-3) + 6 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \end{vmatrix}$, т. е. выполнены все условия окончания итераций.

Таким образом, решение задачи завершено и получена точка минимума $x^* = \begin{pmatrix} 1 \\ -3 \end{pmatrix}$. Оптимальное значение функция $f(x)$ достигает в точке x^* , и это значение равно $f(x^*) = 9 + 9 - 18 - 18 + 18 = 0$.

Ответ: оптимальное решение $x^* = \begin{pmatrix} 1 \\ -3 \end{pmatrix}$, оптимальное значение функции $f(x^*) = 0$.

В случае, если начальное приближение выбрано близко к точке минимума, то рассматриваемый метод имеет быструю сходимость. В противном случае решение задачи может быть достаточно трудоёмким процессом. Вычислительную сложность также составляет вычисление матрицы вторых производных.

Для ускорения работы метод Ньютона может быть модифицирован, например, следующим образом:

1. Добавить регулировку шага. То есть для вычисления последующего приближения можно модифицировать исходную формулу следующим образом:

$$x^{k+1} = x^k + \alpha_k (f''(x^k))^{-1} \cdot f(x^k).$$

Выбор множителя α_k производится, например, из условия минимизации функции вдоль выбранного направления.

2. Пересчитывать матрицу вторых производных не на каждом шаге.

8.2.3. Метод субградиентного спуска

Метод субградиентного спуска является расширением традиционного метода градиентного спуска [31]. Метод градиентного спуска может быть применим лишь к дифференцируемым функциям, а рассматриваемый в данном параграфе метод субградиентного спуска может применяться к функциям любого вида.

Определение 18. Вектор $g \in \mathbb{R}^n$ называется **субградиентом** функции $f(x)$ в точке $x_0 \in \mathbb{R}^n$, если

$$f(x) - f(x_0) \geq \langle g, x - x_0 \rangle, \forall x \in \mathbb{R}^n.$$

Определение 19. Множество субградиентов функции $f(x)$ в точке x_0 называется **субдифференциалом**. Данное множество ограничено и замкнуто.

Рассмотрим задачу вида

$$f(x) \rightarrow \min, x \in \mathbb{R}^n.$$

Функция $f(x)$ должна быть выпуклой и может быть недифференцируемой. Также должна существовать некоторая константа $M > 0$, что $\forall x \in \mathbb{R}^n$ и $\forall g(x) \in \partial f(x) \|g(x)\|_2 \leq M$.

Для дифференцируемых функций градиент совпадает с субградиентом.

Отличие данного метода от предыдущего заключается в том, что в данном методе в качестве направления спуска выбирается субградиент, т. е. переход от предыдущей точки к последующей также осуществляется по правилу

$$x^{k+1} = x^k - \alpha g^k,$$

где g^k — любой субградиент функции $f(x)$ в точке x^k , т. е. $g^k \in \partial f(x^k)$; α — размер шага.

Теорема 12. Пусть x^* – ближайшее решение к x^0 , $R_0 = \|x^0 - x^*\|_2$. Тогда через N итераций субградиентного спуска получим:

$$f(\bar{x}^N) - f(C) \leq \frac{MR_0}{\sqrt{N+1}},$$

где $\bar{x}^N = \frac{1}{N} \sum_{k=0}^N x^k$, константа M была описана ранее. А для достижения заданной точности ε следует произвести число итераций, равное $N = O\left(\frac{M^2 R_0^2}{\varepsilon^2}\right)$.

8.2.4. Метод проксимального градиента

Данный метод является обобщением проективных градиентных методов, которые используются при решении недифференцируемых задач выпуклого программирования. Данный метод применяется при решении задач с регуляризацией вида

$$F(x) = f(x) + R(x) \rightarrow \min_{x \in \mathbb{R}^n},$$

где $f(x): \mathbb{R}^n \rightarrow R - L$ – гладкая выпуклая функция; $R(x): \mathbb{R}^n \rightarrow R \cup \{+\infty\}$ – правильная выпуклая замкнутая функция. Также для $R(x)$ будем требовать возможность вычисления проксимального оператора.

Определение 20 [34]. Пусть функция $R(x): \mathbb{R}^n \rightarrow R \cup \{+\infty\}$ – правильная выпуклая замкнутая функция, тогда назовём проксимальным оператором отображение из $\mathbb{R}^n$ в $\mathbb{R}^n$:

$$prox_R(x) = \arg \min_{y \in \mathbb{R}^n} \left\{ R(y) + \frac{1}{2} \|y - x\|_2^2 \right\}.$$

Таким образом, можно сформулировать алгоритм метода проксимального градиента. Данный алгоритм будет похож на алгоритм метода градиентного спуска, только теперь для каждого шага добавится вычисление проксимального оператора [34].

1. Задаём размер шага $\gamma > 0$, стартовую точку x^0 и количество итераций N .
2. В цикле для k от 0 до $N - 1$:
 - 2.1. Вычисляем $\nabla f(x^k)$.
 - 2.2. Переходим в следующую точку $x^{k+1} = prox_{\gamma R}(x^k - \gamma \nabla f(x^k))$.

В результате выполнения данного алгоритма получим точку x^N , доставляющую минимум целевой функции.

Теорема 13. Пусть $f(x) - L$ – гладкая выпуклая функция, $R(x): \mathbb{R}^n \rightarrow R \cup \{+\infty\}$ – правильная выпуклая замкнутая функция, $\gamma = \frac{1}{L}$. Тогда для любого $N > 0$

$$f(x^N) - f(x^*) \leq \frac{L\|x^0 - x^*\|_2^2}{2N}.$$

То есть при $\gamma = \frac{1}{L}$ через $N = O\left(\frac{LR_0^2}{\varepsilon}\right)$ итераций, где $R_0 = \|x^0 - x^*\|_2$, выполняется $f(x^N) - f(x^*) \leq \varepsilon$.

Данный метод для выпуклых функций может быть ускорен.

В ускоренном градиентном методе (FISTA) алгоритм выглядит следующим образом:

1. Задаём стартовую точку x^0 и количество итераций N . Полагаем $y^0 = x^0, t_0 = 1$.
2. В цикле для k от 0 до $N - 1$:
 - 2.1. Вычисляем $\nabla f(y^k)$.
 - 2.2. Переходим в следующую точку $x^{k+1} = \text{prox}_{\frac{1}{L}R}\left(y^k - \frac{1}{L}\nabla f(y^k)\right)$.
 - 2.3. $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$.
 - 2.4. $y^{k+1} = x^{k+1} + \frac{t_k - 1}{t_{k+1}}(x^{k+1} - x^k)$.

В результате выполнения данного алгоритма получим точку x^N , доставляющую минимум целевой функции.

Теорема 14. Пусть $f(x) - L$ – гладкая выпуклая функция, $R(x): \mathbb{R}^n \rightarrow R \cup \{+\infty\}$ – правильная выпуклая замкнутая функция. Тогда для любого $N > 0$

$$f(x^N) - f(x^*) \leq \frac{2L\|x^0 - x^*\|_2^2}{(N + 1)^2}.$$

То есть через $N = O\left(\sqrt{\frac{LR_0^2}{\varepsilon}}\right)$ итераций, где $R_0 = \|x^0 - x^*\|_2$, выполняется $f(x^N) - f(x^*) \leq \varepsilon$.

8.2.5. Метод стохастического градиентного спуска (SGD)

Стохастическое программирование позволяет учитывать неопределённость в оптимизационных моделях. Это важно с прикладной

точки зрения, так как зачастую реальные прикладные задачи содержат некоторые неизвестные параметры.

Модели стохастического программирования используют знание распределений вероятностей для данных или их оценок. Цель заключается в поиске допустимого решения, максимизирующего математическое ожидание некоторой функции решений и случайных переменных [22; 34].

Метод стохастического градиентного спуска отличается от метода градиентного спуска использованием вместо градиента функции $g(x, \theta)$, такой, что её математическое ожидание по случайной величине θ $E_{\theta}(x, \theta) = \nabla f(x)$. То есть переход от предыдущей точки к последующей осуществляется по правилу

$$x^{k+1} = x^k - \alpha g(x^k, \theta_k),$$

где θ_k – некоторый случайный параметр, на который мы не влияем, но в среднем идём в направлении, обратном направлению вектора градиента.

Рассмотрим на конкретном примере оптимизационной задачи необходимость изучения данного метода и формулировку задачи стохастического градиентного спуска.

Пусть дана следующая оптимизационная задача:

$$\min_x \frac{1}{m} \sum_{i=1}^m f_i(x).$$

Градиент указанной выше целевой функции будет иметь вид

$$\nabla \sum_{i=1}^m f_i(x) = \sum_{i=1}^m \nabla f_i(x).$$

При применении градиентного спуска повторяем следующие шаги:

$$x^{(k)} = x^{(k-1)} - t_k \cdot \frac{1}{m} \sum_{i=1}^m \nabla f_i(x^{(k-1)}), k = 1, 2, 3, \dots$$

Если значение m будет довольно большим, к примеру порядка одного миллиона, то градиентный спуск окажется достаточно медленным. Вместо этого рассмотрим следующий алгоритм:

$$x^{(k)} = x^{(k-1)} - t_k \cdot \nabla f_{i_k}(x^{(k-1)}), k = 1, 2, 3, \dots,$$

где $i_k \in \{1, 2, \dots, m\}$ – некоторый индекс, выбранный на k -й итерации.

Данный алгоритм называется стохастическим градиентным спуском. Его основной плюс заключается в значительной экономии памяти при работе программы.

В общем случае существует два способа выбора индекса i_k на k -й итерации:

1. Правило рандомизации. На каждой итерации выбирать данный индекс случайным образом.
2. Правило цикла. Выбирать индекс i_k циклично по схеме $1, 2, \dots, m, 1, 2, \dots, m, \dots$

На практике правило рандомизации встречается гораздо чаще. Здесь мы имеем:

$$E(\nabla f_{i_k}(x)) = \nabla f(x).$$

Поэтому можно рассматривать метод стохастического градиентного спуска как несмещенную оценку градиента на каждом этапе.

Задания для самостоятельного решения

1. Найти минимум функции $f(x_1, x_2) = x_1^2 + 10x_2^2 - 4x_1 - 4x_2$.
2. Найти минимум функции $f(x_1, x_2) = 3x_1^2 + x_2^2 - x_1x_2 - 4x_1$.
3. Найти минимум функции $f(x_1, x_2) = 2x_1^2 + 2x_2^2 - x_1x_2$.
4. Найти минимум функции $f(x_1, x_2) = x_1^2 - x_2^2 + 5x_1 - 4$.
5. Найти минимум функции $f(x_1, x_2) = 2x_2^2 - x_1$.

8.3. Численные методы решения оптимизационных задач, позволяющие не использовать градиент

В данном параграфе рассмотрим актуальные численные методы решения задач оптимизации, не использующие градиент.

Рассмотрение данных методов связано с тем, что не все функции дифференцируемы, что, соответственно, препятствует нахождению градиента. Например, функция $f(x) = x \sin \frac{1}{x}$ не дифференцируема в нуле, и найти её наименьшее значение на отрезке, содержащем ноль, невозможно.

Поэтому если градиент по каким-либо причинам вычислить нельзя, задачу оптимизации можно решить, например, заменив градиент на численную оценку значения производной функции по некоторому случайным образом выбранному вектору.

То есть если случайным образом выбрать вектор $\bar{v}$ и зафиксировать шаг h , то для минимизации функции $f(\bar{x})$ будем использовать следующую численную оценку значения производной функции f по направлению $\bar{v}$:

$$\frac{f(\bar{x}) - f(\bar{x} + h\bar{v})}{h}.$$

То есть в данном подходе градиент функции не вычисляется явно ни на одном шаге, а сдвиг точки происходит в направлении $\bar{v}$, которое выбирается случайным образом. Получается, что в среднем смещение происходит по антиградиенту сглаженной функции.

Перейдём к рассмотрению ряда численных методов оптимизации, не использующих градиент.

8.3.1. Метод покоординатного спуска

В данном методе в качестве направления спуска выбираем направление одной из координатных осей. Задача поиска наименьшего значения функции нескольких переменных сводится к решению ряда задач одномерной минимизации.

Пусть имеем некоторое приближение x^k и необходимо решить задачу вида $f(x) \rightarrow \min$. Цикл с номером $k + 1$ имеет следующий вид [37]:

1. Спуск по координате x_1 . Для этого фиксируем все остальные координаты $x_2 = x_2^k, x_3 = x_3^k, \dots, x_n = x_n^k$ и решаем задачу минимизации функции одной переменной $f(x_1, x_2^k, x_3^k, \dots, x_n^k) \rightarrow \min$. Например, x_1^{k+1} можно найти как $x_1^{k+1} = x_1^k - \alpha_1^{k+1} \frac{\partial f}{\partial x_1}$, где α_1^{k+1} — некоторый шаг. То есть на этом шаге происходит сдвиг из точки x_1^k в точку x_1^{k+1} в сторону уменьшения значения функции.
2. Спуск по координате x_2 . Для этого фиксируем все остальные координаты $x_1 = x_1^{k+1}, x_3 = x_3^k, \dots, x_n = x_n^k$ и решаем задачу минимизации функции одной переменной $f(x_1^{k+1}, x_2, x_3^k, \dots, x_n^k) \rightarrow \min$. Например, x_2^{k+1} можно найти как $x_2^{k+1} = x_2^k - \alpha_2^{k+1} \frac{\partial f}{\partial x_2}$, где α_2^{k+1} — некоторый шаг. То есть на этом шаге происходит сдвиг из точки x_2^k в точку x_2^{k+1} в сторону уменьшения значения функции.

3. Аналогично производим спуск по остальным координатам.
4. На n -м шаге находим x_n^{k+1} путём решения задачи $f(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \dots, x_n^k) \rightarrow \min_{x_n}$.
5. В результате получаем последовательность точек $(x_1^{k+1}, x_2^k, x_3^k, \dots, x_n^k), (x_1^{k+1}, x_2^{k+1}, x_3^k, \dots, x_n^k), \dots, (x_1^{k+1}, x_2^{k+1}, x_3^k, \dots, x_n^{k+1})$. При переходе из текущей точки в последующую значения функции монотонно убывают.
6. Проверяем условие окончания цикла. Если $|f(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \dots, x_n^{k+1}) - f(x_1^k, x_2^k, x_3^k, \dots, x_n^k)| < \varepsilon$, где ε – некоторое небольшое, заранее заданное число, то конец, иначе – к пункту 1.

Начнём рассмотрение метода покоординатного спуска для минимизации квадратичной функции. Напомним, что квадратичная функция имеет вид $f(x) = \frac{1}{2}x^T Ax + b^T x + c$. Её частные производные по x_i можно представить в виде $\frac{\partial f}{\partial x_i} = \sum_{j=1}^n a_{ij} + b_i$.

Пример 23. Найти минимум функции $f(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2$.

Для удобства запишем функцию в виде $f(x) = \frac{1}{2}x^T Ax$, откуда имеем $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, b = (0; 0)$.

Преобразуем функцию $f(x_1, x_2) = x_1^2 + x_1 x_2 + x_2^2 = \frac{3}{4}(x_1 + x_2)^2 + \frac{1}{4}(x_1 - x_2)^2$. Линии уровня – эллипсы с центром в начале координат.

1. Выбираем начальное приближение, например $x^0(1; 1)$. Тогда $f(x^0) = 3$. Зафиксируем координату x_2 ($x_2 = 1$) и перейдём к решению следующей оптимизационной задачи:

$$f(x_1) = \frac{3}{4}(x_1 + 1)^2 + \frac{1}{4}(x_1 - 1)^2 \rightarrow \min,$$

$$f(x_1) = \frac{3}{4}(x_1^2 + 2x_1 + 1) + \frac{1}{4}(x_1^2 - 2x_1 + 1) = x_1^2 + x_1 + 1 \rightarrow \min,$$

$$\frac{\partial f}{\partial x_1} = 2x_1 + 1 = 0 \Rightarrow x_1 = -\frac{1}{2}.$$

Получили, что $x_1 = -\frac{1}{2}$ – точка минимума. $f(x_1) = \frac{3}{4}$. Проверим условие окончания цикла. Пусть $\varepsilon = 0,05$.

$$|f(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \dots, x_n^{k+1}) - f(x_1^k, x_2^k, x_3^k, \dots, x_n^k)| = \left| \frac{3}{4} - 3 \right| > \varepsilon.$$

Переходим к следующему шагу.

2. Зафиксируем координату $x_1 = -\frac{1}{2}$ и перейдём к решению следующей оптимизационной задачи:

$$f(x_2) = \frac{3}{4} \left(x_2 - \frac{1}{2} \right)^2 + \frac{1}{4} \left(x_2 + \frac{1}{2} \right)^2 \rightarrow \min,$$

$$\frac{\partial f}{\partial x_2} = 2x_2 - \frac{1}{2} = 0 \Rightarrow x_2 = \frac{1}{4}.$$

Получили, что $x_2 = \frac{1}{4}$ — точка минимума. $f(x_2) = \frac{3}{16}$. Проверим условие окончания цикла. $\left| \frac{3}{16} - \frac{3}{4} \right| > \varepsilon$. Переходим к следующему шагу.

3. Зафиксируем координату $x_2 = \frac{1}{4}$ и перейдём к решению следующей оптимизационной задачи:

$$f(x_1) = \frac{3}{4} \left(x_1 + \frac{1}{4} \right)^2 + \frac{1}{4} \left(x_1 - \frac{1}{4} \right)^2 \rightarrow \min,$$

$$\frac{\partial f}{\partial x_1} = 2x_1 + \frac{1}{4} = 0 \Rightarrow x_1 = -\frac{1}{8}.$$

Получили, что $x_1 = -\frac{1}{8}$ — точка минимума. $f(x_1) = \frac{3}{64}$. Проверим условие окончания цикла. $\left| \frac{3}{64} - \frac{3}{16} \right| > \varepsilon$. Переходим к следующему шагу.

4. Зафиксируем координату $x_1 = -\frac{1}{8}$ и перейдём к решению следующей оптимизационной задачи:

$$f(x_2) = \frac{3}{4} \left(x_2 - \frac{1}{8} \right)^2 + \frac{1}{4} \left(x_2 + \frac{1}{8} \right)^2 \rightarrow \min,$$

$$\frac{\partial f}{\partial x_2} = 2x_2 - \frac{1}{8} = 0 \Rightarrow x_2 = \frac{1}{16}.$$

Получили, что $x_2 = \frac{1}{16}$ — точка минимума. $f(x_2) = \frac{3}{256}$. Проверим условие окончания цикла. $\left| \frac{3}{256} - \frac{3}{64} \right| < \varepsilon$. Условие окончания цикла выполнено. Значит, решение оптимизационной задачи завершено и получено оптимальное решение $x^* \left(-\frac{1}{8}; \frac{1}{16} \right)$, $f(x^*) = \frac{3}{256}$.

В ходе решения задачи методом покоординатного спуска была получена следующая траектория спуска: $\{(1; 1); (-\frac{1}{2}; 1)(-\frac{1}{2}; \frac{1}{4}); (-\frac{1}{8}; \frac{1}{4}); (-\frac{1}{8}; \frac{1}{16})\}$. Изобразим графически полученный результат (рис. 19).

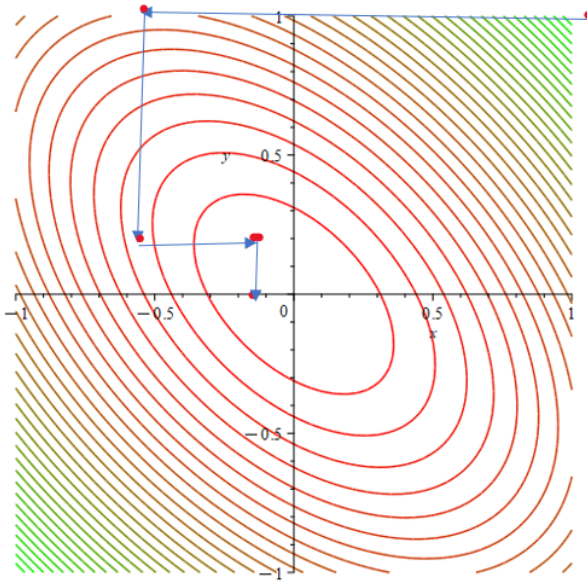


Рис. 19. Метод покоординатного спуска в рассматриваемой задаче

Для построения линий уровня функции $f(x_1, x_2)$ воспользуемся системой компьютерной алгебры Maple.

Для этого сначала зададим функцию:

```
> f := x1^2 + 2*x1*x2 + x2^2;
```

Изобразим линии уровня функции $f(x_1, x_2)$ с помощью команды *contourplot*:

```
> with(plots);
```

```
> contourplot(f(x1, x2), x1 = -1 .. 1, x2 = -1 .. 1, scaling = constrained,
filled = false, coloring = [red, green], contours = 30);
```

Ответ: $x^* \left(-\frac{1}{8}; \frac{1}{16}\right)$ — точка минимума; $f(x^*) = \frac{3}{256}$.

Рассмотрим следующий пример для случая, когда минимизируемая функция не является квадратичной.

Пример 24. Найти минимум функции $f(x) = x_1^3 - 2x_1x_2 + x_2^2 - 3x_1 - 2x_2$.

1. Выбираем начальное приближение, например $x^0(0; 0)$. Тогда $f(x^0) = 0$. Зафиксируем координату x_2 ($x_2 = 0$) и перейдём к решению следующей оптимизационной задачи:

$$\begin{aligned} f(x_1) &= x_1^3 - 3x_1 \rightarrow \min, \\ \frac{\partial f}{\partial x_1} &= 3x_1^2 - 3 = 0 \Rightarrow x_1 = \pm 1, \\ \frac{\partial^2 f}{\partial x_1^2} &= 6x_1. \end{aligned}$$

В точке $x_1 = 1$ вторая производная рассматриваемой функции неотрицательна, значит, в ней функция достигает своего минимума. $f(x_1) = -2$. Проверяем условие окончания цикла. Пусть $\varepsilon = 0,05$. $|f(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \dots, x_n^{k+1}) - f(x_1^k, x_2^k, x_3^k, \dots, x_n^k)| = |-2 - 0| > \varepsilon$. Переходим к следующему шагу.

2. Зафиксируем координату $x_1 = 1$ и перейдём к решению следующей оптимизационной задачи:

$$\begin{aligned} f(x_2) &= x_2^2 - 4x_2 - 2 \rightarrow \min, \\ \frac{\partial f}{\partial x_2} &= 2x_2 - 4 = 0 \Rightarrow x_2 = 2. \end{aligned}$$

Получили, что $x_2 = 2$ – точка минимума. $f(x_2) = -6$. Проверим условие окончания цикла. $|-6 + 2| > \varepsilon$. Переходим к следующему шагу.

3. Зафиксируем координату $x_2 = 2$ и перейдём к решению следующей оптимизационной задачи:

$$\begin{aligned} f(x_1) &= x_1^3 - 7x_1 \rightarrow \min, \\ \frac{\partial f}{\partial x_1} &= 3x_1^2 - 7 = 0 \Rightarrow x_1 = \pm \sqrt{\frac{7}{3}}, \\ \frac{\partial^2 f}{\partial x_1^2} &= 6x_1. \end{aligned}$$

Получили, что $x_1 = \sqrt{\frac{7}{3}}$ — точка минимума. $f(x_1) = \frac{7}{3}\sqrt{\frac{7}{3}} - 7\sqrt{\frac{7}{3}} = -\frac{14}{3}\sqrt{\frac{7}{3}}$. Проверяем условие окончания цикла. $\left| -\frac{14}{3}\sqrt{\frac{7}{3}} + 6 \right| > \varepsilon$. Переходим к следующему шагу.

4. Зафиксируем координату $x_1 = \sqrt{\frac{7}{3}}$ и перейдём к решению следующей оптимизационной задачи:

$$f(x_2) = x_2^2 - \left(2\sqrt{\frac{7}{3}} + 2 \right) x_2 - \frac{2}{3}\sqrt{\frac{7}{3}} \rightarrow \min,$$

$$\frac{\partial f}{\partial x_2} = 2x_2 - \left(2\sqrt{\frac{7}{3}} + 2 \right) = 0 \Rightarrow x_2 = \sqrt{\frac{7}{3}} + 1.$$

Получили, что $x_2 = \sqrt{\frac{7}{3}} + 1 \approx 2,53$ — точка минимума. $f(x_2) \approx -7,4$. Проверяем условие окончания цикла. $|-7,4 + 7,13| > \varepsilon$. Переходим к следующему шагу.

5. Зафиксируем координату $x_2 \approx 2,53$ и перейдём к решению следующей оптимизационной задачи:

$$f(x_1) = x_1^3 - 8,06x_1 + 1,3409 \rightarrow \min,$$

$$\frac{\partial f}{\partial x_1} = 3x_1^2 - 8,06 = 0 \Rightarrow x_1 \approx \pm 1,64,$$

$$\frac{\partial^2 f}{\partial x_1^2} = 6x_1.$$

Получили, что $x_1 \approx 1,64$ — точка минимума. $f(x_1) \approx -7,46$. Проверяем условие окончания цикла. $|-7,46 + 7,4| > \varepsilon$. Переходим к следующему шагу.

6. Зафиксируем координату $x_1 \approx 1,64$ и перейдём к решению следующей оптимизационной задачи:

$$f(x_2) = x_2^2 - 5,28x_2 - 0,509056 \rightarrow \min,$$

$$\frac{\partial f}{\partial x_2} = 2x_2 - 5,28 = 0 \Rightarrow x_2 = 2,64.$$

Получили, что $x_2 \approx 2,64$ – точка минимума. $f(x_2) \approx -7,48$.
 Проверяем условие окончания цикла. $|-7,48 + 7,46| < \varepsilon$. Условие
 окончания цикла выполнено. Значит, решение оптимизационной
 задачи завершено и приближённо получено оптимальное реше-
 ние $x^*(1,64; 2,64)$, $f(x^*) \approx -7,48$.

Данный пример был рассмотрен в параграфе 2 «Задача безусловной
 минимизации». Полученные решения совпали.

В ходе решения задачи методом покоординатного спуска была полу-
 чена следующая траектория спуска:

$$\left\{ (0; 0); (1; 0); (1; 2); \left(\sqrt{\frac{7}{3}}; 2\right); \left(\sqrt{\frac{7}{3}}; 2,53\right); (1,64; 2,53); (1,64; 2,64) \right\}.$$

Изобразим графически полученный результат (рис. 20). Для построе-
 ния линий уровня функции $f(x_1, x_2)$ воспользуемся системой компью-
 терной алгебры Maple.

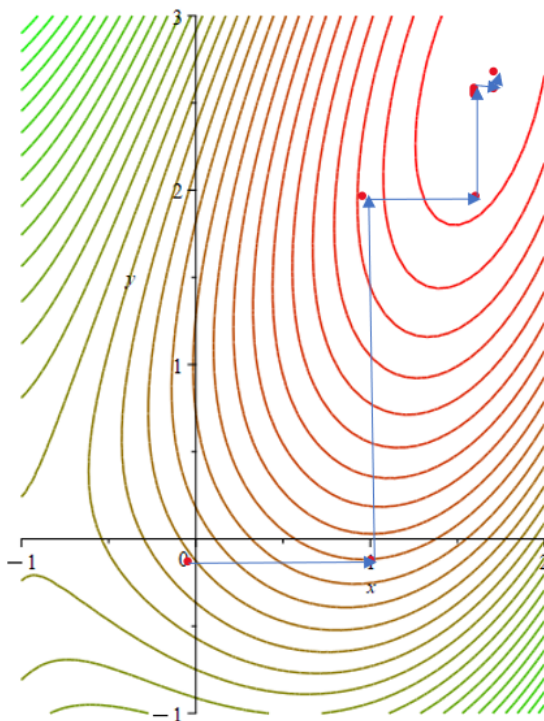


Рис. 20. Метод покоординатного спуска в рассматриваемой задаче

Для этого сначала зададим функцию:

```
> f := x1^3 - 2*x1*x2 + x2^2 - 3*x1 - 2*x2;
```

Изобразим линии уровня функции $f(x_1, x_2, x_3)$ с помощью команды *contourplot*:

```
> with(plots);
```

```
> contourplot(f(x1, x2), x1 = -1 .. 2, x2 = -1 .. 2, scaling = constrained,  
filled = false, coloring = [red, green], contours = 30);
```

Ответ: $x^*(1,64; 2,64)$ – точка минимума; $f(x^*) \approx 7,48$.

8.3.2. Метод штрафных и барьерных функций

Рассмотрим численные методы решения оптимизационных задач для целевой функции с рядом условий, составляющих ограничение задачи.

Рассмотрим оптимизационную задачу вида

$$\begin{cases} f(x) \rightarrow \min, \\ g_i(x) \leq 0, i = 1, \dots, m. \end{cases}$$

Основная идея **метода штрафных функций** [23; 25; 26] заключается в переходе от задачи условной оптимизации к задаче безусловной оптимизации путем изменения исходной целевой функции за счет ограничений.

В данном методе предлагается к целевой функции прибавлять некоторый штраф, который будет увеличиваться по мере удаления от точки минимума. Изначально данный штраф мал или равен нулю, если ограничения выполнены (или выполнены с точностью до ε).

От сформулированной выше задачи условной оптимизации перейдём к задаче безусловной оптимизации с добавленным штрафом:

$$F(x, r) = f(x) + \sum_{i=1}^m P_i(x, r),$$

где функция $P_i, i = 1, \dots, m$ отвечает за штраф для i -го ограничения, а коэффициент r регулирует размер штрафа.

Выделяют следующие виды штрафов:

1. Квадратичный штраф используется для учёта ограничений-равенств:

$$P_i(x, r^k) = r^k (g_i(x))^2.$$

Заметим, что штраф равен нулю в случае, если искомая точка удовлетворяет ограничениям задачи.

2. Штраф типа квадрата срезки:

$$P_i(x, r^k) = r^k (g_i^+(x))^2,$$

где $g_i^+(x)$ – срезка функции:

$$g_i^+(x) = \max\{0, g_i(x)\} = \begin{cases} g_i(x), & \text{если } g_i(x) > 0, \\ 0, & \text{если } g_i(x) \leq 0. \end{cases}$$

Заметим, что штраф равен нулю в случае, если искомая точка удовлетворяет ограничениям задачи. В случае их нарушения штраф будет увеличиваться пропорционально квадрату расстояний до допустимой области.

3. Бесконечный барьер:

$$P_i(x, r^k) = \begin{cases} +\infty \cdot g_i(x), & \text{если } g_i(x) > 0, \\ 0, & \text{если } g_i(x) \leq 0. \end{cases}$$

Сформулируем **алгоритм метода штрафных функций**:

1. Составить штрафную функцию и целевую функцию F . Задать размер штрафа r (например, 0,01), начальную точку отсчёта x^0 , рассчитать частные производные в этой точке по каждому аргументу. Начальная точка, как правило, берется такой, чтобы ограничения нарушались (поэтому данный метод иногда называют методом внешних точек).
2. Любым численным методом решения задачи безусловной минимизации найти минимум x^i функции F .
3. Если величина штрафа меньше заданной точности, то записать x^i в качестве ответа, иначе – увеличить штраф в несколько раз (оптимально увеличивать от 4 до 10 раз) и перейти к шагу 2.

Перейдём к рассмотрению **метода барьерных функций** [23–25].

Данный метод, как и метод штрафных функций, сводит задачу условной минимизации к задаче безусловной минимизации. Это происходит путём прибавления к целевой функции вспомогательной функции, сформированной из ограничений. Данная функция является барьером, не позволяющим выходить за пределы допустимой области. Поэтому начальная точка также должна выбираться из допустимой области. Как только значение приближается к барьеру, значение

целевой функции должно увеличиваться. У самого барьера оно, как правило, стремится к бесконечности.

Выделяют следующие виды барьерных функций:

1. Логарифмический штраф:

$$P_i(x, r^k) = -r^k \ln(g_i^+(x)).$$

Данная функция является неопределённой в недопустимых точках.

2. Обратная функция:

$$P_i(x, r^k) = -r^k \frac{1}{g_i(x)}.$$

Данная функция теряет свои свойства при нарушении ограничений.

При использовании барьерных функций необходимо предусматривать вычислительные процедуры, в которых в случае нарушения ограничений происходит уменьшение шага поиска, поскольку при решении именно этой подзадачи возникают затруднения.

Задания для самостоятельного решения

1. Найти минимум функции $f(x_1, x_2) = x_1^2 + 5x_2^2 - 2x_1 - 2x_2$.
2. Найти минимум функции $f(x_1, x_2) = 3x_1^2 + 2x_2^2 - 3x_1x_2 - 4x_1$.
3. Найти минимум функции $f(x_1, x_2) = x_1^2 + x_2^2 - x_1x_2$.
4. Найти минимум функции $f(x_1, x_2) = x_1^2 - x_2^2 + 3x_1 - 4$.
5. Найти минимум функции $f(x_1, x_2) = x_2^2 - x_1$.

8.4. Эвристические алгоритмы решения оптимизационных задач

Эвристические алгоритмы решения оптимизационных задач представляют собой алгоритмы принятия решений, использующие практический метод, который не является гарантированно точным, но является достаточным для решения поставленной задачи.

Данные алгоритмы широко применяются при решении задач, имеющих высокую вычислительную сложность. Эвристические алгоритмы позволяют значительно сократить время, затраченное на решение, но при этом полученное решение не будет наилучшим по сравнению с другими способами решения. Но всё же в большинстве случаев данные алгоритмы дают приемлемое и по качеству, и по времени решение.

8.4.1. Генетические алгоритмы

Генетические алгоритмы основаны на использовании случайного подбора, комбинировании и вариации искоемых параметров с применением механизмов, аналогичных естественному отбору в природе, для решения оптимизационных задач [27; 35; 44].

В начале работы алгоритма задаётся множество особей, представляющих собой текущую популяцию. Количество особей и сами особи зачастую генерируются случайным образом. Каждая из данных особей оценивается с помощью целевой функции (функции приспособленности).

Из рассматриваемой популяции выбираются особи, к которым применяются генетические операторы – скрещивание и мутация. Результатом применения данных операторов является новое поколение особей, которое также подвергается оценке с помощью функции приспособленности. Наилучшее решение для данной популяции запоминается, и алгоритм завершает работу, если в течение нескольких стадий не удалось улучшить решение или если проведено заданное число итераций.

Этапы генетического алгоритма:

1. Задать функцию приспособленности для особей популяции.
2. Создать начальную популяцию.
3. В популяции произвести размножение, мутирование, вычислить значение целевой функции для каждой особи. Сформировать новое поколение.
4. Проверить выполнение условия остановки. Если оно выполнено, то конец, иначе – вернуться к шагу 3.

Рассмотрим подробнее стадии генетического алгоритма.

Скрещивание может быть одноточечным или двухточечным. В случае простейшего одноточечного скрещивания в массиве каждой хромосомы случайным образом определяется точка и часть массива, лежащая левее этой точки, наследуется от родителя А, а которая правее – от родителя В.

В случае двухточечного скрещивания в массиве потомка выделяется уже не одна, а сразу две точки. В итоге получится три части массива.

Далее рассмотрим два способа формирования родительской пары – инбридинг и аутбридинг. В случае инбридинга первый родитель выбирается случайно, а второй родитель с большой вероятностью является мак-

симально близкой к первому родителю особью. В случае аутбридинга, наоборот, родители являются максимально далёкими особями.

В популяции могут происходить мутации. С помощью мутаций создаются такие цепочки генов, которых не было в первом поколении, но которые необходимы для получения правильного решения. Однако помимо положительных свойств мутации, могут возникать и негативные, связанные с бесполезными цепочками генов. Таким образом, необходимо грамотно работать с настройками мутации, чтобы она не затрудняла решение. Среди мутаций выделяют самые распространённые: инверсию гена; замену гена на случайный; замену гена на соседний. В начале каждой из перечисленных процедур хромосомы получают произвольные вероятности мутации. Если данная вероятность превысит 80 %, то хромосома мутирует.

Выделим два способа формирования нового поколения – элитный и отбор с вытеснением. В случае элитного отбора в новое поколение попадают наилучшие с точки зрения функции приспособленности особи – как предки, так и потомки. Алгоритмы с элитным отбором, как правило, имеют быструю сходимость. В случае отбора с вытеснением в новую популяцию попадают лишь потомки из множества потомков, отобранные с помощью функции приспособленности.

Рассмотрим применение генетических алгоритмов для решения оптимизационных задач.

Пример 25. Найти глобальный минимум функции

$$f(x) = \frac{1}{5}x^4 - \frac{11}{4}x^3 + 3x^2 + 12x$$

на отрезке $[0; 10]$.

Построим график данной функции с помощью системы компьютерной алгебры Maple. Для этого зададим функцию $f(x)$:

$$> f := 1/5*x^4 - 11/4*x^3 + 3*x^2 + 12*x$$

И построим её график на заданном промежутке с помощью команды *plot()*:

$$> \text{with}(plots);$$

$$> \text{plot}(f(x), x = -10 .. 20);$$

На этом отрезке функция принимает наименьшее значение в точке $x \approx 10$ (рис. 21).

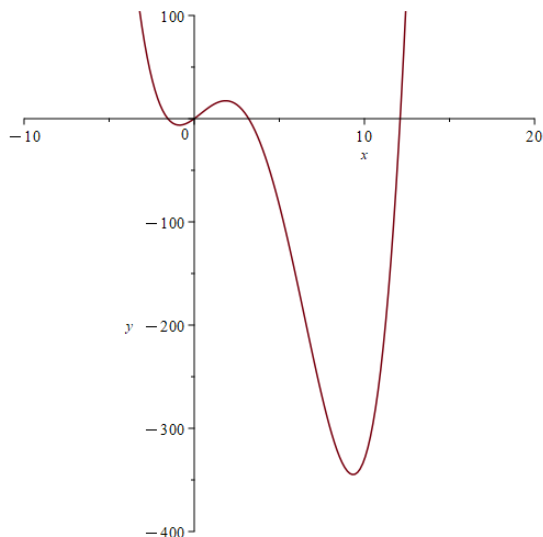


Рис. 21. График функции $f(x)$

Для простоты изложения положим, что x принимает лишь целые значения на рассматриваемом отрезке.

Выберем случайным образом несколько точек из рассматриваемого отрезка: $\{0, 2, 5, 7, 9\}$. Эти числа будем рассматривать в качестве пробных решений задачи.

Запишем выбранные числа в двоичной форме: $\{0000, 0010, 0101, 0111, 1001\}$.

В конкурентной борьбе выживает наиболее приспособленный. Приспособленность особи в рассматриваемой задаче определяется целевой функцией, т. е. меньшие значения целевой функции соответствуют более приспособленным особям. Поэтому вычислим значение функции в рассматриваемых пробных точках:

$$f(0) = 0;$$

$$f(2) = 17,2;$$

$$f(5) = -83,75;$$

$$f(7) = -232,05;$$

$$f(9) = -341,55.$$

Далее переходим к процессу размножения, т. е. на основе исходной популяции создаём новую таким образом, чтобы пробные решения в новой популяции располагались ближе к искомому глобальному минимуму. Для этого сформируем из исходной популяции брачные пары для скрещивания. Каждой особи исходной популяции ставим в соответствие целое число из диапазона от 1 до 5, т. е. пронумеруем их. Процесс размножения заключается в обмене участками хромосом между родителями.

Пусть i -я и j -я особи k -й популяции представлены в виде $p_i^k = (a_1, a_2, \dots, a_n) \in P^k$ и $p_j^k = (b_1, b_2, \dots, b_n) \in P^k$. При одноточечном скрещивании возникают два элемента популяции $k + 1$: $p_i^{k+1} = (a_1, \dots, a_c, b_{c+1}, \dots, b_n) \in P^{k+1}$, $p_j^{k+1} = (b_1, \dots, b_c, a_{c+1}, \dots, a_n) \in P^{k+1}$, где точка c выбирается случайным образом.

Изобразим в виде таблицы процесс одноточечного кроссинговера (табл. 6).

Таблица 6

Одноточечное скрещивание

№	Особь	Выбранный номер	Вторая особь-родитель	Точка кроссинговера	Особь-потомки
1	0000	1	0000	1	0000
2	0010	4	0111	2	0011 0110
3	0101	5	1001	3	0101 1001
4	0111	4	0111	2	0111
5	1001	2	0010	2	1010 0001

Далее следует рассмотреть возможные мутации, т. е. случайные изменения полученных в результате скрещивания хромосом. Пусть вероятность мутации равна 0,2. Для каждого потомка случайным образом сгенерируем число из отрезка $[0, 1]$. Если это число меньше, чем 0,2, то инвертируем случайный ген. Мутации способны улучшить или ухудшить приспособленность особи-потомка.

Оформим в виде таблицы возможные мутации у особей-потомков (табл. 7).

Таблица 7

Возможные мутации потомков

№	Особи-потомки	Случайное число	Выбранный для мутации ген	Потомок после мутации	Приспособленность до мутации	Приспособленность после мутации
1	0011	0,7	3	0011	4,95	4,95
2	0110	0,9	2	0110	-154,8	-154,8
3	0101	0,15	3	0111	-83,75	-232,05
4	1001	0,14	4	1000	-341,55	-300,8
5	1010	0,13	1	0010	-330	17,2
6	0001	0,5	2	0001	12,45	12,45

Теперь из пяти особей-родителей и шести полученных потомков сформируем новую популяцию, в которую отберём пять наиболее приспособленных представителей. В результате получим новое поколение, представителей которого изобразим в виде таблицы (табл. 8).

Таблица 8

Новая популяция

№	Новая популяция	Приспособленность особей в новой популяции
1	1001	-341,55
2	1000	-300,8
3	0111	-232,05
4	0110	-154,8
5	0101	-83,75

К получившейся популяции вновь можно применить кроссинговер, добавить мутацию и отобрать наиболее приспособленных особей в новое поколение. Таким образом, через несколько поколений будет получена популяция из наиболее приспособленных и похожих особей. Решением задачи будет являться наилучшее значение приспособленности.

Из таблицы приспособленности второй популяции можно сделать вывод, что наиболее приспособленная особь расположена на первой строке таблицы. Эта особь соответствует точке $x = 9$.

Программный код с подробными комментариями, реализующий описанный выше алгоритм, предложен в Приложении.

8.4.2. Метод муравьиных колоний

Идея муравьиного алгоритма заключается в имитации поведения муравьев, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи и возможностью менять траекторию своего движения в связи с изменениями в условиях окружающей среды. При движении муравей помечает свой путь феромонами, эта информация используется другими муравьями при поиске оптимального пути [33].

То есть один муравей может воспользоваться некоторым путём к цели. Далее некоторые муравьи могут найти более короткий путь и пометить его феромонами. В результате большинство муравьёв и воспользуются помеченным коротким путём.

Этапы муравьиных алгоритмов [15; 29; 30]:

1. Инициализация алгоритма. То есть создаётся некоторое множество муравьёв и случайным образом задаются необходимые алгоритму параметры.
2. Поиск решения, заключающийся в определении наилучшего пути к цели, который на каждом этапе определяется вероятностью выбора того или иного отрезка пути. Данная вероятность имеет вид

$$P_{ij,k}(t) = \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_{i,k}} (\tau_{il})^\alpha \cdot (\eta_{il})^\beta},$$

где τ_{ij} – уровень феромона на ребре (i, j) ; η_{ij} – величина, обратная расстоянию между узлами i и j ; α и β – параметры (константы).

При значении параметра $\alpha = 0$ муравьиный алгоритм превращается в обычный жадный алгоритм, так как перестаёт учитываться опыт других муравьёв. А при $\beta = 0$ учитывается лишь предыдущий опыт, расстояние же между узлами не имеет значения.

Уровень феромонов задаётся формулой

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \frac{Q}{L_k(t)},$$

где ρ – интенсивность испарения; $L_k(t)$ – цена текущего решения для муравья k ; Q – значение порядка цены оптимального решения. То есть дробь $\frac{Q}{L_k(t)}$ представляет собой феромон, откладываемый муравьём k .

Таким образом, муравьиные алгоритмы хорошо подходят для решения задач, связанных с нахождением оптимального пути, т. е. кратчайшего расстояния между некоторыми пунктами. Классическим примером задач такого рода является задача коммивояжёра. В задаче коммивояжёра есть ряд городов и известны расстояния между городами. Цель – посетить все города, вернуться в пункт отправления и выбрать для этой цели кратчайший маршрут. Рассмотрим пример, показывающий применение муравьиного алгоритма для решения задачи коммивояжёра.

Пример 26. Решить задачу коммивояжёра. Исходная матрица расстояний имеет вид

$$A = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 25 & 41 & 35 \\ 2 & 25 & - & 56 & 62 \\ 3 & 41 & 56 & - & 58 \\ 4 & 35 & 62 & 58 & - \end{pmatrix}.$$

Матрица расстояний представляет собой матрицу весов неориентированного графа. В дальнейшем будем рассматривать перемещения по вершинам графа.

Неориентированный граф, соответствующий заданной матрице расстояний, имеет следующий вид (рис. 22).

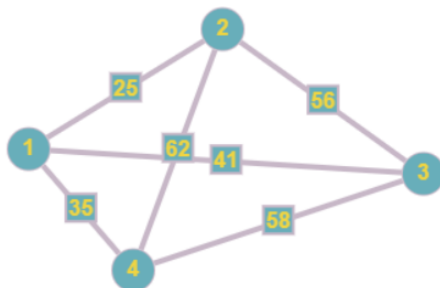


Рис. 22. Неориентированный граф, соответствующий матрице расстояний

Решение задачи будет заключаться в подсчёте вероятностей перехода из одного города в другой и построении маршрута, ориентирующегося на данные вероятности.

Напомним, что вероятность выбора того или иного отрезка пути рассчитывается по формуле

$$P_{ij,k}(t) = \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{l \in J_{i,k}} (\tau_{il})^\alpha \cdot (\eta_{il})^\beta},$$

где τ_{ij} – уровень феромона на ребре (i, j) ; η_{ij} – величина, обратная расстоянию между узлами i и j ; α и β – параметры (константы).

Поэтому начнём решение с задания параметров. Для простоты изложения положим, что нам одинаково важно учитывать и опыт других муравьёв, и расстояния между городами, т. е. $\alpha = 1$ и $\beta = 1$.

Далее переходим к τ_{ij} – уровню феромона на ребре (i, j) . Это значение можно генерировать или случайным образом, или для всех рёбер установить одно и то же числовое значение (т. е. не отдавать предпочтение определённым рёбрам). Зададим значение $\tau_{ij} = 1$. Начальные значения τ_{ij} можно представить в виде матрицы:

$$\tau(1) = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 1 & 1 & 1 \\ 2 & 1 & - & 1 & 1 \\ 3 & 1 & 1 & - & 1 \\ 4 & 1 & 1 & 1 & - \end{pmatrix}.$$

Пусть коммивояжёр выезжает из города с номером 1. Его цель – посетить все остальные города и вернуться в начало маршрута. При этом длина маршрута должна быть наименьшей. Начнём с расчёта вероятностей перемещения в тот или иной город:

$$P_{12}(1) = \frac{(\tau_{12})^1 \cdot (\eta_{12})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{1 \cdot \frac{1}{25}}{1 \cdot \frac{1}{25} + 1 \cdot \frac{1}{41} + 1 \cdot \frac{1}{35}} = 0,43;$$

$$P_{13}(1) = \frac{(\tau_{13})^1 \cdot (\eta_{13})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{1 \cdot \frac{1}{41}}{1 \cdot \frac{1}{25} + 1 \cdot \frac{1}{41} + 1 \cdot \frac{1}{35}} = 0,26;$$

$$P_{14}(1) = \frac{(\tau_{14})^1 \cdot (\eta_{14})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{1 \cdot \frac{1}{35}}{1 \cdot \frac{1}{25} + 1 \cdot \frac{1}{41} + 1 \cdot \frac{1}{35}} = 0,31.$$

Теперь следует задать правило, по которому будет выбираться тот или иной переход.

- Случайным образом сгенерируем число $a \in [0, 1]$.
- Если сгенерированное число a не превосходит $P_{12}(1)$, то следует выбрать маршрут $1 \rightarrow 2$.
- Если сгенерированное число a больше, чем $P_{12}(1)$, но меньше или равно, чем $P_{12}(1) + P_{13}(1)$, то выбираем маршрут $1 \rightarrow 3$.
- Если сгенерированное число a больше, чем $P_{12}(1) + P_{13}(1)$, но меньше или равно, чем $P_{12}(1) + P_{13}(1) + P_{14}(1)$, то выбираем маршрут $1 \rightarrow 4$.

Пусть, например, случайным образом было сгенерировано число $a = 0,2$. Согласно описанному выше правилу, получаем переход $1 \rightarrow 2$.

Таким образом, город 1 уже рассмотрен и можно вычеркнуть его из списка городов. Коммивояжёр находится в городе 2, и надо определить следующий для посещения город. Для этого рассчитаем вероятности перемещения из города 2 в остальные города (кроме первого, так как он уже был рассмотрен):

$$P_{23}(1) = \frac{(\tau_{23})^1 \cdot (\eta_{23})^1}{\tau_{23}\eta_{23} + \tau_{24}\eta_{24}} = \frac{1 \cdot \frac{1}{56}}{1 \cdot \frac{1}{56} + 1 \cdot \frac{1}{62}} = 0,53;$$

$$P_{24}(1) = \frac{(\tau_{24})^1 \cdot (\eta_{24})^1}{\tau_{23}\eta_{23} + \tau_{24}\eta_{24}} = \frac{1 \cdot \frac{1}{62}}{1 \cdot \frac{1}{56} + 1 \cdot \frac{1}{62}} = 0,47.$$

Аналогично прошлому шагу зададим правило, по которому будет выбираться тот или иной переход.

- Случайным образом сгенерируем число $a \in [0, 1]$.
- Если сгенерированное число a не превосходит $P_{23}(1)$, то следует выбрать маршрут $2 \rightarrow 3$.
- Если сгенерированное число a больше, чем $P_{23}(1)$, но меньше или равно, чем $P_{23}(1) + P_{24}(1)$, то выбираем маршрут $2 \rightarrow 4$.

Пусть, например, случайным образом было сгенерировано число $a = 0,72$. Согласно описанному выше правилу, получаем переход $2 \rightarrow 4$.

Таким образом, сформировался следующий маршрут: $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1$. Вычислим его длину:

$$L_0(1) = 25 + 62 + 58 + 41 = 186.$$

Пересчитаем уровень феромона на рёбрах. Для использованных выше рёбер $(1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 1)$ пересчитаем уровень феромона по формуле

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \frac{Q}{L_k(t)},$$

где ρ – интенсивность испарения; $L_k(t)$ – цена текущего решения для муравья k ; Q – значение порядка цены оптимального решения. Для этого зададим значения параметров $\rho = 0,2$, $Q = 200$. Получим:

$$\tau_{12}(2) = (1 - 0,2) \cdot 1 + \frac{200}{186} = 1,88;$$

$$\tau_{24}(2) = (1 - 0,2) \cdot 1 + \frac{200}{186} = 1,88;$$

$$\tau_{43}(2) = (1 - 0,2) \cdot 1 + \frac{200}{186} = 1,88;$$

$$\tau_{31}(2) = (1 - 0,2) \cdot 1 + \frac{200}{186} = 1,88.$$

Для не использованных на первом шаге рёбер значения феромона уменьшаются следующим образом:

$$\tau_{ij}(2) = (1 - \rho) \cdot \tau_{ij}(1).$$

Получим:

$$\tau_{14}(2) = \tau_{23}(2) = \tau_{32}(2) = \tau_{41}(2) = (1 - 0,2) \cdot 1 = 0,8.$$

Тогда матрица уровня феромона примет вид

$$\tau(2) = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 1,88 & 1,88 & 0,8 \\ 2 & 1,88 & - & 0,8 & 1,88 \\ 3 & 1,88 & 0,8 & - & 1,88 \\ 4 & 0,8 & 1,88 & 1,88 & - \end{pmatrix}.$$

На этом первый шаг цикла завершён. Переходим к следующему шагу.

Пусть коммивояжёр выезжает из города с номером 1. Его цель – посетить все остальные города и вернуться в начало маршрута. При этом длина маршрута должна быть наименьшей. Начнём с расчёта вероятностей перемещения в тот или иной город:

$$\begin{aligned} P_{12}(2) &= \frac{(\tau_{12})^1 \cdot (\eta_{12})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{1,88 \cdot \frac{1}{25}}{1,88 \cdot \frac{1}{25} + 1,88 \cdot \frac{1}{41} + 0,8 \cdot \frac{1}{35}}; \\ &= 0,52 \end{aligned}$$

$$P_{13}(2) = \frac{(\tau_{13})^1 \cdot (\eta_{13})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{1,88 \cdot \frac{1}{41}}{1,88 \cdot \frac{1}{25} + 1,88 \cdot \frac{1}{41} + 0,8 \cdot \frac{1}{35}};$$

$$= 0,32$$

$$P_{14}(2) = \frac{(\tau_{14})^1 \cdot (\eta_{14})^1}{\tau_{12}\eta_{12} + \tau_{13}\eta_{13} + \tau_{14}\eta_{14}} = \frac{0,8 \cdot \frac{1}{35}}{1,88 \cdot \frac{1}{25} + 1,88 \cdot \frac{1}{41} + 0,8 \cdot \frac{1}{35}} = 0,16$$

Правило выбора следующего перемещения коммивояжёра формулируем аналогично предыдущему шагу.

Пусть, например, случайным образом было сгенерировано число $a = 0,8$. Согласно описанному выше правилу, получаем переход $1 \rightarrow 3$.

Таким образом, город 1 уже рассмотрен и можно вычеркнуть его из списка городов. Коммивояжёр находится в городе 3, и надо определить следующий для посещения город. Для этого рассчитаем вероятности перемещения из города 3 в остальные города (кроме первого, так как он уже был рассмотрен):

$$P_{32}(2) = \frac{(\tau_{32})^1 \cdot (\eta_{32})^1}{\tau_{32}\eta_{32} + \tau_{34}\eta_{34}} = \frac{0,8 \cdot \frac{1}{56}}{0,8 \cdot \frac{1}{56} + 1,88 \cdot \frac{1}{58}} = 0,31;$$

$$P_{34}(2) = \frac{(\tau_{34})^1 \cdot (\eta_{34})^1}{\tau_{32}\eta_{32} + \tau_{34}\eta_{34}} = \frac{1,88 \cdot \frac{1}{58}}{0,8 \cdot \frac{1}{56} + 1,88 \cdot \frac{1}{58}} = 0,69.$$

Пусть случайным образом было сгенерировано число $a = 0,16$. Согласно описанному выше правилу, получаем переход $3 \rightarrow 2$.

Таким образом, сформировался следующий маршрут: $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1$. Вычислим его длину:

$$L_0(2) = 41 + 56 + 62 + 35 = 194.$$

Так как длина маршрута не уменьшилась, то этот маршрут не приведёт к улучшению решения, поэтому его запоминать не следует. Поэтому после данного прохода считаем все рёбра неиспользованными и уменьшаем уровень феромона на них:

$$\tau_{13}(3) = (1 - 0,2) \cdot 1,88 = 1,504;$$

$$\tau_{32}(3) = (1 - 0,2) \cdot 0,8 = 0,64;$$

$$\tau_{24}(3) = (1 - 0,2) \cdot 1,88 = 1,504;$$

$$\tau_{41}(3) = (1 - 0,2) \cdot 0,8 = 0,64;$$

$$\tau_{12}(3) = \tau_{21}(3) = (1 - 0,2) \cdot 1,88 = 1,504;$$

$$\tau_{34}(3) = \tau_{43}(3) = (1 - 0,2) \cdot 1,88 = 1,504.$$

Тогда матрица уровня феромона примет вид

$$\tau(3) = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 1,504 & 1,504 & 0,64 \\ 2 & 1,504 & - & 0,64 & 1,504 \\ 3 & 1,504 & 0,64 & - & 1,504 \\ 4 & 0,64 & 1,504 & 1,504 & - \end{pmatrix}.$$

На этом второй шаг цикла завершён. Так как длина маршрута не уменьшилась, то её значение не было сохранено.

Далее решение продолжается по следующей схеме: если для некоторого маршрута значение его длины становится меньше, чем значение длины маршрута для предыдущего шага, то следует сохранить данное значение, используя его при пересчёте уровня феромона. Если же значение длины маршрута не уменьшилось, то не следует сохранять длину данного пути – феромон для него просто испаряется с заданной интенсивностью, а не обновляется.

Решение продолжаем до тех пор, пока значение наикратчайшей длины маршрута на текущем шаге будет мало отличаться от значения на предыдущем шаге или до тех пор, пока не будет выполнено заданное заранее достаточно большое количество итераций.

Программный код с подробными комментариями, реализующий описанный выше алгоритм, предложен в Приложении.

8.4.3. Алгоритм имитации отжига

Алгоритм имитации отжига относится к эвристическим методам решения оптимизационных задач, так же как и описанные ранее генетический алгоритм и метод муравьиных колоний. Данный метод может быть применён для решения сложных оптимизационных задач с большим количеством данных. Решение, полученное с помощью данного метода, является приближённым, но достаточным для рассматриваемой задачи. Элемент случайности позволяет при рассмотрении различных областей поиска решения обнаруживать новые области, имеющие достаточно хорошие и точные решения.

Алгоритм имитации отжига основан на физических свойствах металлов. Установлено, что при использовании высоких температур металлы меняют состояние с твёрдого на жидкое, а частицы располагаются случайным образом. В случае если начальная температура достаточно высока, а время охлаждения металла достаточно длительно, то может быть достигнуто состояние с минимальной энергией. Если данные условия не выполняются, то металл переходит в метастабильное состояние, этот процесс называется закаливанием и предполагает резкое охлаждение металла. При процессе закаливания атомы выстраиваются несимметрично.

В случае медленного отжига атомы выстраиваются упорядоченно и симметрично, а материал становится твёрдым. Поэтому данный процесс был положен в основу некоторых алгоритмов оптимизации.

Таким образом, алгоритм имитации отжига основан на математическом описании физического процесса отжига металлов. Процесс отжига предполагает нагревание металла до высокой температуры и затем медленное охлаждение металла. При данном процессе внутренняя энергия металла распространяется равномерно, молекулы металла перемещаются и распространяются стабильно, снимается внутреннее напряжение в металле и устраняются межкристаллические дефекты.

Оптимизационный алгоритм, основанный на имитации отжига, использует аналогичные процессы. Используются процессы, аналогичные нагреванию и охлаждению металлов. Начало работы алгоритма связано с рассмотрением некоторого случайного решения. Далее начальное решение случайным образом изменяется. При этом оно может как улучшиться, так и ухудшиться. Ухудшение решения связано с функцией «охлаждения», которая отвечает за вероятность принятия данного решения. Функция «охлаждения» постепенно уменьшает вероятность ухудшения решения, а с точки зрения решения задачи нахождения глобального оптимума данная функция позволяет не остановить решение задачи на некотором найденном локальном оптимуме, а продолжить и найти глобальный.

Алгоритм имитации отжига включает в себя следующие основные моменты [14; 21; 36; 42]:

1. Использование случайности при переходе от текущего состояния к следующему и изменении состояния решения.
2. Использование худших решений.

3. Постепенное уменьшение вероятности худшего решения с помощью функции «охлаждения» (процесс имитирует медленное охлаждение металла).

Алгоритм начинает свою работу с некоторого достаточно высокого значения температуры, т. е. на данном этапе имитируется процесс нагрева металла. Высокая температура на этом шаге предполагает высокую энергию молекул, а худшие решения могут приниматься с некоторой вероятностью. Использование худших решений является достаточно важным процессом, ведь благодаря некоторому ухудшению найденного локального оптимума можно постепенно достичь глобального оптимума.

Рассмотрим основные формулы, используемые в данном алгоритме. Энергию можно вычислить по формуле

$$E = f(x).$$

То есть энергия соответствует значению функции приспособленности. Изменение энергии можно рассчитать по формуле

$$\Delta E = E_2 - E_1,$$

где E_2 — значение энергии в новом состоянии, E_1 — значение энергии в предыдущем состоянии.

Вероятность принятия худшего решения можно рассчитать по формуле

$$P = e^{\frac{-\Delta E}{T}},$$

где T — текущая температура.

Очевидно, что при увеличении изменения энергии вероятность принятия худшего решения уменьшается.

Формула изменения температуры:

$$T_2 = \alpha T_1,$$

где α — коэффициент изменения температуры, T_2 — новое значение температуры, T_1 — предыдущее значение температуры. Происходит процесс охлаждения металла, т. е. температура постепенно уменьшается.

Новое состояние можно сгенерировать, вычислив значение функции $G(x)$ генерации нового состояния, добавляющей некоторое случайное изменение.

Алгоритм имитации отжига можно разделить на следующие этапы:

1. Создание начального решения. Как отмечалось ранее, начальное решение выбираем случайным образом.
2. Оценка сформулированного решения. На данном шаге необходимо понять, является ли найденное решение целесообразным в контексте решения рассматриваемой проблемы.
3. Случайный поиск. Здесь текущее решение помещается в рабочее решение, которое в дальнейшем изменяется. Видоизменение решения зависит от решаемой задачи.
4. Оценка сформулированного решения. На этом шаге происходит оценка рабочего решения.
5. Использование критерия допуска. Имеется два решения – текущее и рабочее. Для каждого решения можно оценить его эффективность. Эффективность связана с энергией, т. е. чем меньше энергия, тем более эффективным является полученное решение. Далее текущее и рабочее решения сравниваются. Если рабочее решение лучше, чем текущее, т. е. имеет меньшую энергию, то рабочее решение перемещаем в текущее и переходим к следующему этапу. Если рабочее решение хуже текущего, то формулируется критерий допуска для ответа на вопрос о применении рабочего решения. Для этого рассчитывается вероятность допуска.
6. Понижение температуры. После ряда шагов, проведённых для некоторого значения температуры, происходит её понижение. Например, температуру можно понизить на некоторую небольшую константу.

Рассмотрим несколько классических примеров применения алгоритма имитации отжига.

1. Размещение N ферзей

Задача: разместить N ферзей на шахматной доске размера $N \times N$ таким образом, чтобы они не угрожали друг другу.

Решение данной задачи представляем в стандартном виде. Для простоты будем считать, что каждая строка и каждый столбец могут содержать лишь одного ферзя. Для хранения информации о положении ферзей создадим массив из N элементов, содержащий информацию об индексах строк, в которых расположены ферзи.

Изначально формируем произвольное решение, т. е. позволяем каждому ферзю занять некоторую строку, соответствующую столбцу. Далее начинается проход по столбцу и для каждого столбца выбирается произвольное число от 1 до N . Затем текущий столбец и произвольно выбранный столбец перемещаются. В конце данного алгоритма будет получено оптимальное решение, при этом конфликты среди ферзей как по горизонтали, так и по диагонали будут исключены.

Введём в рассмотрение основные понятия алгоритма имитации отжига в контексте рассматриваемой задачи. Так, полагаем, что энергия отвечает за количество конфликтов, возникающих в кодировке. Целью решения является определение такой ситуации, при которой энергия равна нулю. Значения температуры используем начиная с некоторого положительного числа и постепенно уменьшаем данное значение.

Для программной реализации сначала инициализируем начальные значения переменных, текущее решение. Далее текущее решение перемещается в рабочее и запускается алгоритм, работающий до тех пор, пока текущая температура не станет меньше или равна конечной температуре.

Для текущего значения температуры изменяется рабочее решение, рассчитывается его энергия, которая сравнивается с уже имеющимся значением для текущего решения. Если энергия рабочего решения лучше, чем энергия текущего решения, то принимается рабочее решение. Иначе оценивается вероятность допуска. Разница между рабочей и текущей энергиями задаёт такой параметр, как разность энергий. Если выполнено условие допуска, то принимается рабочее решение, а рабочее решение копируется в текущее.

Если рабочее решение не было принято, то текущее решение копируется поверх рабочего. На следующей итерации старое рабочее решение не сохраняется, а заменяется на текущее.

После выполнения некоторого количества итераций снижаем температуру путём умножения текущего значения на некоторый коэффициент, меньший единицы. В конце алгоритма выводится решение.

Опишем программную реализацию данной задачи на языке программирования Python.

Каждое состояние кодируем перестановкой чисел от 0 до $n - 1$, т. е. располагаем i -го ферзя на пересечении i -й строки и столбца с номером

p_i . Определим функцию f как функцию, отвечающую за число верно расставленных ферзей.

Подключаем библиотеки *random* и *math* для генерации случайных чисел и работы с математическими функциями соответственно:

```
import random
import math
```

Далее задаём размер доски (n) и количество итераций в алгоритме (k):

```
n = 100
k = 1000
```

Определяем функцию $f(p)$, отвечающую за количество успешно расставленных ферзей (s):

```
def f(p):
    s = 0
    for i in range(n):
        d = 1
        for j in range(i):
            if abs(i - j) == abs(p[i] - p[j]):
                d = 0
        s += d
    return s
```

Для генерации случайного действительного числа от 0 до 1 определяем функцию *rnd()*:

```
def rnd():
    return random.random()
```

Генерируем начальную перестановку (v) и помещаем значение функции f для данной перестановки в переменную *ans*, хранящую лучшее значение функции, т. е. будущий ответ:

```
v = list(range(n))
random.shuffle(v)
ans = f(v)
```

Далее полагаем значение температуры равным единице и постепенно в цикле уменьшаем это значение, создавая рабочую перестановку (u) и обменивая её с текущей (v):

```
t = 1
for i in range(k):
```

```

if ans >= n:
    break
t *= 0.99
u = v[:]
random_indices = random.sample(range(n), 2)
u[random_indices[0]], u[random_indices[1]] = u[random_indices[1]],
u[random_indices[0]]
val = f(u)
if val > ans or rnd() < math.exp((val - ans) / t):
    v = u
    ans = val

```

Выводим количество свободных диагоналей:

```
print(" ".join(str(x + 1) for x in v))
```

2. Задача коммивояжёра

Задача коммивояжёра рассматривалась ранее. Напомним её постановку. Пусть есть N городов, соединённых дорогами. Необходимо посетить все города по одному разу и вернуться в исходный город, при этом полученный маршрут должен быть кратчайшим.

Введём в рассмотрение некоторую функцию $f(x)$ и поставим цель – минимизировать данную функцию. Значение данной функции представляет собой длину пути, а аргумент – перестановку городов.

Начинаем с некоторого начального решения x_0 , т. е. с некоторой перестановки городов. Будем уменьшать длину пути. Вводим в рассмотрение t – некоторое число, соответствующее значению температуры, которое будет изменяться.

Алгоритм заключается в выполнении следующих шагов (до тех пор, пока не будет получено оптимальное решение или пока не закончится время, отведённое на выполнение программы):

1. Уменьшаем температуру. Для этого будем умножать текущее значение температуры на некоторый коэффициент γ , близкий к единице. Такой выбор коэффициента позволяет осуществить небольшое изменение температуры.
2. Выбираем некоторое случайное состояние u , в которое можно перейти из текущего состояния, выполнив некоторое небольшое преобразование.

3. Рассчитываем вероятность p , с которой присваиваем случайному состоянию y значение текущего состояния.

Эвристические моменты, присутствующие в данном алгоритме, связаны со следующим:

- в начале алгоритма, поскольку выбранное решение и так является плохим, то можно рассматривать высокую температуру и переход в худшее состояние;
- в случае если функция $f(x)$ изменяется не сильно, то алгоритм будет работать лучше. Например, в качестве небольшого изменения можно рассматривать изменение порядка двух городов, расположенных рядом в рассмотренном маршруте;
- если новое состояние хуже, чем предыдущее, то вероятность должна уменьшаться. Чем выше температура, тем выше вероятность. Если состояние y не хуже, т. е. $f(y) \leq f(x)$, то всё равно осуществляем переход в данное состояние. В противном случае производим переход в состояние y , при этом вероятность данного перехода $p = e^{\frac{f(x)-f(y)}{t}}$.

3. Нахождение глобального минимума функции

Для нахождения глобального минимума функции на вход подаются значения a и b – границы отрезка и t – начальное значение температуры. Далее выбираем некоторую точку x_0 – начальное приближение. Для получения нового состояния системы используем все описанные выше параметры для того, чтобы при высоких значениях параметра t использовать любую точку из рассматриваемого интервала, а в случае понижения температуры рассматривать точку, находящуюся в окрестности текущей.

Для нахождения вероятности принятия нового состояния системы используем изменение энергии. В случае если при переходе из старой точки в новую значение функции уменьшилось, то полагаем вероятность принятия данной точки равной 1. Иначе рассчитываем данную вероятность по формуле $p = e^{\frac{\Delta E}{t}}$. Принимаем новое состояние в том случае, если получившееся значение превосходит случайно сгенерированное число из промежутка от 0 до 1. Меняем температуру в случае удачно полученного нового состояния системы.

Задания для самостоятельного решения

1. Найти минимум функции $f(x_1, x_2) = 3x_1^2 + x_2^2 - x_1x_2 - 4x_1$.
2. Найти минимум функции $f(x_1, x_2) = x_1^2 + 10x_2^2 - 4x_1 - 4x_2$.
3. Найти минимум функции $f(x_1, x_2) = 2x_1^2 + 2x_2^2 - x_1x_2$.
4. Найти минимум функции $f(x_1, x_2) = x_1^2 - x_2^2 + 5x_1 - 4$.
5. Найти минимум функции $f(x_1, x_2) = 2x_2^2 - x_1$.
6. Решить задачу коммивояжёра. Матрица расстояний имеет вид

$$A = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 15 & 28 & 16 \\ 2 & 15 & - & 11 & 18 \\ 3 & 41 & 11 & - & 28 \\ 4 & 16 & 18 & 28 & - \end{pmatrix}.$$

7. Решить задачу коммивояжёра. Матрица расстояний имеет вид

$$A = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 32 & 18 & 42 \\ 2 & 32 & - & 35 & 20 \\ 3 & 18 & 35 & - & 48 \\ 4 & 42 & 20 & 48 & - \end{pmatrix}.$$

СПИСОК ЛИТЕРАТУРЫ

1. *Арушанян И. О.* Алгоритмы безусловной минимизации функций многих переменных: учеб. пособие для вузов. М., 2017. 44 с.
2. *Барон Л. А., Нуриев Н. К., Старыгина С. Д.* Численные методы для IT инженеров: учеб. пособие для вузов. Казань, 2012. 176 с.
3. *Бахтин В. И.* Метод множителей Лагранжа: метод. пособие для студентов спец. 1-31 03 01-03 «Математика (экономическая деятельность)» / В. И. Бахтин, И. А. Иванишко, А. В. Лебедев, О. И. Пиндрик. Минск: БГУ, 2012. 40 с.
4. *Буйначев С. К., Боклаг Н. Ю.* Основы программирования на языке Python: учеб. пособие. Екатеринбург: Изд-во Урал. ун-та, 2014. 91 с.
5. *Васильев А. Н.* Программирование на Python в примерах и задачах. М.: Эксмо, 2021. 616 с.
6. *Васильев Ф. П.* Методы оптимизации. М.: Факториал Пресс, 2002. 414 с.
7. *Веремеенко Т. В.* Высшая математика: учеб.-метод. пособие: в 4 ч. Ч. 4. Математическое программирование / авт.-сост. Т. В. Веремеенко; под ред. Л. Г. Третьяковой. 2-е изд., испр. Минск: ГИУСТ БГУ, 2010. 158 с.
8. *Власов П. А.* Методы вычислений [Электронный ресурс]. URL: <https://alexey-abakumkin.github.io/lectures-sem-2/CM/CM2.pdf> (дата обращения: 01.10.2023).
9. *Воронцова Е. А.* Выпуклая оптимизация: учеб. пособие / Е. А. Воронцова, Р. Ф. Хильдебранд, А. В. Гасников, Ф. С. Стонякин. М.: МФТИ, 2021. 364 с.
10. *Гасников А. В.* Современные численные методы оптимизации. Метод универсального градиентного спуска: учеб. пособие. Изд. 2-е, доп. М.: МФТИ, 2018. 286 с.
11. *Говорухин В. Н., Цибулин В. Г.* Введение в Maple. Математический пакет для всех. М.: Мир, 1997. 205 с.
12. *Городецкий С. Ю.* Лекции по методам нелинейной оптимизации: учеб. пособие. Н. Новгород: Нижегород. гос. ун-т, 2018. 219 с.
13. *Гребенникова И. В.* Методы оптимизации: учеб.-метод. пособие. Екатеринбург: Изд-во Урал. ун-та, 2017. 148 с.
14. *Землянский А. В., Штыков Г. А.* Программная реализация метода имитации отжига и анализ его эффективности // Вестник Пензенского государственного университета. 2017. № 4 (20). С. 76–81.
15. Как муравьи решают проблемы коммивояжеров [Электронный ресурс]. URL: <https://tproger.ru/articles/kak-muravi-reshajut-problemy-kommi-voja-zhgorov> (дата обращения: 10.06.2024).

16. Кушнирук Н. Н., Сельвинский В. В. Методы оптимизации. Практикум. Благовещенск: Изд-во АМГУ, 2011. 96 с.
17. Лекции. Теория оптимизации [Электронный ресурс]. URL: https://dep805.ru/education/portal/4/to/l1_4tochm.pdf (дата обращения: 22.06.2024).
18. Лутц М. Изучаем Python. 3-е изд. Пер. с англ. СПб.: Символ-Плюс, 2009. 848 с.
19. Максимей И. В., Серёгина В. С. Задачи и модели исследования операций: учеб.-метод. пособие. Ч. 2. Методы нелинейного и стохастического программирования. Гомель: БелГУТ, 1999. 103 с.
20. Манита Л. А. Условия оптимальности в конечномерных нелинейных задачах оптимизации: учеб. пособие. М.: Московский гос. ин-т электроники и математики, 2010. 84 с.
21. Метод отжига [Электронный ресурс]. URL: <https://algorithmica.org/ru/annealing> (дата обращения: 20.06.2024).
22. Метод стохастического градиента [Электронный ресурс]. URL: http://www.machinelearning.ru/wiki/index.php?title=Метод_стохастического_градиента (дата обращения: 22.06.2024).
23. Метод штрафных функций [Электронный ресурс]. URL: http://www.machinelearning.ru/wiki/index.php?title=Метод_штрафных_функций (дата обращения: 21.06.2024).
24. Методы оптимизации. Метод барьеров [Электронный ресурс]. URL: http://www.machinelearning.ru/wiki/images/9/91/MOM018_Practice3.pdf (дата обращения: 01.10.2023).
25. Методы штрафных и барьерных функций [Электронный ресурс]. URL: https://ozlib.com/885104/informatika/metody_shtrafnyh_barernyh_funktsiy (дата обращения: 21.06.2024).
26. Мижидон А. Д. Численные методы оптимизации: учеб. пособие. Улан-Удэ: Изд-во ВСГТУ, 2011. 156 с.
27. Мицель А. А. Эвристические методы оптимизации: учеб. пособие. Томск: Изд-во Томского гос. ун-та систем упр. и радиоэлектроники, 2022. 73 с.
28. Мицель А. А., Шелестов А. А., Романенко В. В. Методы оптимизации: учеб. пособие: в 2 ч. Ч. 1. Томск: Изд-во Томского гос. ун-та систем упр. и радиоэлектроники, 2020. 350 с.
29. Муравьиные алгоритмы [Электронный ресурс]. URL: http://www.machinelearning.ru/wiki/index.php?title=Муравьиные_алгоритмы (дата обращения: 22.06.2024).

30. Муравьиный алгоритм. Задача коммивояжёра [Электронный ресурс]. URL: <https://habr.com/ru/companies/timeweb/articles/754462/> (дата обращения: 10.06.2024).
31. *Нестеров Ю. Е.* Методы выпуклой оптимизации. М.: МЦНМО, 2010. 281 с.
32. Одномерная минимизация функций. Прямые методы [Электронный ресурс]. URL: <https://studfile.net/preview/4437257/page:6/> (дата обращения: 01.10.2023).
33. *Олейникова С. А.* Численные методы решения оптимизационных задач. Воронеж: Изд-во ВГТУ, 2021. 108 с.
34. Оптимизация в ML [Электронный ресурс]. URL: <https://education.yandex.ru/handbook/ml/article/optimizaciya-v-ml> (дата обращения: 21.06.2024).
35. *Панченко Т. В.* Генетические алгоритмы: учеб.-метод. пособие / под ред. Ю. Ю. Тарасевича. Астрахань: Изд. дом «Астраханский университет», 2007. 87 [3] с.
36. Популяционные алгоритмы оптимизации: Алгоритм имитации отжига (Simulated Annealing, SA) [Электронный ресурс]. URL: <https://www.mql5.com/ru/articles/13851?ysclid=lyuboiuxzs153813846> (дата обращения: 15.06.2024).
37. *Прокопенко Н. Ю.* Методы оптимизации: учеб. пособие. Н. Новгород: ННГАСУ, 2018. 118 с.
38. *Рейзлин В. И.* Численные методы оптимизации: учеб. пособие. Томск: Изд-во Нац. исслед. Томского политехн. ун-та, 2013. 105 с.
39. *Савотченко С. Е., Кузьмичёва Т. Г.* Методы решения математических задач в Maple: учеб. пособие. Белгород: Изд. Белаудит, 2001. 116 с.
40. *Самарский А. А.* Введение в численные методы. М.: Наука, 1982. 269 с.
41. *Самарский А. А., Гулин А. В.* Численные методы. М.: Наука, 1989. 432 с.
42. *Сидоров С. Г.* Алгоритм отжига. Методические указания / под ред. И. Ф. Ясинского. Иваново: УИУНЛ ИГЭУ, 2015. 20 с.
43. Угловые и главные миноры [Электронный ресурс]. URL: https://studopedia.su/10_143080_uglovie-i-glavnie-minori.html?ysclid=m03fmb73nd169899103 (дата обращения: 23.06.2024).
44. Genetic Algorithms – Quick Guide [Электронный ресурс]. URL: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_quick_guide.htm (дата обращения: 14.06.2024).

ПРИЛОЖЕНИЕ

1. Описание программного кода, реализующего **Пример 25**.

Цель: найти глобальный минимум функции

$$f(x) = \frac{1}{5}x^4 - \frac{11}{4}x^3 + 3x^2 + 12x$$

на отрезке $[0; 10]$.

Программа написана на языке программирования Python [4; 5; 18].

Начнём с подключения библиотек. Подключим библиотеку *numpy* для работы с массивами и матрицами, вычисления высокоуровневых математических функций при работе с данными; библиотеку *sympy* для символьных математических вычислений:

```
import numpy as np
import sympy as sp
```

Задаём переменную для использования в математических выражениях с помощью команды *symbols* из библиотеки *sympy*:

```
x = sp.symbols('x')
```

Задаём функцию приспособленности. В качестве функции приспособленности выступает функция $f(x)$ из условия примера:

```
f = 1/5 * x**4 - 11/4 * x**3 + 3*x**2 + 12*x
```

Далее формируем начальную популяцию. Пусть в начальную популяцию попадают 5 особей (точек). И случайным образом генерируем особей (точки) из промежутка от 0 до 11:

```
number_of_members = 5
population = np.random.randint(0, 11, number_of_members)
```

Выведем сформированную начальную популяцию:

```
print("Number of members = 5")
print("Initial population: ", population)
```

В результате получим сгенерированную случайным образом начальную популяцию особей, представляющую собой 5 случайно выбранных точек из промежутка от 0 до 11:

Initial population: [6 3 5 1 4]

Зададим количество итераций. Каждая итерация соответствует генерации новой популяции. Так как рассматривается учебный пример, то пусть это количество будет небольшим – например, 5 итераций. Для каждой итерации (для каждого поколения) выполняем действия, описанные ниже.

Все особи из популяции должны быть представлены в двоичной форме записи для возможности выполнения таких дальнейших биологических процессов, как мутация и кроссинговер. Поэтому далее переводим значения сгенерированных случайных чисел в двоичную систему счисления:

```
bin_population = np.empty(number_of_members, dtype='<U13')
for i in range(0, number_of_members):
    bin_population[i] = bin(int(population[i]))[2:].zfill(4)
```

В программе присутствуют случайные параметры. Первый случайный параметр *rand_parent* отвечает за номер партнёра, выбранного для скрещивания (в нашем примере решено выбирать для скрещивания именно случайного партнёра, хотя это необязательно). Второй случайный параметр *rand_cross* отвечает за точку кроссинговера – случайное число в диапазоне от 1 до 4:

```
rand_parent = np.random.randint(0, number_of_members,
                                number_of_members)
rand_cross = np.random.randint(0, 4, number_of_members)
```

Далее переходим к формированию массива потомков. Формирование потомков начинается с процесса скрещивания родительских особей. В массив потомков *children* добавляем генотип особи, полученной в результате кроссинговера. В качестве точки кроссинговера выбирается значение *rand_cross*, в качестве номера родителя, выбранного для скрещивания, выбирается значение *rand_parent*:

```
children = []
for i in range(0, number_of_members):
    if i != rand_parent[i]:
        children.append(bin_population[i][:rand_cross[i]+1]+
                        + bin_population[rand_parent[i]][rand_cross[i]+1:])

        children.append(bin_population[rand_parent[i]][:rand_cross[i]+1]+
                        + bin_population[i][rand_cross[i]+1:])
```

Следующий этап в формировании нового поколения связан с возможностью появления мутаций. Для этого случайным образом сгенерируем вероятность мутации *prob_mutation* – число из промежутка от 0 до 1:

```
prob_mutation = np.random.rand(1, 1)[0][0]
```

Каждому потомку случайным образом поставим в соответствие случайное число *mutation_number* из отрезка [0, 1]. Если это число меньше, чем *prob_mutation*, то инвертируем случайный ген. Номер гена для инвертирования представляет собой переменную *mutation_position*, сгенерированную случайным образом:

```
mutation_number = np.random.rand(1, len(children))[0]
mutation_position = np.random.randint(0, 4, len(children))
```

Далее осуществляем процесс мутации, где для каждого потомка сравниваем его *mutation_number* с *prob_mutation*, и если это число оказывается меньше, чем *prob_mutation*, то инвертируем ген с номером *mutation_position*:

```
for i in range(len(children)):
    if mutation_number[i] < prob_mutation:
        children[i] = children[i][:mutation_position[i]] \
+ str(int(not int(children[i][mutation_position[i]]))) \
+ children[i][mutation_position[i]+1:]
```

Теперь создаём общий массив, в который добавляем и родителей, и потомков:

```
new_population = np.unique(np.append(children, bin_population))
```

Нужно проверить, не попали ли в популяцию особи, не принадлежащие рассматриваемому отрезку [0; 10], и удалить их из *new_population* в случае попадания. Для этого перейдём обратно от двоичного представления чисел к десятичному:

```
for i in range(len(new_population)):
    new_population[i] = int(int("0b" + new_population[i].zfill(8), 2))
```

Удалим точки, большие, чем 10, из сформированного массива особей:

```
to_delete = []
for i in range(0, len(new_population)):
    if int(new_population[i]) > 10:
        to_delete.append(i)
```

Сформируем новую популяцию из особей, удовлетворяющих условию попадания в рассматриваемый интервал:

```
new_population = np.delete(new_population, to_delete)
```

Определим значения функции в точках:

```
f_new_population = np.empty(len(new_population))
```


И сформируем итоговую популяцию, в которую попадут лучшие особи из поколения родителей и поколения потомков:

```
for i in range(len(f_new_population)):
    f_new_population[i] = f.subs(x, new_population[i])
index_sorted_specs = np.argsort(f_new_population)
for i in range(number_of_members):
    population[i] = new_population[index_sorted_specs[i]]
print("Number of iteration = ", iteration+1, ": ", population)
```

Получили следующие результаты. Начальная популяция имеет вид
Initial population: [6 3 5 1 4]

В результате выполнения программы было сгенерировано 5 популяций потомков. На первой итерации получили популяцию:

Number of iteration = 1 : [6 5 4 0 3]

Обратим внимание, что в эту популяцию попали особи 0, 3, 4, которые далеки от оптимального решения. Таким образом, постепенно будем наблюдать улучшение решения, и в итоговой популяции должны быть особи, близкие к правой границе интервала.

Вторая популяция имеет вид

Number of iteration = 2 : [7 6 5 4 0]

Здесь также присутствуют особи, которые далеки от оптимального решения, – например, особь 0 всё ещё присутствует в сформированной популяции.

Третья популяция имеет вид

Number of iteration = 3 : [7 6 5 4 0]

Заметим, что третья популяция не отличается от второй. Такая ситуация вполне возможна в связи со случайностью многих используемых параметров.

Четвёртая популяция имеет вид

Number of iteration = 4 : [9 7 6 5 4]

На данном шаге происходит значительное улучшение решения, ведь из популяции ушла особь 0, которая очень далека от оптимального решения. Добавилась особь 9, которая очень близка к оптимальному решению. Популяция стала более приспособленной, чем популяция на предыдущем шаге. Особи из данной популяции близки к оптимальному решению.

Пятая популяция имеет вид

Number of iteration = 5 : [9 7 6 5 4]

Популяция осталась такой же, как и на предыдущем шаге.

Таким образом, популяция [9 7 6 5 4] является полученным решением. Так как в рассмотренном решении было использовано лишь 5 итераций, то полученное решение ещё далеко от оптимального. В случае увеличения количества итераций начиная уже с десятого поколения достигается оптимальное решение:

$$\text{Number of iteration} = 10 : [10\ 9\ 8\ 7\ 6]$$

Из сформированной на последнем шаге популяции выбираем лучшую особь. То есть особь, доставляющую целевой функции наименьшее значение. Она и является оптимальным решением. На этом отрезке функция принимает наименьшее значение в точке $x = 10$.

Конечно, в данной задаче можно получить более точное решение, если ввести в рассмотрение большее количество особей в популяции, использовать не только целые числа, но и десятичные дроби. Также возможно использовать различные варианты подбора второго родителя в пару для размножения, изменить стиль мутации и кроссинговера.

2. Описание программного кода, реализующего **Пример 26.**

Цель: решить задачу коммивояжёра. Исходная матрица расстояний имеет вид

$$A = \begin{pmatrix} & 1 & 2 & 3 & 4 \\ 1 & - & 25 & 41 & 35 \\ 2 & 25 & - & 56 & 62 \\ 3 & 41 & 56 & - & 58 \\ 4 & 35 & 62 & 58 & - \end{pmatrix}.$$

Программа написана на языке программирования Python.

Начнём с подключения библиотек. Подключим библиотеку *numpy* для работы с массивами и матрицами, вычисления высокоуровневых математических функций при работе с данными; библиотеку *sympy* для символьных математических вычислений:

```
import numpy as np
```

```
import sympy as sp
```

Задаём матрицу расстояний:

```
dist = np.array([[0, 25, 41, 35],
[25, 0, 56, 62],
[41, 56, 0, 58],
[35, 62, 58, 0]])
```

Начало маршрута – первый город. В Python нумерация начинается с нуля, поэтому

```
first_city = 0
```

По ходу выполнения программы будем добавлять рассмотренные города в массив *route*:

```
route = [first_city]
```

Далее формируем матрицу феромонов и заполняем её случайными числами:

```
pheromone_matrix = np.zeros((4, 4))
```

```
for i in range(4):
```

```
    for j in range(i+1, 4):
```

```
        pheromone_matrix[i][j] = pheromone_matrix[j][i] \
            = np.random.randint(0, 11, 1)[0]
```

Следует обойти и рассмотреть все города. Список номеров всех городов будем хранить в массиве *indexes*:

```
indexes = [0, 1, 2, 3]
```

Далее начинается основной цикл, в котором для каждого рассматриваемого города из списка всех городов будет выполняться ряд действий. Во-первых, вычисляются вероятности перехода в тот или иной город. Для этого сначала рассчитываем знаменатель в формуле вероятности:

```
div_prob = 0
```

```
for i in range(4):
```

```
    if i not in route:
```

```
        div_prob += pheromone_matrix[first_city][i] \
            / dist[first_city][i]
```

Далее создаём массив вероятностей, в котором храним значения вероятностей перехода в тот или иной город:

```
probs = np.zeros(4)
```

```
for i in range(4):
```

```
    if i not in route:
```

```
        probs[i] = pheromone_matrix[first_city][i] \
            / dist[first_city][i]
```

```
probs /= div_prob
```

Для задания правила, по которому будет выбираться тот или иной переход, сгенерируем случайным образом число *a* из отрезка от 0 до 1:

```
a = np.random.rand(1, 1)[0][0]
```

Далее ненулевые вероятности сравниваются со сгенерированным числом a по правилу, описанному в Примере 26:

- Если сгенерированное число a не превосходит $P_{12}(1)$, то следует выбрать маршрут $1 \rightarrow 2$.
- Если сгенерированное число a больше, чем $P_{12}(1)$, но меньше или равно, чем $P_{12}(1) + P_{13}(1)$, то выбираем маршрут $1 \rightarrow 3$.
- Если сгенерированное число a больше, чем $P_{12}(1) + P_{13}(1)$, но меньше или равно, чем $P_{12}(1) + P_{13}(1) + P_{14}(1)$, то выбираем маршрут $1 \rightarrow 4$.

Если условие справедливо, то цикл прерывается, найден нужный город:

```
prob_not_null = 0.  
for index in left_indexes:  
    if prob_not_null < a <= prob_not_null + probs[index]:  
        route.append(index)  
        break  
    prob_not_null += probs[index]
```

Теперь выходим из только что найденного города и повторяем все шаги:

```
first_city = route[-1]
```

После этого для завершения маршрута возвращаемся в первый город.

Построен некоторый маршрут:

```
route.append(0)  
print("route = ", route)
```

Вычисляем длину пройденного пути:

```
L = 0  
for i in range(len(route)-1):  
    L += dist[route[i]][route[i+1]]
```

Теперь следует изменить значения уровня феромона. Для этого случайным образом генерируем интенсивность испарения ρ :

```
rho = np.random.rand(1, 1)[0][0]
```

И изменяем значения элементов матрицы феромонов:

```
for i in range(4):  
    for j in range(i+1, 4):  
        pheromone_matrix[j][i] = pheromone_matrix[i][j] \  
            = pheromone_matrix[i][j] * (1 - rho)
```

Генерируем случайным образом некоторое большое число Q (значение порядка цены оптимального решения) в пределах от 90 до 150:

```
 $Q = np.random.randint(90, 150)$ 
```

Изменяем в матрице феромонов значения тех элементов, которые были задействованы в рассмотренном пути:

```
for i in range(len(route) - 1):  
    pheromone_matrix[route[i]][route[i+1]] = \  
    pheromone_matrix[route[i+1]][route[i]] = \  
    pheromone_matrix[route[i+1]][route[i]] + float(Q)/L
```

Выводим длину получившегося пути:

```
print("L = ", L)
```

Реализуем несколько таких проходов. Пусть количество полученных маршрутов равно 5. Для каждого маршрута выводим его номер, порядок городов, по которому совершается обход, и длину получившегося маршрута:

```
itnum = 1  
route = [0, 3, 1, 2, 0]  
L = 194
```

```
itnum = 2  
route = [0, 3, 2, 1, 0]  
L = 174
```

```
itnum = 3  
route = [0, 2, 1, 3, 0]  
L = 194
```

```
itnum = 4  
route = [0, 3, 2, 1, 0]  
L = 174
```

```
itnum = 5  
route = [0, 3, 2, 1, 0]  
L = 174
```

Таким образом, оптимальное значение длины пути, полученное в ходе выполнения данных итераций, равно 174. Так как в задаче было выбрано небольшое количество городов, то достаточно быстро удалось получить решение. В случае большего количества городов следует увеличить количество итераций для получения точного результата.

Учебное издание

ЗЕМЛЯКОВА Ирина Александровна

Нелинейное программирование в примерах и задачах

Редактор *И. А. Ивахина*

Корректор *И. А. Ивахина*

Компьютерная вёрстка *О. Ф. Жуковой*

Подписано в печать 16.12.2024 г.

Бумага офсетная. Формат 60×84 1/16. Тираж 30 экз.

Усл. печ. лист. 7,35. Уч.-изд. л. 5,5. Заказ № 9763.

Издательство Южного федерального университета.

Отпечатано в отделе полиграфической, корпоративной и сувенирной продукции
Издательско-полиграфического комплекса КИБИ МЕДИА ЦЕНТРА ЮФУ.

344090, г. Ростов-на-Дону, пр. Стачки, 200/1, тел (863) 243-41-66.



9785927547388