ГЛАВА 14

ОСНОВЫ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Настоящая глава посвящена общей задаче математического программирования и методам ее решения, т.е. задаче минимизации (максимизации) нелинейной функции при ограничениях, заданных нелинейными неравенствами и равенствами (задача условной оптимизации), а также задаче без каких-либо ограничений (задача безусловной оптимизации).

Необходимость изучения задач нелинейного программирования продиктована тем, что на практике встречаются ситуации по принятию решений, которые не всегда могут быть описаны линейными моделями, представляющими собой задачи линейного программирования, рассмотренные во второй главе первого тома "Исследование операций". Нелинейные зависимости возникают, например, в следующих задачах:

– составления смесей (приготовление бензиновых смесей);

– максимизации прибыли фирмы-монополиста (цена реализации не постоянна, а нелинейно зависит от объема реализации продукции фирмы);

– управления производственными процессами (регулирование температурного режима доменной печи металлургического завода);

– управления запасами;

– выбора портфеля ценных бумаг.

Таким образом, множество разнообразных ситуаций приводит к нелинейной формулировке ограничений и (или) целевой функции задач по нахождению оптимальных решений.

В общем случае задачи нелинейного программирования достаточно сложны и нет общего эффективного метода решения любой из них. При выборе метода решения задачи нелинейного программирования необходимо учитывать свой- ства целевой функции и ограничений, задающих область допустимых стратегий (решений).

14.1 Задача нелинейного программирования и ее геометрическая интерпретация. Классификация задач

Общая задача математического программирования формулируется следующим образом: найти вектор $x = (x_1, x_2, \ldots\ldots, x_n)$, дающий экстремум (минимум или максимум) функции $z = f(x_1, x_2, \ldots, x_n) = f(x)$ на заданном множестве K евклидова пространства R^n $(K \subseteq R^n)$. В дальнейшем будем предполагать, что ищется минимум функции $z = f(x_1, x_2, \ldots, x_n)$ и множество K определяется конечной системой неравенств и равенств:

$$\varphi_i(x) = \varphi_i(x_1, x_2, \ldots, x_n) \le 0 \quad (i = \overline{1, m});$$

$$\psi_r(x) = \psi_r(x_1, x_2, \ldots, x_n) = 0 \quad (r = \overline{1, l}),$$

где $f, \varphi_1, \varphi_2, \ldots, \varphi_m, \psi_1, \psi_2, \ldots, \psi_l$ – определенные на R^n функции. Функцию $z = f(x_1, x_2, \ldots, x_n)$ называют целевой функцией, или критерием оптимальности (эффективности). Каждое условие $\varphi_i(x_1, x_2, \ldots, x_n) \le 0$ $(i = \overline{1, m})$ называют ограничением-неравенством, а условия вида

$\psi_r(x_1, x_2, \ldots, x_n) = 0 \ (r = \overline{1, l})$ ограничением-равенством.

Вектор $x \in K$, т.е. удовлетворяющий всем ограничениям задачи, называют допустимым решением или допустимой точкой. Задача математического программирования заключается в нахождении такого допустимого решения x^*, для которого $f(x) \geq f(x^*)$ при всех допустимых x.

Решение x^* называется оптимальным решением задачи математического программирования. Если множество K совпадает с R^n $(K = R^n)$, т.е. нет никаких ограничений на x, то такая задача называется задачей безусловной оптимизации, если же $K \subset R^n$, – то задачей условной оптимизации.

Отметим, что часто ограничения задачи задают и в такой форме:

$$\varphi_i(x_1, x_2, \ldots, x_n) \leq b_i \quad (i = \overline{1, m});$$
$$\psi_r(x_1, x_2, \ldots, x_n) = d_r \quad (r = \overline{1, l}),$$

которая сводится к вышеуказанной переносом величин b_i и d_r в левую часть ограничений:

$$\varphi_i(x_1, x_2, \ldots, x_n) - b_i \leq 0 \quad (i = \overline{1, m}),$$
$$\psi_r(x_1, x_2, \ldots, x_n) - d_r = 0 \quad (r = \overline{1, l}).$$

Итак, задача математического программирования имеет вид

$$\min z = f(x) \tag{14.1}$$

при условиях

$$\begin{aligned} &\varphi_i(x) \leq 0 \quad (i = \overline{1, m}), \\ &\psi_r(x) = 0 \quad (r = \overline{1, l}). \end{aligned} \tag{14.2}$$

Если все функции $f(x), \varphi_i(x), \psi_r(x)$ линейные, то соответствующая задача математического программирования является задачей линейного программирования. Если хотя бы одна из указанных функций нелинейна, то соответствующую задачу называют задачей нелинейного программирования.

Отметим, что сформулированная задача является чрезвычайно общей. В терминах равенств формулируются, например, ограничения того, что переменная x_j должна принимать только значения 0 или 1. Действительно, равенство $x_j^2 - x_j = 0$ или $x_j(x_j - 1) = 0$ равносильно условию, что x_j может принимать только значения 0 или 1.

Отметим также, что любое неравенство $\varphi_i(x) \leq 0$ можно привести к равенству $\varphi_i(x) + y_i^2 = 0$ введением дополнительной переменной y_i. С другой стороны, любое равенство $\varphi_i(x) = 0$ можно заменить парой неравенств

$$-\varphi_i(x) \leq 0,$$
$$-\varphi_i(x) \leq 0.$$

Таким образом, систему ограничений задачи нелинейного программирования всегда можно свести к системе ограничений равенств или к системе ограничений неравенств, что используется в ряде теоретических исследований. Более того, систему ограничений-равенств $\varphi_i(x) = 0 \ (i = \overline{1, m})$ можно свести к единственному ограничению $\sum_{i=1}^{m} \varphi_i^2(x) = 0$, т.е. можно в принципе рассматривать задачу $\min z = f(x)$ только при одном ограничении $\sum_{i=1}^{m} \varphi_i^2(x) = 0$.

Однако эти преобразования не всегда упрощают решение задачи нелинейного программирования, а, наоборот, часто приводят к потере присущей исходной задаче специфики, что должно учитываться при поиске подходящего метода решения.

Как известно (см. главу 2 первого тома данного учебного пособия), в задачах линейного программирования точки экстремума целевой функции являются вершинами многогранника решений. В задачах же нелинейного программирования, как следует из рассматриваемых ниже примеров, они могут лежать внутри области K или на границе, но не обязательно в вершине области K, даже если область K является многогранником; более того сама область K может быть невыпуклой и даже несвязной, т.е. состоящей из двух или более непересекающихся частей. Все это приводит к большим трудностям при решении задач нелинейного программирования.

Рассмотрим примеры задач нелинейного программирования, иллюстрирующие указанные их особенности.

Пример 14.1. Найти минимальное и максимальное значение функции

$$z = (x_1 - 3)^2 + (x_2 - 5)^2$$

при ограничениях

$$\begin{cases} 2x_1 + 3x_2 \geq 12, \\ 2x_1 + 3x_2 \leq 18, \\ x_1 \geq 0,\ x_2 \geq 0. \end{cases}$$

Решение. Область допустимых решений K представляет собой многоугольник $ABCD$ (рис. 14.1).

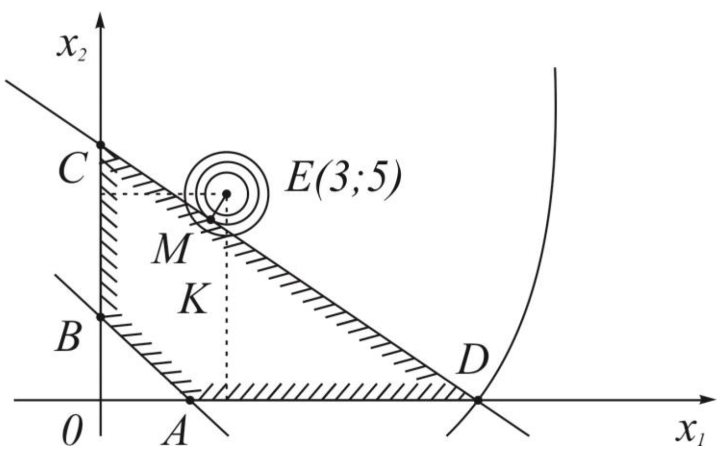

Рис. 14.1. Графический способ решения задачи

Рассмотрим линии уровня целевой функции. Полагая $z = C > 0$, получим уравнение $(x_1 - 3)^2 + (x_2 - 5)^2 = C,$ которое является уравнением окружности радиуса $R = \sqrt{C}$ с центром в точке $E(3;5)$. Например, при $C = 1$ получим окружность радиуса $R = 1$ с центром в точке E, а при $C = 4$ – окружность радиуса 2 с тем же центром E. Таким образом, линиями уровня целевой функции являются концентрические окружности с центром в точке $E(3;5)$. С увеличением $C = R^2$ значения функции z увеличиваются. Отсюда следует, что минимум функции z достигается в точке касания окружности с прямой CD, функция z имеет максимум в вершине $D(9;0)$. Очевидно, что

$$C_{\min} = z_{\min} = ME^2, \text{ и } |ME| = \frac{2 \cdot 3 + 3 \cdot 5 - 18}{\sqrt{2^2 + 3^2}} = \frac{3}{\sqrt{13}}.$$

Следовательно, $z_{\min} = \dfrac{9}{13}$. $z_{\max} = z(D) = (9-3)^2 + (0-5)^2 = 36 + 25 = 61.$

Пример 14.2. Пусть область допустимых значений останется прежней, а $z = (x_1 - 3)^2 + (x_2 - 2)^2$.

Найти минимальное и максимальное значение этой функции.

Решение. Минимальное значение функция z принимает в точке $E(3;2)$ (рис. 14.2) и $z_{\min} = 0$.

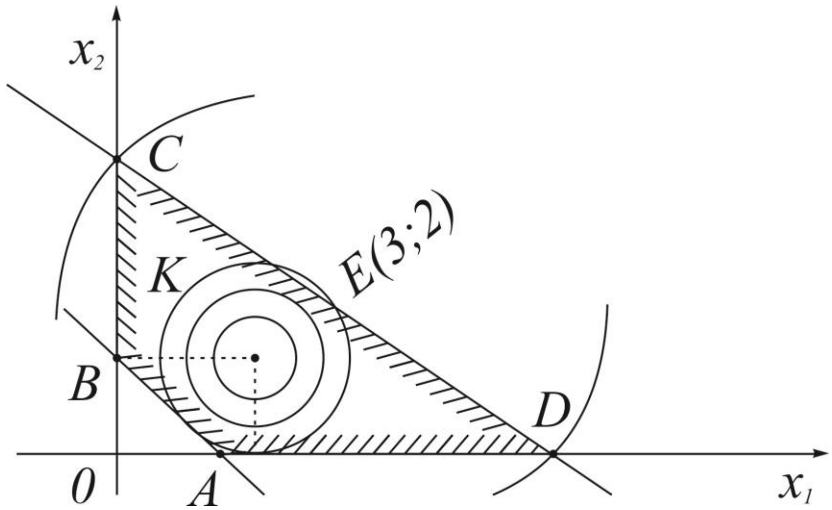

Рис. 14.2. Графический способ решения задачи

Функция z имеет два локальных максимума: в вершине $C(6;0)$ функция $z(C) = (0-3)^2 + (6-2)^2 = 25$ и в вершине $D(9;0)$ функция $z(D) = (9-3)^2 + (0-2)^2 = 40$, причем глобальный максимум достигается в точке D.

Пример 14.3. Найти минимальное и максимальное значение функции

$$z = x_1^2 + x_2^2$$

при ограничениях

$$\begin{cases} x_1 \cdot x_2 \le 4, \\ x_1 + x_2 \ge 5, \\ x_1 \le 8, \\ x_2 \le 7, \\ x_1 \ge 0,\ x_2 \ge 0. \end{cases}$$

Решение. Область допустимых значений изображена на рис. 14.3 и состоит из двух отдельных частей (область несвязная) $ABCD$ и $ELMN$.

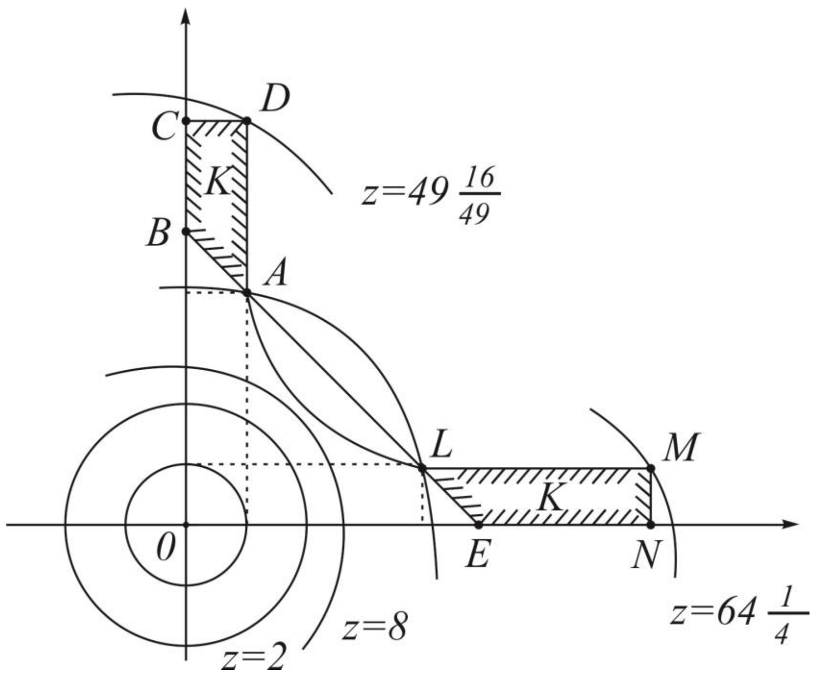

Рис. 14.3. Графическое решение задачи

Линиями уровня функции z являются концентрические окружности с центром в точке $O(0;0)$. Минимальное значение функции $z=17$ достигается в точках $A(1;4)$ и $L(4;1)$. Функция z имеет два локальных максимума: в точке $D\left(\frac{4}{7};7\right)$ функция $z(D)=49+\frac{16}{49}=49\frac{16}{49}$ и в точке $M\left(8;\frac{1}{2}\right)$ функция $z(M)=64+\frac{1}{4}=64\frac{1}{4}$. Точка $M\left(8;\frac{1}{2}\right)$ является точкой глобального максимума.

Приведенные примеры показывают, что в отличие от задачи линейного программирования задача нелинейного программирования может иметь несколько локальных экстремумов и область допустимых решений не только может быть невыпуклой, но даже и несвязной (см. пример 14.3).

Если область K допустимых решений определена (построена), то нахождение решения задачи нелинейного программирования сводится к определению такой точки этой области, через которую проходит гиперповерхность (при $n=2$ линия уровня) наинизшего (наивысшего) уровня: $f(x_1,x_2,\ldots,x_n)=C$. Эта точка может находиться как внутри области K, так и на границе допустимых ее значений. На основе изложенного процесс нахождения решения задачи нелинейного программирования с использованием ее геометрической интерпретации состоит из следующих этапов:

1. построение области K допустимых решений задачи, заданной системой ограничений;
2. построение гиперповерхностей (линий) уровня с уравнением $f(x_1,x_2,\ldots,x_n)=C$;
3. определение гиперповерхности (линии) наинизшего (наивысшего) уровня на множестве допустимых значений (функция может быть и неограниченной снизу или сверху);
4. нахождение координат точки, через которую проходит гиперповерхность наинизшего (наивысшего) уровня и вычисление значения целевой функции в этой точке.

Очевидно, что такой геометрический способ решения задачи осуществим лишь для очень простых задач и практически применим лишь для задач с двумя переменными.

Нахождение решения задач нелинейного программирования в общем случае связано с многими трудностями:

- во-первых, даже если нет дополнительных ограничений, то известные классические методы поиска экстремума теоретически позволяют найти только точки локального экстремума;
- во-вторых, наличие дополнительных ограничений приводит к задаче условного экстремума, и экстремум целевой функции может находиться как внутри, так и на границе области допустимых решений; поиск экстремума, находящегося на границе области допустимых решений требует специальных методов исследования;
- в-третьих, в реальных задачах нелинейного программирования целевая функция и функции, задающие ограничения, могут быть недифференцируемыми и тогда классические методы поиска экстремума неприменимы и, следовательно, нужны специальные методы недифференцируемой оптимизации.

Таким образом, рассматривая все множество разнообразных задач нелинейного программирования можно предложить их классификацию по ряду принципов.

1. По числу переменных:
   - одномерные задачи оптимизации;
   - многомерные задачи оптимизации.
2. По наличию ограничений:
   - задачи безусловной оптимизации (без ограничений);
   - задачи условной оптимизации (с ограничениями).

3. По наличию свойства дифференциальности функций, задающих задачу нелинейного программирования:
   - задачи оптимизации с дифференцируемыми функциями;
   - задачи оптимизации с недифференцируемыми функциями.

4. По специфическим свойствам целевой функции и области допустимых решений:
   - задачи выпуклого программирования;
   - задачи квадратичного программирования;
   - задачи сепарабельного программирования;
   - задачи дробного программирования.

При дальнейшем изложении в основном будем придерживаться этой классификации.

14.2 Примеры задач нелинейного программирования

Приведем примеры задач нахождения оптимальных решений, математические модели которых являются задачами нелинейного программирования.

1. Задача размещения складов

Пусть задано расположение n рынков и спрос на каждом из них (n – потребители и объемы их спроса). Спрос может быть удовлетворен из m складов с заданными емкостями.

Требуется разместить эти m складов так, чтобы общее расстояние, подсчитанное с весовыми коэффициентами, которые равны объемам товаров, перевезенных со складов на рынки (к потребителям), было минимальным.

Отметим, что на практике таким критерием оценки решения является показатель тонно-километры.

Построим модель такой задачи. Обозначим:

(x_i, y_i) – неизвестные координаты i-го склада ($i = \overline{1,m}$);

c_i – емкость i-го склада ($i = \overline{1,m}$);

(a_j, b_j) – известные координаты j-го рынка (потребителя) ($j = \overline{1,n}$);

r_j – известный объем спроса на j-м рынке ($j = \overline{1,n}$);

$d_{ij} = \sqrt{(x_i - a_j)^2 + (y_i - b_j)^2}$ – расстояние от i-го склада до j-го рынка ($i = \overline{1,m}; j = \overline{1,n}$);

x_{ij} – объем товара, поставляемого с i-го склада j-му рынку ($i = \overline{1,m}; j = \overline{1,n}$). Тогда задача размещения m складов и определения объемов перевозок со складов на рынки формулируется математически следующим образом:

$$\min z = \sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} x_{ij}$$

при условиях

$$\sum_{j=1}^{n} x_{ij} \le c_i \quad (i = \overline{1,m}),$$

$$\sum_{i=1}^{m} x_{ij} = r_j \quad (j = \overline{1,n}),$$

$$x_{ij} \ge 0 \quad (i = \overline{1,m}; j = \overline{1,n}).$$

В ней переменными величинами являются координаты (x_i, y_i) складов и объемы перевозок x_{ij}. Полученная задача является задачей нелинейного программирования, так как целевая функция z является нелинейной. Очевидно, что если расположение складов известно, то величины d_{ij} известны и искомыми величинами останутся только объемы перевозок x_{ij}, и в этом случае мы придем к обычной транспортной задаче, рассмотренной нами в третьей главе.

2. Задача планирования производства

Фирма на планируемый период должна изготовить a единиц продукции, используя n технологических способов. При производстве x_j единиц продукции j-м технологическим способом затраты равны $c_j x_j + d_j x_j^2$ $(c_j > 0; d_j > 0)$. Объемы используемых ресурсов i-го вида $(i = \overline{1,m})$ и нормы их использования j-м технологическим способом соответственно равны b_i и a_{ij}.

Определить, сколько изделий надо изготовить каждым способом, чтобы общие затраты на производство были минимальными.

Математическая модель этой задачи сводится к задаче нелинейного (квадратичного) программирования

$$\min z = \sum_{j=1}^{n}(c_j x_j + d_j x_j^2)$$

при ограничениях

$$\sum_{j=1}^{n} x_j = a,$$

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

$$x_j \ge 0 \quad (j = \overline{1,n}).$$

3. Задача выбора портфеля ценных бумаг

Специалисты, работающие в банках и страховых компаниях, уделяют немало внимания разработке математических моделей, помогающих определить наилучший набор ценных бумаг (акций, облигаций и т.п.) на выделенную сумму денег. Этими моделями учитывается оценка ожидаемой прибыли от приобретения предлагаемого набора ценных бумаг, а также связанная с этим оценка риска.

Приведем простой вариант такой модели, являющейся задачей нелинейного программирования.

Пусть x_j – доля средств, выделяемых для приобретения j-го вида ценных бумаг ($j = \overline{1,n}$). Предположим, что прибыль к концу планового периода на каждую вложенную денежную единицу характеризуется двумя показателями:

1) a_j – фактическая прибыль, которая представляет собой случайную величину ($j = \overline{1,n}$);

2) α_j – ожидаемая прибыль ($j = \overline{1,n}$).

Пусть оптимальная прибыль на одну вложенную денежную единицу для всего набора ценных бумаг должна быть не ниже заданной величины b. Тогда ограничениями этой задачи будут служить условия

$$\sum_{j=1}^{n} x_j = 1,$$

$$\sum_{j=1}^{n} \alpha_j x_j \ge b,$$

$$x_j \ge 0 \quad (j = \overline{1,n}),$$

где левая часть второго неравенства выражает ожидаемую прибыль в расчете на одну денежную единицу вложенных средств. Риск учитывается целевой функцией

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j,$$

которую следует минимизировать, где $\sigma_{ij} = M[(a_i - \alpha_i)\cdot(a_j - \alpha_j)]$ есть ковариация (или смешанный момент второго порядка) прибыли для ценных бумаг вида i и вида j, определяемая как математическое ожидание произведения отклонений этих случайных величин от их математических ожиданий.

Для дискретных случайных величин имеем

$$\sigma_{ij} = p_{ij}(a_i - \alpha_i)\cdot(a_j - \alpha_j),$$

где p_{ij} – вероятность приобретения одновременно ценных бумаг i-го и j-го вида. Величины σ_{ij} удобно задавать в виде квадратной симметричной матрицы n-го порядка:

$$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2n} \\ & & & \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_{nn} \end{pmatrix},$$

называемой корреляционной матрицей, где $\sigma_{ij} = \sigma_{ji}$. Отметим, что для расчета величин σ_{ij} можно использовать известную формулу $\sigma_{ij} = M(a_i, a_j) - \alpha_i \cdot \alpha_j \quad (i = \overline{1,m}; j = \overline{1,n})$.

Таким образом, модель задачи выбора портфеля ценных бумаг в рассматриваемом варианте сводится к следующей задаче нелинейного программирования:

$$\min z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_{ij}x_i x_j$$

при ограничениях

$$\sum_{j=1}^{n} x_j = 1,$$

$$\sum_{j=1}^{n} \alpha_j x_j \ge b,$$

$$x_j \ge 0 \quad (j = \overline{1,n}).$$

Приведенные примеры содержательных задач нелинейного программирования убеждают нас в том, что существуют множество разнообразных ситуаций, которые приводят к необходимости изучения математических моделей, являющихся задачами нелинейного программирования.

14.3 Методы одномерной оптимизации

Пусть задана одномерная функция $f(x)$, для которой требуется найти ее экстремальные значения. Такая задача называется задачей одномерной оптимизации. Как известно, необходимым условием экстремума функции одной переменной $f(x)$ является условие $f'(x) = 0$. Если решить это уравнение и найти множество его корней X^*, то любой $x \in X^*$ может быть точкой экстремума. Решение уравнения $f'(x) = 0$ в явном виде удается найти лишь в отдельных случаях. Задача отыскания корней уравнения $f'(x) = 0$, как правило, столь же сложна как и задача поиска экстремума функции $f(x)$. Этим диктуется необходимость рассмотрения эффективных методов, разработанных специально для задач оптимизации функции одной переменной.

Отдельное рассмотрение численных методов поиска экстремума функции одной переменной связано со следующими обстоятельствами:

1) эти методы являются основой многих приближенных алгоритмов решения задач нелинейного программирования;

2) в некоторых задачах, используя те или иные приемы, удается непосредственно с помощью методов одномерной оптимизации получать решение задачи многих переменных;

3) классы функций одной переменной являются удобной моделью для исследования эффективности методов оптимизации.

Следует отметить, что даже для произвольной функции одной переменной не существует универсальных методов нахождения ее экстремума. Поэтому приходится строить алгоритмы, ориентированные на те или иные классы функций, встречающиеся в реальных прикладных задачах.

Рассмотрим численные методы минимизации унимодальных функций.

Определение 14.1. Функция $y = f(x)$, где $x \in [a,b]$ называется унимодальной на $[a,b]$, если существует такая точка $x^* \in [a,b]$, что:

$$f(x_1) > f(x_2), \text{ если } x_1 < x_2 < x^* \text{ и } x_1, x_2 \in [a,b];$$

$$f(x_1) < f(x_2), \text{ если } x^* < x_1 < x_2 \text{ и } x_1, x_2 \in [a,b].$$

Очевидно, что строго выпуклые функции унимодальны (рис. 14.4).

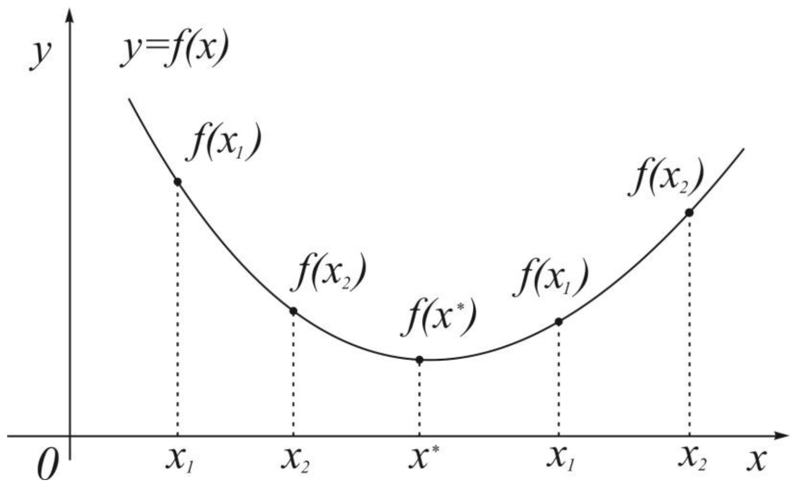

Рис. 14.4. Выпуклая унимодальная функция

Существуют невыпуклые и даже разрывные унимодальные функции (рис. 14.5).

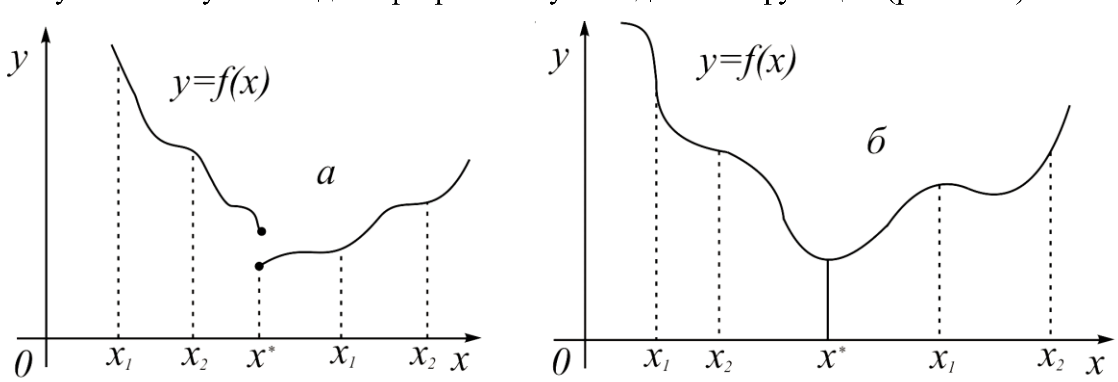

Рис. 14.5. Невыпуклая унимодальная функция: а – разрывная; б – непрерывная

Если функция $f(x)$ непрерывная, то свойство унимодальности функции означает наличие у нее единственного минимума. Будем в дальнейшем считать x^* точкой минимума унимодальной функции, полагая, что $\inf f(x)$ на $[a,b]$ достигается.

Очевидно, для унимодальной функции $y = f(x)$ на $[a,b]$, имеют место следующие утверждения:

– если $f(x_1) \le f(x_2)$, то $x^* < x_2$;

– если $f(x_1) \ge f(x_2)$, то $x^* > x_1$.

Предположение о том, что функция рассматривается на $[a,b]$, не является существенным. Действительно, если предположить, что она определена на всей прямой или полупрямой, то легко построить отрезок, в котором содержится точка x^*. Пусть, например, функция $y = f(x)$ определена на полупрямой $(-\infty, b]$. Вычислим значение функции в точках x_1 и $x_2 = x_1 + h$, где $h > 0$. Если $f(x_1) > f(x_2)$, то тогда $x^* > x_1$ и, следовательно, $x^* \in [x_1, b]$. Если же $f(x_1) < f(x_2)$, то следует вычислять последовательно $f(x_i)$, где $x_i = x_{i-1} + h$ $(i = 3, 4, \ldots)$ до тех пор, пока не будет выполнено условие $f(x_i) \ge f(x_{i-1})$, тогда $x^* \in [x_{i-2}, x_i]$.

Таким образом, всегда можно считать, что унимодальная функция рассматривается на некотором $[a,b]$.

Рассмотрим алгоритмы минимизации унимодальных функций, использующих в основном информацию о значениях минимизируемой функции $y = f(x)$, где число точек, в которых вычисляется значение функции, предлагается фиксированным и обозначается через n.

1. Равномерный (пассивный) поиск. Отрезок $[a,b]$ делится на подынтервалы точками $x_k = a + kh$ $(k = \overline{1,n})$, где $b = a + (n+1)h$. Шаг $h = \dfrac{b-a}{n+1}$ (рис. 14.6).

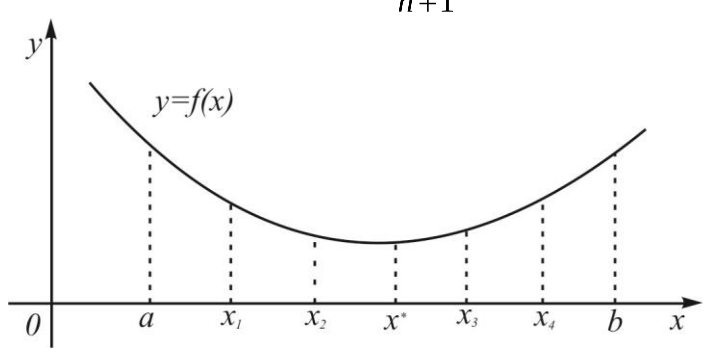

Рис. 14.6. Равномерный поиск минимума

Вычисляются значения функции в каждой из n точек сетки $y_k \equiv f(x_k)$ и выбирается точка x с наименьшим значением y. Если функция y унимодальна, то точка x^* ее минимума принадлежит $[x-h, x+h]$. Этот отрезок называют отрезком локализации минимума. Центр этого отрезка x и принимается за аппроксимацию x^*.

Для дальнейшего сокращения полученного интервала можно использовать этот же метод, разделив интервал $[x-h, x+h]$ на некоторое число интервалов с новым более мелким шагом h_1. На практике этот метод сначала используют с крупным шагом, а затем переходят к малому шагу.

Пример 14.4. Найти минимум функции $y = x^2 + 2x$, $x \in [-3;5]$.

Решение. Очевидно, что точный минимум этой функции достигается в точке $x = -1$ и $y_{\min} = -1$.

Применим метод равномерного поиска. Разделим $[-3;5]$ на $n = 9$ частей. Шаг $h = \dfrac{5+3}{9+1} = 0{,}8$. Точки деления $x_k = -3 + 0{,}8 \cdot k$ $(k = \overline{1,9})$. Вычислим значение функции в точках x_k. Результаты вычислений по методу пассивного поиска приведены в табл. 14.1.

Таблица 14.1		
k	x_k	y_k
1	–2,2	–10,44
2	–1,4	–10,84
3	–0,6	–10,84
4	–0,2	–10,44
5	–1,2	–13 ,44
6	–1,8	–16,54
7	–2,6	–11,96
8	–3,4	–18,36
9	–4,2	–26,04

Минимальное значение $y = -0{,}84$ достигается в двух точках $x_2 = -1{,}4$ и $x_3 = -0{,}6$. Любая их них может быть взята в качестве x^*. Так как точка x^* точного минимума принадлежит $[-2{,}2;-0{,}6]$ или $[-1{,}4;0{,}2]$, то она принадлежит их пересечению $[-1{,}4;-0{,}6]$. В качестве точки x^* можно взять середину последнего интервала $x^* = \dfrac{-1{,}4-0{,}6}{2} = -1$, что, в частности, точно совпало с точкой минимума рассматриваемой функции.

Очевидно, что можно предложить более эффективные процедуры, которые используют информацию, полученную на предыдущих шагах (итерациях) для определения положения точек последующих вычислений. Такие процедуры получили название методов последовательного поиска, к которым относятся метод дихотомии, метод золотого сечения, метод Фибоначчи, метод полиномиальной интерполяции.

2. Метод дихотомии. Рассмотрим унимодальную функцию, которую требуется минимизировать на $[a,b]$. Очевидно, что наименьшее число вычислений значений функции, которые необходимы для сокращения интервала поиска минимума (интервала локализации минимума) равно двум. На рис. 14.7 приведены две точки локализации x_1 и x_2.

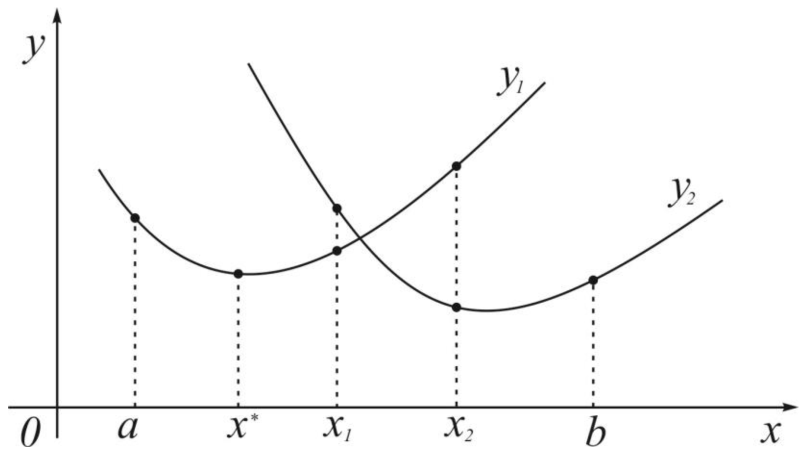

Рис. 14.7. Возможные интервалы локализации

При $y = y_1$ имеем $y(x_1) < y(x_2)$ и, следовательно, интервалом локализации является $[a, x_2]$. При $y = y_2$ имеем $y(x_1) > y(x_2)$ и интервалом локализации является $[x_1, b]$.

Таким образом, интервал локализации зависит от функции y. Заметим, что заранее неизвестно, будет ли $y(x_1) < y(x_2)$ или $y(x_1) > y(x_2)$ (если $y(x_1) = y(x_2)$, то интервалом локализации будет $[x_1, x_2]$). Поэтому оптимальная стратегия выбора точек x_1 и x_2 избежать наихудшей ситуации, что может быть достигнуто, если в качестве точек x_1 и x_2 взять две симметричные от середины $[a,b]$ точки на расстоянии $\delta > 0$, где δ – малая величина (рис. 14.8), т.е. полагаем $x_1 = \frac{a+b}{2} - \delta$ и $x_2 = \frac{a+b}{2} + \delta$.

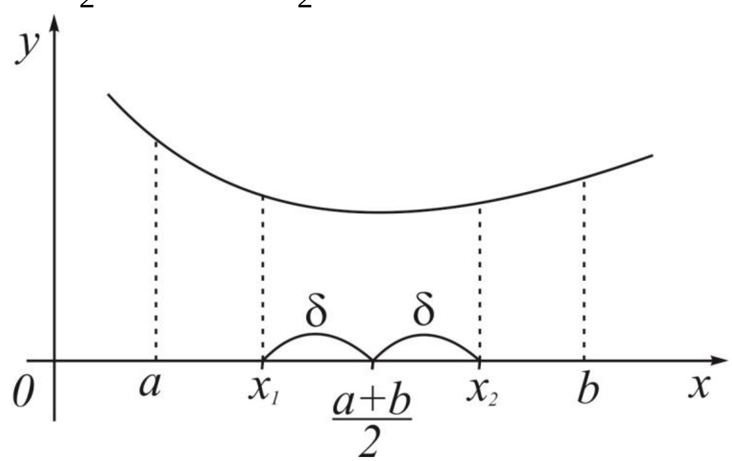

Рис. 14.8. Выбор точек в методе дихотомии

Таким образом, в дихотомическом поиске место каждой из двух первых точек x_1 и x_2 выбирается симметрично на расстоянии δ от середины $[a,b]$. В зависимости от значений функции $y = f(x)$ в точках x_1 и x_2 определяется новый отрезок локализации минимума. Затем процесс повторяется.

Алгоритм метода дихотомии для минимизации унимодальной функции на $[a,b]$ состоит из следующих этапов:

1. *Начальный этап* состоит в выборе константы $\delta > 0$ и допустимой конечной длины отрезка локализации $\varepsilon > 0$. Положим $k = 1$ и перейдем к основному этапу, считая $a_1 = a$ и $b_1 = b$.

2. *Основной этап. Шаг 1.* Если $|b_k - a_k| < \varepsilon$, то процесс следует остановить, точка минимума x^* принадлежит отрезку $[a_k, b_k]$ и в качестве ее приближенного значения следует взять $\frac{a_k + b_k}{2}$. Если же $|b_k - a_k| \geq \varepsilon$, то следует положить $x_1^k = \frac{a_k + b_k}{2} - \delta$, $x_2^k = \frac{a_k + b_k}{2} + \delta$ и вычислить $y(x_1^k)$ и $y(x_2^k)$ и перейти к шагу 2.

Шаг 2. Если $y(x_1^k) < y(x_2^k)$, то следует положить $a_{k+1} = a_k$ и $b_{k+1} = x_2^k$. В противном случае $a_{k+1} = x_1^k$ и $b_{k+1} = b_k$. Заменить k на $k+1$ и перейти к шагу 1.

Отметим, что длина отрезка локализации после первой пары вычислений равна $\frac{b-a}{2} + \delta$. Вторая пара вычислений проводится в точках, отстоящих на расстоянии δ по обе стороны от середины первого отрезка локализации. В результате получается второй отрезок локализации длиной $\left(\frac{b-a}{2} + \delta\right)/2 + \delta = \frac{b-a}{4} + \frac{3}{2}\delta$. После третьей пары вычислений длина отрезка локализации станет равной $\left(\frac{b-a}{4} + \frac{3}{2}\delta\right)/2 + \delta == \frac{b-a}{8} + \frac{7}{4}\delta$ и т.д. Длина l_k отрезка локализации в начале $(k+1)$-й итерации равна $l_k = \frac{b-a}{2^k} + \frac{2^k - 1}{2^{k-1}}\delta = \frac{b-a}{2^k} + 2\left(1 - \frac{1}{2^k}\right)\delta.$

Последняя формула может быть использована для определения числа k итераций, необходимых для достижения желаемой точности ε. Так как каждая итерация требует двух вычислений данной функции $y = f(x)$, то указанная формула может быть использована и для определения числа вычислений значений функции $y = f(x)$.

Пример 14.5. Найти минимум функции $y = x^2 + 2x$ на $[-3; 5]$ методом дихотомии.

Решение. Выберем согласно начальному этапу алгоритма величины $\delta = 0,5$ и $\varepsilon = 1$ и перейдем к основному этапу, состоящему из двух шагов.

Первая итерация. Шаг 1. Полагаем $a_1 = -3$ и $b_1 = 5$ и проверяем условие $|b_1 - a_1| < 1$, т.е. $|5 - (-3)| = |5 + 3| = 8 > 1$. Вычисляем

$$x_1^1 = \frac{a_1 + b_1}{2} - \delta = \frac{-3 + 5}{2} - 0,5 = 0,5 \text{ и } x_2^1 = \frac{-3 + 5}{2} + 0,5 = 1,5$$

и значения функции в этих точках

$$y(x_1^1) = (0,5)^2 + 2 \cdot 0,5 = 1,25 \text{ и } y(x_2^1) = (1,5)^2 + 2 \cdot 1,5 = 5,25.$$

Переходим к шагу 2.

Шаг 2. Так как $y(x_1^1) < y(x_2^1)$, (т.е. $1,25 < 5,25$), то полагаем $a_2 = a_1 = -3$ и $b_2 = x_2^1 = 1,5$. Получаем отрезок локализации минимума $[-3; 1,5]$. Переходим опять к шагу 1.

Вторая итерация. Шаг 1. Проверяем условие $|b_2 - a_2| < 1$. Так как $|1,5 - (-3)| = 4,5 > 1$, то

$$x_1^2 = \frac{a_2 + b_2}{2} - 0,5 = \frac{-3 + 1,5}{2} - 0,5 = -1,25, \; x_2^2 = \frac{-3 + 1,5}{2} + 0,5 = -0,25,$$

$$y(x_1^2) = (-1,25)^2 - 2,5 = -0,9375 \text{ и } y(x_2^2) = (-0,25)^2 - 0,5 = -0,4375.$$

Переходим к шагу 2.

Шаг 2. Так как $y(x_1^2) < y(x_2^2)$ (т.е. $-0,9375 < -0,4375$), то отрезком локализации минимума будет $[a_3, b_3] = [-3; -0,25]$. Переходят к шагу 1.

Третья итерация. Шаг 1. Проверяем условие $|b_3 - a_3| < 1$. Так как $|-0,25 + 3| > 1$, то полагаем

$$x_1^3 = \frac{-3 - 0,25}{2} - 0,5 = -2,125 \text{ и } x_2^3 = \frac{-3 - 0,25}{2} + 0,5 = -1,125$$

и вычисляем

$$y(x_1^3) = (-2,125)^2 + 2 \cdot (-2,125) = -0,266,$$
$$y(x_2^3) = (-1,125)^2 + 2 \cdot (-1,125) = -0,984.$$

Переходим к шагу 2.

Шаг 2. Так как $y(x_1^3) > y(x_2^3)$ (т.е. $-0,266 > -0,984$), то отрезком локализации минимума будет $[-2,125; -0,25]$. Переходим к следующей итерации и т.д.

Результаты вычислений по методу дихотомии приведены в табл. 14.2.

Таблица 14.2

Номер итерации	x_1^k	x_2^k	$y(x_1^k)$	$y(x_2^k)$	Отрезок локализации
1	–0,5000	–1,5000	–1,2500	–5,2500	[–3;1,5]
2	–1,2500	–0,2500	–0,9375	–0,4375	[–3;– 0,25]
3	–2,1250	–1,1250	–0,2660	–0,9840	[–2,125;– 0,25]
4	–1,6870	–0,6875	–0,5270	–0,9020	[–1,6875;– 0,25]
5	–1,4680	–0,4680	–0,7810	–0,7170	[–1,6875;– 0,468]
6	–1,5780	–0,5780	–0,6660	–0,8220	[–1,578;– 0,468]
7	–1,5230	–0,5230	–0,7260	–0,7220	[–1,523;– 0,468]
8	–1,4955	–0,4955	–0,7545	–0,7455	[–1,523;– 0,4955]
9	–1,5092	–0,5092	–0,7408	–0,7592	[–1,5092;– 0,4955]
10	–1,5023	–0,5023	–0,7481	–0,7523	[–1,5023;– 0,4955]
11	–1,4989	–0,4989	–0,7509	–0,7489	[–1,4989;– 0,4955]

Так как практически длина последнего отрезка локализации минимума равна заданному $\varepsilon = 1$, то процесс останавливаем и в качестве точки минимума берем середину последнего отрезка локализации минимума, т.е. полагаем $x^* = \dfrac{-1,51 - 0,49}{2} = -1$ и $y(x^*) = -1$. Отметим, что с шагом $\delta = 0,5$ дальше уменьшать этот отрезок локализации нельзя, и поэтому, чтобы получить большую точность, надо величину δ уменьшить и продолжить процесс, начиная уже с последнего отрезка локализации минимума $[-1,5; -0,5]$.

Замечание. Применяя формулу $l_k = \frac{b-a}{2^k} + 2\left(1 - \frac{1}{2^k}\right)\delta$ для нашего примера получаем, что при $\delta = 0,5 = \frac{1}{2}$

$$l_k = \frac{5+3}{2^k} + 2\left(1 - \frac{1}{2^k}\right)\cdot\frac{1}{2} = \frac{8}{2^k} + 1 - \frac{1}{2^k} = 1 + \frac{7}{2^k} > 1$$

при любом конечном k. При больших k значения $l_k \approx 1$.

Таким образом, величина ε в нашем примере при $\delta = 0{,}5$ не может быть взята меньшей 1. При $\delta = 0,25 = \frac{1}{4}$ имеем $l_k = \frac{15}{2^{k+1}} + \frac{1}{2} > \frac{1}{2}$. Следовательно, величина ε не может быть взято меньшей 0,5.

3. Метод золотого сечения. Опишем более эффективный метод минимизации унимодальных функций.

Пусть на k-й итерации этого метода отрезок локализации минимума равен $[a_k, b_k]$. Новый отрезок локализации минимума $[a_{k+1}, b_{k+1}]$ будет равен $[x_k, b_k]$, если $f(x_k) \geq f(x_{k+1})$ и $[a_k, x_{k+1}]$, если $f(x_k) \leq f(x_{k+1})$. Точки x_k и x_{k+1} выбираются исходя из следующих соображений:

1. Длина нового отрезка локализации минимума $b_{k+1} - a_{k+1}$ не должна зависеть от результата на k-й итерации, т.е. от того, выполняется ли неравенство $f(x_1^k) \geq f(x_2^k)$ или $f(x_2^k) \geq f(x_1^k)$. Кроме того, должно выполняться равенство $b_k - x_1^k = x_2^k - a_k$. Таким образом, если

$$x_1^k = a_k + (1-\alpha)(b_k - a_k), \text{где } \alpha \in (0;1)$$

то для x_2^k должно быть

$$x_2^k = a_k + \alpha(b_k - a_k).$$

С учетом равенства $b_k - x_1^k = x_2^k - a_k$ отсюда следует, что

$$b_{k+1} - a_{k+1} = \alpha(b_k - a_k).$$

2. Для новой $(k+1)$-й итерации x_1^{k+1} и x_2^{k+1} выбираются так, что либо x_1^{k+1} совпадает с x_2^k, либо x_2^{k+1} совпадает с x_1^k. Если этого достичь, то на $(k+1)$-й итерации потребуется только одно новое вычисление функции $f(x)$, что и является одним из важных преимуществ метода золотого сечения. Покажем что такой выбор можно осуществить. Рассмотрим рис. 14.9.

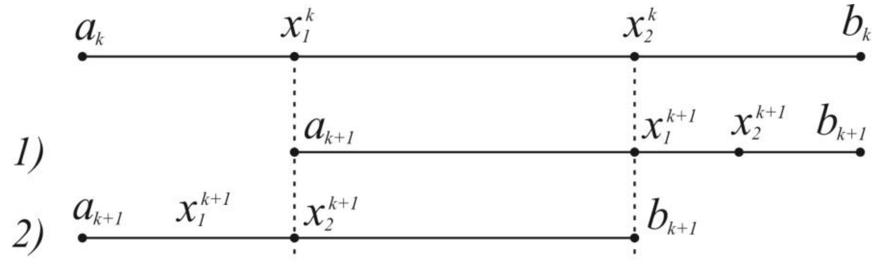

Рис. 14.9. Правило золотого сечения

Возможны следующие два случая.

Случай 1. Значение $f(x_1^k) > f(x_2^k)$. Тогда $a_{k+1} = x_1^k$ и $b_{k+1} = b_k$, и при $x_1^{k+1} = x_2^k$ (см. рис. 14.9 1) получим

$$x_2^k = x_1^{k+1} = a_{k+1} + (1-\alpha)(b_{k+1} - a_{k+1}) = x_1^k + (1-\alpha)(b_k - x_1^k).$$

Подставляя в это равенство выражения

$$x_1^k = a_k + (1-\alpha)(b_k - a_k) \quad \text{и} \quad x_2^k = a_k + \alpha(b_k - a_k),$$

получаем

$$a_k + \alpha(b_k - a_k) = a_k + (1-\alpha)(b_k - a_k) + (1-\alpha)(b_k - a_k - (1-\alpha)(b_k - a_k)),$$

откуда

$$[\alpha - (1-\alpha) - (1-\alpha) + (1-\alpha)^2](b_k - a_k) = 0.$$

Так как $b_k - a_k > 0$, то получим

$$-2 + 3\alpha + 1 - 2\alpha + \alpha^2 = 0,$$

или

$$\alpha^2 + \alpha - 1 = 0.$$

Случай 2. Значение $f(x_1^k) \le f(x_2^k)$. Тогда $a_{k+1} = a_k$ и $b_{k+1} = x_2^k$, и при $x_2^{k+1} = x_1^k$ получаем

$$a_k + (1-\alpha)(b_k - a_k) = a_{k+1} + \alpha(b_{k+1} - a_{k+1}).$$

Так как $a_k = a_{k+1}$ (см. рис. 14.9 2), то, учитывая ранее полученное равенство

$$b_{k+1} - a_{k+1} = \alpha(a_k - b_k),$$

имеем

$$(1-\alpha)(b_k - a_k) = \alpha^3(b_k - a_k),$$

откуда и получаем

$$\alpha^2 + \alpha - 1 = 0.$$

Корнями уравнения $\alpha^2 + \alpha - 1 = 0$ являются $\alpha_1 = \dfrac{-1+\sqrt{5}}{2} > 0$ и $\alpha_2 = \dfrac{-1-\sqrt{5}}{2} < 0$. Учитывая тот факт, что $\alpha \in (0;1)$, берем только первый корень $\alpha_1 = \dfrac{-1+\sqrt{5}}{2} \approx 0,618$.

Таким образом, если x_1^k и x_2^k на k-й итерации выбраны в соответствии с формулами

$$x_1^k = a_k + (1-\alpha)(b_k - a_k);$$
$$x_2^k = a_k + \alpha(b_k - a_k),$$

где $\alpha \approx 0,618$, то длина отрезка локализации минимума уменьшается с постоянным коэффициентом $\alpha \approx 0,618$. На первой итерации необходимы два вычисления значений данной функции $f(x)$ в точках x_1^1 и x_2^1, а на каждой последующей требуется только одно вычисление, так как либо $x_1^{k+1} = x_2^k$, либо $x_2^{k+1} = x_1^k$.

Отсюда алгоритм метода золотого сечения для минимизации унимодальной функции на $[a,b]$ состоит из двух этапов.

Начальный этап. Выбираем допустимую длину $\varepsilon > 0$ отрезка локализации минимума. Полагаем $a = a_1$ и $b = b_1$ и находим точки x_1^1 и x_2^1 по формулам

$$x_1^1 = a_1 + (1-\alpha)(b_1 - a_1),$$
$$x_2^1 = a_1 + \alpha(b_1 - a_1),$$

где $\alpha \approx 0{,}618$ $\left(\alpha = \dfrac{-1+\sqrt{5}}{2} \right)$.

Вычисляем значение данной функции в этих точках, т.е. $f(x_1^1)$ и $f(x_2^1)$.

Основной этап, состоящий из следующих четырех шагов.

Шаг 1. Если $b_k - a_k < \varepsilon$, то следует остановиться. Точка минимума функции принадлежит отрезку $[a_k, b_k]$. В противном случае нужно сравнить значения функции в точках x_1^k и x_2^k. И тогда, если $f(x_1^k) \geq f(x_2^k)$, то необходимо перейти к шагу 2, а если $f(x_1^k) \leq f(x_2^k)$, – то к шагу 3.

Шаг 2. Положить $a_{k+1} = x_1^k$ и $b_{k+1} = b_k$, $x_1^{k+1} = x_2^k$, $x_2^{k+1} = a_{k+1} + \alpha(b_{k+1} - a_{k+1})$. Вычислить значение функции $f(x_2^{k+1})$ и перейти к шагу 4.

Шаг 3. Положить $a_{k+1} = a_k$, $b_{k+1} = x_2^k$, $x_2^{k+1} = x_1^k$ и $x_1^{k+1} = a_{k+1} + (1-\alpha)(b_{k+1} - a_{k+1})$. Вычислить значение функции $f(x_1^{k+1})$ и перейти к шагу 4.

Шаг 4. Заменить k на $k+1$ и перейти к шагу 1.

Пример 14.6. Найти минимум $f(x) = x^2 + 2x$ на $[-3; 5]$ методом золотого сечения.

Решение. Согласно описанному алгоритму осуществляем начальный этап.

Начальный этап. Полагаем $a_1 = -3, b_1 = 5$ и выбираем $\varepsilon = 0{,}2$. Первые две точки определяем следующим образом: $x_1^1 = -3 + (1 - 0{,}618)\cdot 8 = 0{,}056$ и $x_2^1 = -3 + 0{,}618 \cdot 8 = 1{,}944$. Вычисляем значения данной функции в найденных точках:

$$f(x_1^1) = (0{,}056)^2 + 2 \cdot 0{,}056 = 0{,}115;$$

$$f(x_2^1) = (1{,}944)^2 + 2 \cdot 1{,}944 = 7{,}667$$

и переходим к основному этапу.

Основной этап. Так как $f(x_2^1) > f(x_1^1)$, то переходим к третьему шагу. Новый отрезок локализации минимума будет $[-3; 1{,}944]$, а так как его длина $4{,}944 > 0{,}2$, то процесс продолжается, т.е. полагаем $a_2 = a_1 = -3, b_2 = x_2^1 = 1{,}944, x_2^2 = x_1^1 = 0{,}056$, находим $x_1^2 = -3 + 0{,}382 \cdot 4{,}944 = -1{,}112$ и вычисляем $f(x_1^2) = -0{,}987$. Таким образом (согласно шагу 4) получаем $[a_2; b_2] = [-3; 1{,}944]$, точки $x_1^2 = -1{,}112; x_2^2 = 0{,}056$ и $f(x_1^2) = -0{,}387$, $f(x_2^2) = 0{,}115$, и опять переходим к шагу 1.

Результаты вычислений по методу золотого сечения приведены в табл. 14.3.

Таблица 14.3

Номер итерации	Отрезок локализации		Точки вычислений функции		Значения функции	
	a_k	b_k	x_1^k	x_2^k	$f(x_1^k)$	$f(x_2^k)$
1	–3	–5,944	0,056	1,944	0,115	7,667
2	–3	–1,944	–1,112	0,056	–0,987	0,115
3	–3	–0,056	–1,832	–1,112	–0,308	–0,987
4	–1,832	–0,056	–1,112	–0,664	–0,987	–0,887
5	–1,832	–0,664	–1,384	–1,112	–0,853	–0,987

6	–1,384	–0,664	–1,112	–0,936	–0,987	–0,996
7	–1,384	–0,936	–1,208	–1,112	–0,957	–0,987
8	–1,208	–0,936	–1,112	–1,032	–0,987	–0,999
9	–1,112	–0,936				

На 9-й итерации длина отрезка локализации функции $[a_9, b_9] = [-1,112; -0,936]$ т.е. длина $l = -0,936 - (-1,112) = 0,176 < 0,2 = \varepsilon$. В качестве точки минимума можно взять середину этого отрезка, т.е. $x^* = -1,024$.

Замечание. В методе золотого сечения длины отрезков связаны условием

$$l_{k-1} = l_k + l_{k+1},$$

и, кроме того, отношение длин последовательных интервалов постоянно, т.е. $\frac{l_{k+1}}{l_k} = \frac{l_k}{l_{k-1}} = \alpha$.

Разделив равенство $l_{k-1} = l_k + l_{k+1}$ на l_{k-1}, получим $\frac{l_{k+1}}{l_{k-1}} = 1 - \alpha$.

Из отношения длин последовательных интервалов получаем, что $l_{k+1} \cdot l_{k-1} = l_k^2$. Разделив последнее равенство на l_{k-1}^2, получаем $\frac{l_{k+1}}{l_{k-1}} = \frac{l_k^2}{l_{k-1}^2} = \alpha^2$. Следовательно, $\alpha^2 = 1 - \alpha$. Таким образом, получается уравнение $\alpha^2 + \alpha - 1 = 0$.

Деление отрезка в таком постоянном отношении α и носит название *золотого сечения*.

4. Метод Фибоначчи. Метод Фибоначчи подобно методу золотого сечения требует два вычисления значения функции на первой итерации и только по одному на каждой последующей итерации. Но этот метод отличается от метода золотого сечения тем, что величина отрезка локализации минимума меняется от итерации к итерации.

Этот метод основан на последовательности Фибоначчи, которая определяется следующим образом:

$$F_0 = F_1 = 1, \; F_{k+1} = F_k + F_{k-1}, \quad k = 1, 2, \ldots$$

Эта последовательность имеет вид

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, \ldots$$

Пусть на k-й итерации отрезок локализации минимума функции $f(x)$ будет $[a_k, b_k]$. Рассмотрим две точки x_1^k и x_2^k, определяемые по формулам

$$x_1^k = a_k + \frac{F_{n-k-1}}{F_{n-k+1}}(b_k - a_k), \; k = \overline{1, n-1};$$

$$x_2^k = a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k), \; k = \overline{1, n-1},$$

где n – заданное общее число вычислений значений функции. Новый отрезок локализации минимума $[a_{k+1}, b_{k+1}]$ определяется следующим образом:

- если $f(x_1^k) \geq f(x_2^k)$, то $a_{k+1} = x_1^k$ и $b_{k+1} = b_k$, т.е. $[a_{k+1}, b_{k+1}] = [x_1^k, b_k]$;
- если $f(x_1^k) \leq f(x_2^k)$, то $a_{k+1} = a_k$ и $b_{k+1} = x_2^k$, т.е. $[a_{k+1}, b_{k+1}] = [a_k, x_2^k]$.

Таким образом, в первом случае длина отрезка локализации равна

$$b_{k+1}-a_{k+1}=b_k-x_1^k=b_k-a_k-\frac{F_{n-k-1}}{F_{n-k+1}}\left(b_k-a_k\right)=$$

$$=\left(1-\frac{F_{n-k-1}}{F_{n-k+1}}\right)\left(b_k-a_k\right)=\frac{F_{n-k}}{F_{n-k+1}}\left(b_k-a_k\right),$$

а во втором

$$b_{k+1}-a_{k+1}=x_2^k-a_k=\frac{F_{n-k}}{F_{n-k+1}}(b_k-a_k).$$

Таким образом, в обоих случаях длина отрезка локализации сокращается с коэффициентом $\frac{F_{n-k}}{F_{n-k+1}}$.

Теперь покажем, что на $(k+1)$-й итерации либо $x_1^{k+1}=x_2^k$, либо $x_2^{k+1}=x_1^k$, так что потребуется лишь одно новое вычисление значения функции $f(x)$.

Предположим, что $f(x_1^k)>f(x_2^k)$. Тогда $a_{k+1}=x_1^k$ и $b_{k+1}=b_k$ и x_1^{k+1} вычисляется по следующей формуле (см. формулу для x_1^k с заменой k на $k+1$):

$$x_1^{k+1}=a_{k+1}+\frac{F_{n-k-2}}{F_{n-k}}(b_{k+1}-a_{k+1})=x_1^k+\frac{F_{n-k-2}}{F_{n-k}}(b_k-x_1^k).$$

Заменяя $x_1^k=a_k+\frac{F_{n-k-1}}{F_{n-k+1}}(b_k-a_k)$, получаем

$$x_1^{k+1}=a_k+\left(\frac{F_{n-k-1}-F_{n-k-2}}{F_{n-k}}\right)(b_k-a_k)=a_k+\frac{F_{n-k}}{F_{n-k+1}}(b_k-a_k)=x_2^k.$$

Аналогично, если $f(x_1^k)\le f(x_2^k)$, получаем $x_2^{k+1}=x_1^k$.

Таким образом, в обоих случаях на $(k+1)$-й итерации требуется только одно вычисление значения функции $f(x)$. Отметим, что в конце $(n-2)$-й итерации будет выполнено $(n-1)$ вычислений функции, и для $k=n-1$ получаем $x_1^{n-1}=x_2^{n-1}=\frac{1}{2}\left(a_{n-1}+b_{n-1}\right)$, как следует из формул для вычисления x_1^k и x_2^k. Следовательно, либо $x_1^{n-1}=x_2^{n-2}$, либо $x_2^{n-1}=x_1^{n-2}$, и тем самым не должно делаться новых вычислений функции. Практически точка последнего вычисления функции несколько перемещается влево или вправо от средней точки $x_1^{n-1}=x_2^{n-1}$.

Отметим, что в отличие от метода дихотомии и метода золотого сечения в методе Фибоначчи требуется, чтобы общее число n вычислений значения функции было выбрано заранее. Длина отрезка локализации на k-й итерации сокращается с коэффициентом $\frac{F_{n-k}}{F_{n-k+1}}$. Отсюда следует, что после $n-1$ итераций длина отрезка локализации будет равна $\frac{b-a}{F_n}$.

Таким образом, число n должно быть выбрано так, чтобы величина $\frac{b-a}{F_n}$ была меньше заданного $\varepsilon>0$.

Алгоритм метода Фибоначчи состоит из двух основных этапов – начального и основного.

Начальный этап. Выбрать допустимую конечную длину $\varepsilon > 0$ отрезка локализации минимума и константу $\delta > 0$ различимости.

Задать начальный $[a, b]$ и положить $a_1 = a$ и $b_1 = b$.

Выбрать число n вычислений функции так, чтобы $\dfrac{b_1 - a_1}{F_n} < \varepsilon$ или $F_n > \dfrac{b_1 - a_1}{\varepsilon}$.

Положить

$$x_1^1 = a_1 + \frac{F_{n-2}}{F_n}(b_1 - a_1) \text{ и } x_2^1 = a_1 + \frac{F_{n-1}}{F_n}(b_1 - a_1).$$

Вычислить $f(x_1^1)$ и $f(x_2^1)$ и перейти к основному этапу при $k = 1$.

Основной этап. Шаг 1. Если $f(x_1^k) > f(x_2^k)$, то перейти к шагу 2, а если $f(x_1^k) \le f(x_2^k)$, – то к шагу 3.

Шаг 2. Положить $a_{k+1} = x_1^k$, $b_{k+1} = b_k$ и положить $x_1^{k+1} = x_2^k$, а $x_2^{k+1} = a_{k+1} + \dfrac{F_{n-k-1}}{F_{n-k}}(b_{k+1} - a_{k+1})$. Если $k = n-2$, то перейти к шагу 4, в противном случае вычислить $f(x_2^{k+1})$, заменить k на $k+1$ и перейти к шагу 1.

Шаг 3. Положить $a_{k+1} = a_k, b_{k+1} = x_2^k, x_2^{k+1} = x_1^k$ и $x_1^{k+1} = a_{k+1} + \dfrac{F_{n-k-2}}{F_{n-k}}(b_{k+1} - a_{k+1})$. Если $k = n-2$, то перейти к шагу 4, в противном случае вычислить $f(x_1^{k+1})$ и перейти к шагу 1, заменив k на $k+1$

Шаг 4. Положить $x_1^n = x_1^{n-1}$ и $x_2^n = x_1^n + \delta$. Если $f(x_1^n) > f(x_2^n)$, то положить $a_n = x_1^n$ и $b_n = b_{n-1}$. Если же $f(x_1^n) \le f(x_2^n)$, то положить $a_n = a_{n-1}$ и $b_n = x_1^n$. Процесс вычислений остановить. Оптимальное решение x^* принадлежит отрезку локализации минимума $[a_n, b_n]$. Можно положить $x^* = \dfrac{a_n + b_n}{2}$.

Пример 14.7. Найти минимум функции $f(x) = x^2 + 2x$ на $[-3; 5]$, решенный ранее предыдущими методами.

Решение. На начальном этапе потребуем, чтобы длина конечного отрезка локализации минимума не превосходила числа $\varepsilon = 0{,}2$, и выберем константу $\delta = 0{,}01$. Так как $F_n > \dfrac{b-a}{\varepsilon}$, то $F_n > \dfrac{8}{0{,}2} = 40$. Откуда следует, что общее число n вычислений функции равно 9 (см. числа Фибоначчи $F_8 = 34, F_9 = 55$). Два первых вычисления значений функции осуществляются в точках

$$x_1^1 = -3 + \frac{F_7}{F_9} \cdot 8 = 0{,}054545, \quad x_2^1 = -3 + \frac{F_8}{F_9} \cdot 8 = 1{,}945454,$$

значения функции соответственно равны

$$f(x_1^1) = 0{,}112165 \quad \text{и} \quad f(x_2^1) = 7{,}6757,$$

и так как $f(x_1^1) < f(x_2^1)$, то новый отрезок локализации функции будет $[-1; 1{,}945]$. Далее процедура вычислений повторяется исходя из полученного отрезка по описанному алгоритму. Результаты вычислений по методу Фибоначчи приводятся в табл. 14.4.

Таблица 14.4

Номер итерации	Отрезок локализации		Точки вычислений функции		Значения функции	
	a_k	b_k	x_1^k	x_2^k	$f(x_1^k)$	$f(x_2^k)$
1	–3	$-5,944$	0,0545	1,9455	0,1121	7,6760
2	–3	$-1,9455$	$-1,1091$	0,0545	$-0,9881$	0,1121
3	–3	$-0,0545$	$-1,8364$	$-1,1091$	$-0,3005$	$-0,9881$
4	$-1,8364$	$-0,0545$	$-1,1091$	$-0,6727$	$-0,9881$	$-0,8929$
5	$-1,8364$	$-0,6727$	$-1,4000$	$-1,1091$	$-0,8400$	$-0,9881$
6	$-1,4000$	$-0,6727$	$-1,1091$	$-0,9636$	$-0,9881$	$-0,9987$
7	$-1,1091$	$-0,6727$	$-1,9636$	$-0,8182$	$-0,9987$	$-0,9669$
8	$-1,1091$	$-0,8182$	$-1,9636$	$-0,9636$	$-0,9987$	$-0,9987$
9	$-1,1091$	$-0,9636$	$-0,9636$	$-0,9536$	$-0,9987$	$-0,9979$

При $k=8$ имеем $x_1^8 = x_2^8 = -0{,}0636$, так что на этом нет необходимости вычислять функцию. Для $k=9$ положим $x_1^9 = x_1^8 = -0{,}9636$, а $x_2^9 = x_1^9 + \delta = -0{,}9636 + 0{,}01 = -0{,}9536$.

Поскольку $f(x_2^9) > f(x_1^9)$, то конечный отрезок локализации минимума $[a_9, b_9]$ равен $[-1{,}1091; -0{,}9636]$, длина которого равна $0{,}1455 < 0{,}2$. В качестве приближенного значения точки минимума x^* возьмем середину этого отрезка, т.е. $x^* = -1{,}03636$.

5. Методы полиномиальной интерполяции. Суть методов полиномиальной интерполяции заключается в том, что по полученным в ходе вычислений точкам строится интерполяционный многочлен (обычно второй или третьей степени) и его точка минимума принимается за точку очередного вычисления функции.

Рассмотрим только метод парабол, т.е. метод квадратичной интерполяции, когда в качестве полинома интерполяции берется квадратный трехчлен $y = ax^2 + bx + c$.

Пусть дан интервал $[a_1, b_1]$ локализации минимума функции $f(x)$ и пусть дана точка c_1 из этого интервала, т.е. $a_1 < c_1 < b_1$. Тройку чисел a_1, c_1, b_1 можно построить, сделав, например, несколько шагов по методу дихотомии или золотого сечения. Построим квадратный трехчлен y_1 график которого проходит через три точки $(a_1, f(a_1)), (c_1, f(c_1))$ и $(b_1, f(b_1))$. Этот трехчлен будет иметь вид

$$y_1 = \left(\frac{f(b_1) - f(c_1)}{b_1 - c_1} + \frac{f(a_1) - f(c_1)}{c_1 - a_1} \right) \cdot \frac{(x - c_1)(x - b_1)}{b_1 - a_1} + \frac{f(b_1) - f(c_1)}{b_1 - c_1} \cdot (x - c_1) + f(c_1).$$

Так как $f(x)$ предполагается унимодальной, то хотя бы одно из неравенств $f(c_1) \le f(a_1)$, $f(c_1) \le f(b_1)$ строгое, и тогда коэффициент при x^2 в квадратном трехчлене y_1 положителен. Легко убедиться, что y_1 достигает минимума в точке

$$s_1 = c_1 + \frac{1}{2} \frac{(b_1 - c_1)^2 (f(a_1) - f(c_1)) - (c_1 - b_1)^2 (f(b_1) - f(c_1))}{(b_1 - c_1)(f(a_1) - f(c_1)) + (c_1 - a_1)(f(b_1) - f(c_1))},$$

которая находится из условия $\frac{dy_1}{dx}=0$, причем $\frac{a_1+c_1}{2}\le s_1\le\frac{c_1+b_1}{2}$. Эта точка и выбирается в качестве новой очередной точки вычисления функции.

Заметим, что если $s_1=c_1$, то в качестве точки очередного вычисления значения функции можно выбрать точку $\frac{a_1+c_1}{2}$.

Как и в методе золотого сечения или Фибоначчи, определим новый отрезок локализации $[a_2,b_2]$ с лежащей внутри него точкой c_2, в которой значение функции $f(x)$ уже вычислено. Очевидно, что $c_2=c_1$ или $c_2=s_1$. Далее опять строится квадратный трехчлен y_2, график которого проходит через точки $(a_2, f(a_2)),(c_2, f(c_2))$ и $(b_2, f(b_2))$ и определяется его точка минимума s_2 и т.д.

В рассмотренных нами методах при заданной длине ε конечного отрезка локализации минимума функции необходимое число n вычислений функции может быть определено как наименьшее положительное целое, удовлетворяющее условию: длина отрезка локализации $l_n=b_n-a_n<\varepsilon$. Имеем следующие оценки:

– для метода равномерного поиска $n\ge\frac{2(b-a)}{\varepsilon}-1$;

– для метода дихотомии $(\frac{1}{2})^{\frac{n}{2}}\ge\frac{\varepsilon}{b-a}$;

– для метода золотого сечения $(0,618)^{n-1}\ge\frac{\varepsilon}{b-a}$;

– для метода Фибоначчи $F_n\ge\frac{b-a}{\varepsilon}$.

Из этих выражений видно, что число n в указанных методах зависит от величины $\frac{b-a}{\varepsilon}$. Для фиксированного значения $\frac{b-a}{\varepsilon}$ наименьшее число n вычисления значения функции $f(x)$ отвечает более эффективному алгоритму. С этой точки зрения наиболее эффективным алгоритмом является метод Фибоначчи, затем метод золотого сечения, метод дихотомии и, наконец, метод равномерного поиска.

6. Минимизация дифференцируемых функций. Подчеркнем, что рассмотренные выше методы не используют производных минимизируемой функции $f(x)$. Если заданная функция $f(x)$ дифференцируема, то информацию о производных следует использовать при построении процедуры нахождения приближенного значения минимума данной функции. Имеется ряд эффективных методов, использующих информацию о производных в выбираемых точках. К таким методам, в частности, относятся метод деления отрезка пополам и метод Ньютона.

Рассмотрим для начала метод деления отрезка пополам.

Предположим, что требуется минимизировать унимодальную дифференцируемую функцию $f(x)$ на $[a,b]$. Пусть на k-й итерации отрезок локализации минимума равен $[a_k,b_k]$. Предположим, что имеется точка $x_k\in[a_k,b_k]$ и известна $f'(x_k)$. Возможны следующие три случая:

Случай 1. $f'(x_k)=0$, тогда x_k – точка минимума функции (см. рис. 14.10 а).

Случай 2. $f'(x_k) > 0$, тогда для $x > x_k$ имеем $f'(x_k)(x - x_k) > 0$, а это означает, что $f(x) > f(x_k)$, и тогда точка минимума лежит левее точки x_k, так что новым отрезкам локализации минимума $[a_{k+1}, b_{k+1}]$ будет отрезок $[a_k, x_k]$ (см. рис. 14.10 б).

Случай 3. $f'(x_k) < 0$, тогда для $x < x_k$ получаем что $f'(x_k)(x - x_k) > 0$. В этом случае минимум лежит справа от x_k и новым отрезком локализации минимума $[a_{k+1}, b_{k+1}]$ будет отрезок $[x_k, b_k]$ (см. рис. 14.10 в).

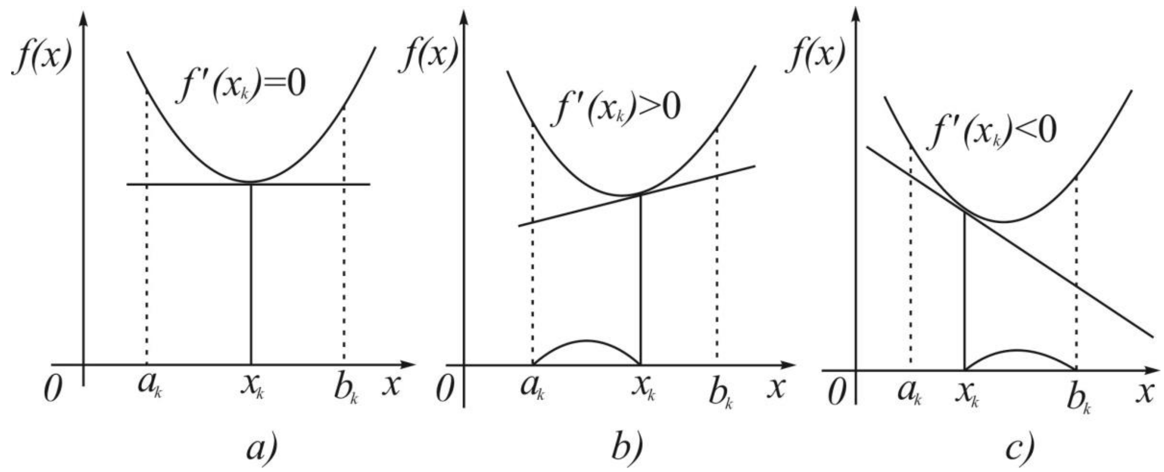

Рис. 14.10. Расположение точек в методе деления отрезка пополам

В рассматриваемом методе полагаем $x_k = \dfrac{a_k + b_k}{2}$, т.е. x_k есть середина рассматриваемого отрезка локализации минимума. Таким образом, на любой итерации k производная $f'(x)$ вычисляется в средней точке отрезка локализации минимума. В зависимости от заданной точности процесс либо останавливается, либо строится новый отрезок локализации минимума. Процедура деления отрезка пополам очень похожа на метод дихотомии, только в методе дихотомии необходимы на каждой итерации два вычисления значения функции, а в методе деления отрезка пополам требуется только одно вычисление производной $f'(x_k)$. Заметим, что длина отрезка локализации минимума после n итераций равна $\dfrac{b-a}{2^n}$ так, что метод сходится к точке минимума x^* с любой заданной точностью. В частности, если задана длина $\varepsilon > 0$ конечного отрезка локализации минимума, то n должно быть выбрано наименьшим целым, удовлетворяющим условию

$$\frac{\varepsilon}{b-a} \geq (\frac{1}{2})^n.$$

Алгоритм метода деления отрезка пополам можно представить в виде осуществления двух этапов.

Начальный этап. Пусть $[a, b]$ – заданный интервал, а ε – требуемая длина конечного отрезка локализации минимума. Положим n равным наименьшему положительному целому числу, удовлетворяющему условию $\left(\dfrac{1}{2}\right)^n \leq \dfrac{\varepsilon}{b-a}$. Положим $a_1 = a, b_1 = b$ и $k = 1$ и перейдем к основному этапу.

Основной этап. Шаг 1. Положим $x_k = \dfrac{a_k + b_k}{2}$ и вычислим $f'(x_k)$. Если $f'(x_k) = 0$, то остановимся; x_k – оптимальное решение. В противном случае перейдем к шагу 2, если $f'(x_k) > 0$ и к шагу 3, если $f'(x_k) < 0$.

Шаг 2. Положим $a_{k+1} = a_k, b_{k+1} = x_k$ и перейдем к шагу 4.

Шаг 3. Положим $a_{k+1} = x_k, b_{k+1} = b_k$ и перейдем к шагу 4.

Шаг 4. Если $k = n$, то процесс закончен, точка минимума принадлежит $[a_{k+1}, b_{k+1}]$. В противном случае заменим k на $k+1$ и перейдем к шагу 1.

Замечание. В данном алгоритме можно не находить число n на начальном этапе, а каждый раз на шаге 4 вместо условия $k = n$ проверять выполнение неравенства $b_{k+1} - a_{k+1} \le \varepsilon$. Если это условие выполняется, то процесс прекратить.

Пример 14.8. Найти минимум $f(x) = x^2 + 2x$ при $x \in [-3; 6]$.

Решение. Пусть длина отрезка локализации минимума не должна превышать $\varepsilon = 0{,}2$. Из условия $\left(\dfrac{1}{2}\right)^n < \dfrac{0{,}2}{6+3} = 0{,}022$ находим, что $n = 6$. Результаты вычислений методом деления отрезка пополам приведены в табл. 14.5.

Таблица 14.5

Номер итерации	Отрезок локализации		Середина интервала	Значение производной
k	a_k	b_k	x_k	$f'(x_k)$
1	–3	–6,8907	–1,5000	–5,0000
2	–3	–1,5000	–0,7500	–0,5000
3	–3	–0,7500	–1,8750	–1,7500
4	–1,8750	–0,7500	–1,3125	–0,6250
5	–1,3125	–0,7500	–1,0313	–0,0625
6	–1,0313	–0,7500	–0,8907	–0,2186
7	–1,0313	–0,8907	–0,9610	

Конечный отрезок локализации минимума $[a_7, b_7]$ равен $[-1{,}0313; -0{,}8907]$ и его длина равна $0{,}1406 < 0{,}2$. В качестве точки минимума берем середину этого отрезка $x^* = x_7 = -0{,}961$.

Рассмотрим теперь метод Ньютона. Он основан на использовании квадратичной аппроксимации функции $f(x)$ в заданной точке x_k. Используя разложение функции $f(x)$ в окрестности точки x_k в ряд Тейлора, получаем, что квадратичная аппроксимация $Q(x)$ функции $f(x)$ имеет вид

$$Q(x) = f(x_k) + f'(x_k)\cdot(x - x_k) + \frac{1}{2} f''(x_k)\cdot(x - x_k)^2.$$

В качестве очередной точки x_{k+1} берется точка, в которой производная функции $Q(x)$ обращается в ноль, т.е. из уравнения

$$f'(x_k) + f''(x_k)\cdot(x - x_k) = 0$$

получаем

$$x = x_k - \frac{f'(x_k)}{f''(x_k)} = x_{k+1}.$$

Таким образом получаем основную формулу вычислений по методу Ньютона:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}, \quad k = 0,1,2,\ldots$$

Процесс останавливаем, когда $|x_{k+1} - x_k| < \varepsilon$ или когда $|f'(x_k)| < \varepsilon$, где $\varepsilon > 0$ – заранее заданное малое число.

Отметим, что в отличии от метода деления пополам, использующий только первую производную минимизируемой функции $f(x)$ в методе Ньютона используются и вторые производные функции $f(x)$, т.е. метод Ньютона может быть применен только для минимизации дважды дифференцируемых функций, причем таких, для которых $f''(x_k) \neq 0$ на каждом шаге k.

Известно, что метод Ньютона вообще говоря не сходится к стационарной точке x^* при произвольной выбранной начальной точке x_1. (Если же начальная точка x_1 достаточно близка к стационарной точке, то сходимость будет обеспечена.)

Следующий пример, ярко иллюстрирует этот недостаток метода Ньютона.

Пример 14.9. Рассмотрим функцию $f(x)$, определенную следующим образом

$$f(x) = \begin{cases} 4x^3 - 3x^4, \text{ если } x \geq 0, \\ 4x^3 + 3x^4, \text{ если } x < 0. \end{cases}$$

Эта функция всюду дважды дифференцируема, Стационарными точками этой функции являются $x = 0, x = 1, x = -1$, в чем нетрудно убедиться, приравняв первую производную этой функции к нулю. Первая производная имеет вид

$$f'(x) = \begin{cases} 12x^2 - 12x^3, \text{ если } x \geq 0, \\ 12x^2 + 12x^3, \text{ если } x < 0, \end{cases}$$

а вторая производная

$$f''(x) = \begin{cases} 24x - 36x^2, \text{ если } x \geq 0, \\ 24x + 36x^2, \text{ если } x < 0. \end{cases}$$

В точке $x = 1$ эта функция имеет максимум, равный $+1$, в точке $x = -1$ имеет минимум, равный -1, а стационарная точка $x = 0$ является точкой перегиба (рис. 14.11)

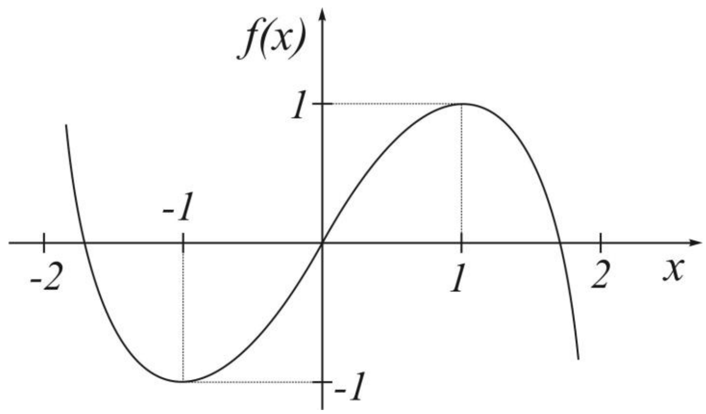

Рис. 14.11. График функции

Применим метод Ньютона с точки $x_1 = 0{,}4$. Результаты вычислений с использованием формулы $x_{k+1} = x_k - \dfrac{f'(x_k)}{f''(x_k)}$ приводятся в табл. 14.6.

Таблица 14.6

k	x_k	$f'(x_k)$	$f''(x_k)$	x_{k+1}
1	0,4	1,1520	3,8400	0,1000
2	0,1000	0,1080	2,0400	0,0471
3	0,0471	0,0253	1,0497	0,0229
4	0,0229	0,0062	0,5315	0,0113
5	0,0113	0,0015	0,2673	0,0056
6	0,0056	0,0004	0,1341	0,0028

Из табл. 14.6 видно, что процесс сходится к стационарной точке $x = 0$.

Если же положить $x_1 = 0{,}6$, то, как видно из табл. 14.7, получаемые точки x_{k+1} по указанной выше формуле попеременно принимают значения 0,6 и $-0{,}6$, и процесс зацикливается.

Таблица 14.7

k	x_k	$f'(x_k)$	$f''(x_k)$	x_{k+1}
1	$-0{,}6$	1,728	$-1{,}44$	$-0{,}6$
2	$-0{,}6$	1,728	$-1{,}44$	$-0{,}6$
3	$-0{,}6$	1,728	$-1{,}44$	$-0{,}6$
4	$-0{,}6$	1,728	$-1{,}44$	$-0{,}6$

Таким образом, выбор начальной точки в методе Ньютона играет важную роль – она должна быть достаточно близка к искомой стационарной точке, что заранее не всегда можно установить.

14.4 Задачи безусловной многомерной оптимизации

В этом параграфе обсуждается общая постановка задачи безусловной многомерной оптимизации, повторяются некоторые сведения из общего курса высшей математики, относящиеся к данной теме.

1. Условия оптимальности задач. Задача безусловной многомерной оптимизации проста и естественна: задана функция n переменных $x = (x_1, x_2, \ldots, x_n)$ $z = f(x) = f(x_1, x_2, \ldots, x_n)$ $(n \geq 2)$, определенная на всем n-мерном пространстве R^n. Требуется найти точки минимума или максимума функции $f(x)$ и ее значения в найденных точках.

В дальнейшем будем рассматривать только задачу на минимум, так как задача на максимум сводится к задаче на минимум заменой функции z функцией $-z$.

Задачу будем записывать в виде

$$z = f(x) \to \min, \quad x = (x_1, x_2, \ldots, x_n) \in R^n.$$

Напомним, что точка $x^* \in R^n$ называется:

- точкой глобального минимума функции $z = f(x)$, если $f(x^*) \leq f(x)$ для всех $x \in R^n$;
- точкой локального минимума функции $z = f(x)$, если $f(x^*) \leq f(x)$ для всех $x \in U_\varepsilon(x^*)$, где $U_\varepsilon(x^*)$ – ε-окружность точки x^*, которую можно взять в виде шара радиусом $\varepsilon > 0$ с центром в точке x^*.

Если условия $f(x^*) \leq f(x)$ выполняются как строгие неравенства, то говорят, что x^* – точка строгого глобального и локального минимума функции $z = f(x)$ соответственно. Говорят также, что точка x^* реализует величину $z^* = f(x^*) = \min\limits_{x \in R^n} f(x)$.

Для задачи максимизации функции $f(x)$ на R^n заменим в данных выше определениях слово "минимум" на "максимум" и знак $\leq$ в неравенстве $f(x^*) \leq f(x)$ на противоположный $\geq$, получим соответствующие понятия точек глобального и локального максимума. Точки минимума и максимума функции $z = f(x)$ называют точками экстремума, а сами задачи нахождения минимума и максимума функции $z = f(x)$ экстремальными задачами, или задачами оптимизации.

При изучении любого типа задач оптимизации важное место занимает вопрос об условиях оптимальности или условиях экстремума. Различают необходимые и достаточные условия оптимальности.

Интерес к условиям оптимальности объясняется тем, что они:

- составляют теоретическую основу методов изучения свойств задач оптимизации;
- используются при построении и обосновании численных методов решении задач оптимизации;
- позволяют в некоторых случаях явно найти решение рассматриваемой задачи оптимизации.

Теория необходимых и достаточных условий оптимальности для задач безусловной оптимизации, как правило, излагается в курсе высшей математики обычно для функции двух переменных. Напомним соответствующие результаты.

Пусть $\operatorname{grad} f(x^*) = \left(\dfrac{\partial f}{\partial x_1}\left(x^*\right), \ldots, \dfrac{\partial f}{\partial x_n}\left(x^*\right) \right)$ – градиент функции $z = f(x)$ в точке $x^* \in R^n$, представляющий собой вектор, координатами которого являются частные производные функции $z = f(x)$, вычисленные в точке x^*.

Следующая теорема указывает необходимое условие локальной оптимальности – условие оптимальности первого порядка.

Теорема 14.1. *Пусть функция $z = f(x)$ дифференцируема в точке $x^* \in R^n$. Если x^* – точка локального минимума функции $z = f(x)$, то grad $f(x^*) = 0$, т.е.*

$$\frac{\partial f}{\partial x_j} = 0 \quad \text{при} \quad x_j = x_j^* \quad (j = \overline{1,n}).$$

Точка x^*, удовлетворяющая этим условиям оптимальности первого порядка, называется стационарной точкой функции $z = f(x)$. Но это условие не является достаточным, т.е. не всякая стационарная точка является точкой локального минимума функции $z = f(x)$.

Достаточное условие оптимальности использует информацию о вторых частных производных функции $z = f(x)$. Рассмотрим так называемую матрицу Гессе H (гессиан) из вторых частных производных функции $f(x)$, вычисленные в точке x^*, т.е. матрицу вида

$$H(x^*) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

Тогда достаточное условие локальной оптимальности задается следующей теоремой.

Теорема 14.2. *Пусть $z = f(x)$ дважды дифференцируема в точке x^*. Пусть grad $f(x^*) = 0$, т.е. точка x^* является стационарной, а матрица Гессе $H(x^*)$ положительно определена. Тогда x^* – точка строгого локального минимума функции $f(x)$.*

Отметим, что для проверки матрицы на положительную определенность, как правило, используется критерий Сильвестра.

Замечание. *Если рассматривается задача на максимум, то необходимые условия оптимальности те же, а достаточные условия отличаются лишь тем, что матрица Гессе $H(x^*)$ должна быть отрицательно определенной.*

Доказательства теорем 14.1 и 14.2 основаны на теореме Тейлора о разложении функции $f(x)$ как функции многих переменных в окрестности точки x^*.

Приведенные теоремы 14.1 и 14.2 позволяют в ряде случаев явным образом решить задачу безусловной оптимизации. Проиллюстрируем это на следующем примере.

Пример 14.10. Дана функция

$$z = x_1^2 + x_2^2 + x_3^2 - x_1 - 2x_3 - x_2 x_3.$$

Найти экстремумы этой функции.

Решение. Найдем стационарные точки. Из необходимого условия $\operatorname{grad} z(x^*) = 0$ следует что

$$\frac{\partial z}{\partial x_1} = 2x_1 = 1,$$

$$\frac{\partial z}{\partial x_2} = 2x_2 - x_3 = 0,$$

$$\frac{\partial z}{\partial x_3} = 2x_3 - x_2 = 2.$$

Решением этой системы является $x_1 = \frac{1}{2},\ x_2 = \frac{2}{3},\ x_3 = \frac{4}{3}.$ Следовательно, единственной стационарной точкой будет точка $x^* = \left(\frac{1}{2}; \frac{2}{3}; \frac{4}{3}\right)$. Чтобы проверить выполнение достаточного условия, вычислим вторые частные производные

$$\frac{\partial^2 z}{\partial x_1^2} = 2,\ \frac{\partial^2 z}{\partial x_1 \partial x_2} = 0,\ \frac{\partial^2 z}{\partial x_1 \partial x_3} = 0,\ \frac{\partial^2 z}{\partial x_2^2} = 2,\ \frac{\partial^2 z}{\partial x_2 \partial x_3} = -1,\ \frac{\partial^2 z}{\partial x_3^2} = 2$$

и составим матрицу Гессе $H(x^*)$:

$$H(x^*) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}.$$

Главные миноры этой матрицы $\Delta_1 = 2 > 0, \Delta_2 = 4 > 0$ и $\Delta_3 = 6 > 0$. Следовательно, матрица Гессе является положительно определенной согласно критерию Сильвестра и точка $x^* = \left(\frac{1}{2}; \frac{2}{3}; \frac{4}{3}\right)$ является точкой минимума. Значение минимума

$$z^* = \frac{1}{4} + \frac{4}{9} + \frac{16}{9} - \frac{1}{2} - \frac{8}{3} - \frac{8}{9} = -4\frac{1}{4}.$$

Отметим, если $H(x^*)$ – неопределенная матрица, то x^* не является точкой экстремума (в таком случае x^* – так называемая седловая точка). Если $H(x^*)$ – полуопределенная матрица, то вопрос о том, является ли x^* точкой экстремума остается открытым и требуются дополнительные исследования для окончательного решения вопроса.

Классическая теория безусловной оптимизации функции многих переменных требует решения в общем случае нелинейной системы уравнений

$$\frac{\partial f}{\partial x_j} = 0 \ \ (j = \overline{1, n}),$$

что является сложной, а в некоторых случаях неразрешимой задачей. Поэтому возникает необходимость использования численных приближенных методов для решения задачи безусловной многомерной оптимизации. Имеется много численных методов и их модификаций для нахождения экстремума функции $f(x)$, среди которых выделим лишь следующие: градиентный метод, метод Ньютона и метод сопряженных градиентов.

2. Градиентный метод. Любой приближенный численный метод решения задачи безусловной оптимизации основан на вычислении характеристик данной функции (значений данной функции и ее производных). На основе полученной информации строится приближение к решению задачи – искомой точке минимума x^*.

Для каждой конкретной задачи вопрос о том, какие характеристики следует использовать, зависит от свойств минимизируемой функции и имеющихся возможностей по хранению и обработке информации.

Алгоритмы, использующие лишь информацию о значениях минимизируемой функции (см., например, методы золотого сечения, Фибоначчи, дихотомии для одномерной оптимизации), называются алгоритмами нулевого порядка; алгоритмы, использующие также информацию о значениях первых производных, – алгоритмами первого порядка; алгоритмы, использующие, кроме того, информацию о вторых производных, – алгоритмами второго порядка.

Многие методы минимизации функций нескольких переменных относятся к так называемым методам спуска. В методах спуска направление движения к минимуму на каждом шаге выбирается так, чтобы значение минимизируемой функции уменьшалось.

Говорят, что вектор $g = (g_1, g_2, \ldots, g_n)$ задает направление убывания (спуска) функции $z = f(x)$ в точке x, если $f(x+hg) < f(x)$ при малых $h > 0$. Сам вектор g называют направлением убывания или спуска функции $f(x)$. Таким, образом, любой достаточно малый сдвиг из x в направлении вектора g приводит к уменьшению функции $f(x)$.

Метод минимизации функции $z = f(x)$ называют методом спуска, если на каждой k-й итерации вектор g_k задает направление убывания функции $f(x)$ в точке x_k т.е.

$$x_{k+1} = x_k + h_k \cdot g_k$$

и число $h_k > 0$ и таково, что

$$f(x_{k+1}) < f(x_k).$$

Простейшим примером метода спуска является градиентный метод, в котором $g_k = -grad\, f(x_k)$.

Обычно коэффициенты h_k в методе спуска определяются из условия

$$f(x_k + h_k g_k) = \min_h f(x_k + hg_k).$$

Такой способ выбора числа h_k является наилучшим, в том смысле, что он обеспечивает достижение наименьшего значения функции $f(x)$ вдоль данного направления. Однако этот способ выбора h_k требует решения на каждом шаге одномерной задачи минимизации.

Рассмотрим градиентный метод, в котором вектор g_k направления спуска берется равным антиградиенту, т.е. $g_k = -grad\, f(x_k)$. В градиентном методе последовательность точек приближения $x_1, x_2, \ldots$ строится по правилу

$$x_{k+1} = x_k - h_k \cdot grad\, f(x_k),$$

где $k = 0, 1, 2, \ldots$, а x_0 – начальная заданная точка. Такой выбор вектора g_k определяется тем, что градиент функции в данной точке x_k указывает направление наибольшего возрастания функции, а, следовательно, антиградиент указывает направление наибольшего убывания функции из данной точки x_k. Если длина шага h_k выбирается из условия минимизации функции вдоль направления антиградиента, то получается вариант градиентного метода, носящий название метода наискорейшего спуска. Этот метод является одной из наиболее фундаментальных процедур минимизации дифференцируемой функции нескольких переменных и был предложен Коши в середине XIX века.

Алгоритм наискорейшего спуска состоит из двух этапов.

Начальный этап. Задать константу $\varepsilon > 0$, по которой процесс вычислений останавливается. Выбрать начальную точку x_0, положить $k = 0$, вычислить $grad\, f(x)$ в точке x_0 и перейти к основному этапу.

Основной этап. Если $|grad\, f(x_k)| = \sqrt{\sum_{j=1}^{n}(\frac{\partial f}{\partial x_j})^2} < \varepsilon$, то процесс остановить, в противном случае положить $g_k = -grad\, f(x_k)$, найти h_k, решив задачу минимизации $f(x_k + hg_k)$ при $h > 0$. Положить $x_{k+1} = x_k + h_k \cdot g_k$, заменить k на $k+1$ и повторить основной этап.

Пример 14.11. Найти минимум функции двух переменных

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

Решение. Решим полученную задачу по методу наискорейшего спуска при начальной точке $x_0 = (0;3)$. Градиент этой функции в произвольной точке $x = (x_1, x_2)$ имеет вид

$$grad\, z = (4(x_1 - 2)^3 + 2(x_1 - 2x_2); -4(x_1 - 2x_2)).$$

В начальной точке $x_0 = (0;3)$ $grad\, z(x_0) = (44;24)$. Значение функции в начальной точке $z(x_0) = 52$. Выберем константу останова ε равной 0,1. На этом предварительный этап заканчивается. Переходим к основному этапу.

Так как $|grad\, z(x_0)| = \sqrt{(-44)^2 + (24)^2} \approx 50{,}12 > \varepsilon$, то процесс вычисления продолжается. Находим точку $x_1 = x_0 - h_0 grad\, z(x_0) = (0 + 44h_0; 3 - 24h_0)$, где h_0 есть решение на минимум задачи

$$\min_h f(x_1 + hg_1) = \min_h f(44h; 3 - 24h) =$$
$$= \min_h [(44h - 2)^4 + (44h - 6 + 48h)^2] = \min_h [(44h - 2)^4 + (92h - 6)^2].$$

Минимум этой функции одной переменной h достигается приближенно при $h = 0{,}062$ и, следовательно, новая точка $x_1 = (2{,}73;1{,}51)$. В этой новой точке градиент x_1 есть вектор (0,73; 1,28) и его модуль, равный $\sqrt{(0{,}73)^2 + (1{,}28)^2} \approx 1{,}47$, больше $\varepsilon = 0{,}1$. Следовательно, второй этап надо повторить, построив точку x_2, взяв в качестве исходной найденную точку x_1. Результаты вычислений по методу наискорейшего спуска представлены в табл. 14.8.

Таблица 14.8

Номер итерации	Точки x_k	Значения функции	Градиент функции	Модуль градиента	Значение коэффициента
		$f(x_k)$	в точке x_k		h_k
0	(0,00; 3,00)	52,000	$(-44{,}00;24{,}00)$	50,12	0,062
1	(2,73; 1,51)	50,340	(0,73;1,28)	51,47	0,240
2	(2,52; 1,20)	50,090	$(0{,}80;-0{,}48)$	50,93	0,110

3	(2,43; 1,28)	5 0,040	(0,18;0,28)	5 0,33	0,310
4	(2,37; 1,16)	5 0,020	(0,30;– 0,20)	5 0,36	0,120
5	(2,33; 1,18)	5 0,010	(0,08;0,12)	5 0,14	0,360
6	(2,30; 1,14)	5 0,009	(0,15;– 0,08)	5 0,17	0,130
7	(2,28; 1,15)	5 0,007	(0,05;0,08)	5 0,09	

Таким образом, получена точка $x_7 = (2{,}28;1{,}15)$, в которой модуль градиента меньше заданного $\varepsilon = 0{,}1$. Она и принимается за точку минимума. Отметим, что точкой точного минимума этой функции, как нетрудно убедиться, используя классические методы дифференциального исчисления, является точка $x^* = (2;1)$.

Отметим, что метод наискорейшего спуска обычно хорошо работает на первых шагах процесса минимизации, но вблизи стационарной точки он часто работает плохо, поскольку делаются маленькие шаги по направлениям. Поэтому вблизи стационарной точки вместо того, чтобы движение осуществлялось вдоль антиградиента, оно осуществляются вдоль специально подобранных направлений, на чем мы в данном учебном пособии останавливаться не будем. Читатель может ознакомиться с этими вопросами по специальной литературе.

3. Метод Ньютона. Метод Ньютона является методом второго порядка, т.е. использует вычисление и вторых частных производных минимизируемой функции $f(x)$.

Предположим, что функция $f(x)$ дважды дифференцируема на R^n, причем матрица Гессе не вырождена в любой точке R^n. В методе Ньютона последовательность точек $x_0, x_1, x_2, \ldots$ генерируется по следующим соображениям: для очередной точки x_k имеем приближение для этой функции, основанное на разложении Тейлора этой функции в окрестности точки x_k:

$$f(x) \approx f(x_k) + \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x_j - x_j^k) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j}(x_i - x_i^k)\cdot(x_j - x_j^k).$$

Для определения следующей точки x_{k+1} минимизируется функция $f_k(x)$, являющаяся квадратичной частью приращения $f(x) - f(x_k)$, т.е. решается задача

$$\min f_k(x),$$

где

$$f_k(x) = \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x_j - x_j^k) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j}(x_i - x_i^k)\cdot(x_j - x_j^k).$$

В векторно-матричной форме функция $f_k(x)$ имеет вид

$$f_k(x) = grad\, f(x_k)\cdot(x - x_k) + \frac{1}{2}(x - x_k)^T H(x_k)\cdot(x - x_k),$$

где $(x - x_k)$ – вектор столбец $\begin{pmatrix} x_1 - x_1^k \\ x_2 - x_2^k \\ \\ x_n - x_n^k \end{pmatrix}$, $H(x_k)$ – матрица Гессе, вычисленная в точке x_k и

$grad\, f(x_k)$ – градиент функции $f(x)$ в точке x_k.

В качестве точки x_{k+1} берется точка, в которой частные производные функции $f_k(x)$ равны нулю (необходимое условие оптимальности), т.е.

$$grad\, f_k(x) = 0 \text{ или } grad\, f(x_k) + H(x_k)\cdot(x - x_k) = 0,$$

откуда получаем

$$x_{k+1} = x_k - H^{-1}(x_k)\cdot grad\, f(x_k), \quad k = 0,1,2,\ldots$$

Это равенство дает рекуррентную формулу для точек, генерируемых методом Ньютона.

Предполагая, что $grad\, f(x^*) = 0$, $H(x^*)$ – положительно определенная матрица в точке локального минимума x^* и функция $f(x)$ дважды непрерывно дифференцируема, получаем, что $H(x_k)$ положительно определена в точках, близких к x^*, и, следовательно, точка x_{k+1} является точкой минимума для функции $f_k(x)$, являющейся квадратичной аппроксимацией данной функции $f(x)$.

Пример 14.12. Найти минимум функции $z = (x_1 - 2)^4 + (x_1 - 2x_2)^2$.

Решение. Как и в градиентном методе вычисления начинаем с точки $x_0 = (0;3)$, а на каждой следующей итерации значения координат новой точки x_{k+1} определяются по формуле

$$x_{k+1} = x_k - H^{-1}(x_k)\cdot grad\, f(x_k),\ k = 0,1,2,\ldots$$

Результаты вычислений по методу Ньютона представлены в табл. 14.9.

Таблица 14.9

Номер итерации	Точки x_k	Значения функции $f(x_k)$	Градиент Функции в точке x_k	Матрица Гессе $H(x_k)$	Обратная матрица $H^{-1}(x_k)$
0	(0,00; 3,00)	52,000	$(-44{,}00; 24{,}00)$	$\begin{pmatrix} 50 & -4 \\ -4 & 8 \end{pmatrix}$	$\frac{1}{384}\cdot\begin{pmatrix} 8 & -4 \\ -4 & 50 \end{pmatrix}$
1	(0,67; 0,33)	5 3,130	$(-9{,}39; -0{,}04)$	$\begin{pmatrix} 23{,}23 & -4{,}0 \\ -4{,}0 & 8{,}0 \end{pmatrix}$	$\frac{1}{169{,}84}\cdot\begin{pmatrix} 8 & 24{,}0 \\ 4 & 23{,}23 \end{pmatrix}$
2	(1,11; 0,56)	5 0,630	$(-2{,}84; -0{,}04)$	$\begin{pmatrix} 11{,}50 & -4{,}0 \\ -4{,}0 & 8{,}0 \end{pmatrix}$	$\frac{1}{76}\cdot\begin{pmatrix} 8 & 24{,}0 \\ 4 & 11{,}5 \end{pmatrix}$
3	(1,41; 0,70)	5 0,120	$(-0{,}80; -0{,}04)$	$\begin{pmatrix} 16{,}18 & -4{,}0 \\ -4{,}0 & 8{,}0 \end{pmatrix}$	$\frac{1}{33{,}44}\cdot\begin{pmatrix} 8 & 24{,}0 \\ 4 & 16{,}18 \end{pmatrix}$
4	(1,61; 0,80)	5 0,020	$(-0{,}22; -0{,}04)$	$\begin{pmatrix} 13{,}83 & -4{,}0 \\ -4{,}0 & 8{,}0 \end{pmatrix}$	$\frac{1}{14{,}64}\cdot\begin{pmatrix} 8 & 24{,}0 \\ 4 & 13{,}83 \end{pmatrix}$

5	(1,74; 0,87)	5 0,005	$(-0{,}07;-0{,}00)$	$\begin{pmatrix} 12{,}81 & -4{,}0 \\ -4{,}0 & 8{,}0 \end{pmatrix}$	$\dfrac{1}{6{,}48}\cdot\begin{pmatrix} 8 & 24{,}0 \\ 4 & 12{,}81 \end{pmatrix}$
6	(1,83; 0,91)	5 0,000	$(0{,}003;-0{,}04)$		

После шести итераций получена точка $x_6 = (1{,}83;0{,}91)$, которая и принимается за приближенное решение. В этой точке модуль градиента данной функции $\approx 0{,}04$, и процедура остановлена.

В этом примере значение функции от итерации к итерации убывает. В общем случае это не так. Метод Ньютона сходится лишь при условии, что процесс начинается из начальной точки, достаточно близкой к точке оптимума. Но такая информация, как правило, отсутствует. Следует отметить, что в общем случае может иметь место ситуация, когда $H(x_k)$ вырождена и x_{k+1} нельзя определить. И только в случае, когда начальная точка x_0 достаточно близка к точке минимума x^*, в которой $grad\, f(x^*) = 0$ и $H(x^*)$ невырожденная матрица, метод Ньютона, как доказано, сходится к x^*. Сложность определения хорошей начальной точки (начального приближения) x_0 является существенным недостатком метода Ньютона.

Имеются различные модификации метода Ньютона, которые гарантируют сходимость независимо от выбора начальной точки x_0. Обсудим следующие две модификации:

1) метод Ньютона с регулировкой шага;

2) метод Ньютона с аппроксимацией матрицы Гессе.

Метод Ньютона с регулировкой шага основан на том, что очередная точка x_{k+1} определяется по формуле

$$x_{k+1} = x_k + h_k g_k,\ k = 0,1,2,\ldots,$$

где

$$h_k > 0 \ \text{è} \ g_k = -H^{-1}(x_k)\cdot grad\, f(x_k).$$

При $h_k = 1$ он совпадает с методом Ньютона. Выбор коэффициента h_k производится из условия минимизации функции $f(x_k + h_k g_k)$ вдоль заданного направления g_k, что требует решения задачи минимизации функции одной переменной h_k.

Существуют и другие методы выбора коэффициента h_k. Доказано, что метод Ньютона с регулировкой шага сходится при любой начальной точке $x_0 \in R^n$. Метод Ньютона с регулировкой шага является одним из наиболее применяемых в вычислительной практике.

Другой эффективной модификацией метода Ньютона является аппроксимация матрицы $H(x_k)$, которая гарантирует сходимость метода независимо от выбора начальной точки x_0. В этой модификации очередная точка x_{k+1} генерируется по формуле

$$x_{k+1} = x_k + h_k g_k,\ k = 0,1,2,\ldots,$$

где

$$g_k = -B\cdot grad\, f(x_k).$$

Коэффициент h_k определяется также из условия минимизации данной функции вдоль направления g_k или как в методе Ньютона $h_k = 1$. В качестве матрицы B возьмем матрицу

$(\varepsilon \cdot E + H(x_k))^{-1}$, где E – единичная матрица, $H(x_k)$ – матрица Гессе в точке x_k. Величина $\varepsilon \ge 0$ определяется следующим образом: зафиксируем число $\delta > 0$, и пусть $\varepsilon \ge 0$ – наименьшее число, при котором все собственные значения матрицы $(\varepsilon \cdot E + H(x_k))$ больше или равны δ. Это обеспечивает положительную определенность матрицы B.

Существуют и другие способы построения матрицы B, на которых останавливаться не будем.

Модифицированные ньютоновские методы являются эффективным средством решения задач безусловной оптимизации, имеющих высокую скорость сходимости.

4. Методы сопряженных направлений. Высокая скорость сходимости метода Ньютона основана на том, что он минимизирует квадратичную функцию, являющуюся квадратичной аппроксимацией заданной функции $f(x)$ на каждой итерации. Эти методы позволяют найти минимум функции за определенное заданное число шагов.

Некоторые модификации метода Ньютона позволяют эффективно найти минимум за n шагов квадратичной функции

$$f(x) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j + \sum_{j=1}^{n} b_j x_j = \frac{1}{2}(Ax \cdot x) + (b \cdot x),$$

где $(Ax \cdot x)$ и $(b \cdot x)$ – скалярные произведения соответствующих векторов и A – симметрическая положительно определенная матрица.

Методы сопряженных направлений позволяют найти минимум квадратичной функции за конечное число шагов. В этих методах строятся такие направления $g_0, g_1, \ldots, g_{n-1}$, что последовательность n одномерных минимизаций вдоль этих направлений приводит к нахождению минимума квадратичной функции, т.е.

$$f(x^*) = f(x_n) = \min_{x \in R^n} f(x)$$

при любой начальной точке $x_0 \in R^n$ и

$$x_{k+1} = x_k + h_k g_k, \quad k = 0, 1, 2, \ldots, n-1,$$

а h_k на каждом шаге находим, решая задачу одномерной оптимизаци,

$$f(x_k + h_k g_k) = \min_h f(x_k + h g_k).$$

Оказывается, что указанным свойством обладает система взаимно сопряженных относительно матрицы A направлений.

Рассмотрим понятие сопряженных направлений. Пусть A – симметрическая положительно определенная матрица. Векторы g_1 и g_2 называются сопряженными относительно матрицы A, если они отличны от нуля и выполняется условие $(Ag_1 \cdot g_2) = 0$. Векторы $g_0, g_1, \ldots, g_k$ из R^n называются взаимно сопряженными относительно квадратной матрицы A размерности n, если все они отличны от нуля и $(Ag_i \cdot g_j) = 0$ при $i \ne j,\ 0 \le i, j \le k$. Установлено, что взаимно сопряженные векторы линейно независимы.

Проиллюстрируем на следующем примере понятие сопряженности и значение сопряженных направлений для минимизации квадратичных функций.

Пример 14.13. Найти точки минимума квадратичной функции

$$z = f(x_1, x_2) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 6x_2.$$

Решение. Матрицей Гессе заданной функции является $H = \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix}$. Построим два сопряженных направления g_1 и g_2. В качестве вектора g_1 выберем вектор $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Вектор $g_2 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ должен удовлетворять условию $Hg_1 \cdot g_2 = 0$.

$$Hg_1 = \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ -2 \end{pmatrix}.$$

Скалярное произведение векторов Hg_1 и g_2, равное $4x_1 - 2x_2$, должно быть равно нулю, т.е. векторы Hg_1 и g_2 должны быть перпендикулярными. Из условия $4x_1 - 2x_2 = 0$ получаем $x_2 = 2x_1$. Полагая $x_1 = 1$, получаем вектор $g_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$.

Пусть минимизация данной функции z начинается вдоль направления g_1 из начальной точки $x_0 = (-1/2;1)$. Найдем точку x_1 по формуле $x_1 = x_0 + h_1 g_1$, где h_1 находится из условия

$$\min_{h_1} f(x_0 + h_1 g_1) = \min_{h_1} f(x_1^1, x_2^1) = \min_{h_1} f\left(-\frac{1}{2} + h_1; 1\right) =$$

$$= \min_{h_1}\left(2\left(-\frac{1}{2} + h_1\right)^2 + 2 - 2\left(-\frac{1}{2} + h_1\right) - 6\right).$$

Минимум достигается при $h_1 = 1$. Тогда точка $x_1 = x_0 + g_1 = \left(\frac{1}{2}; 1\right)$. Теперь минимизируем данную функцию из точки x_1 по направлению g_2, т.е. находим новое значение h_2, для которого достигается

$$\min_{h_2} f(x_1 + h_2 g_2) = \min_{h_2} f(x_1^2, x_2^2) = \min_{h_2} f\left(-\frac{1}{2} + h_2 g_1^2; 1 + h_2 g_2^2\right) =$$

$$= \min_{h_2}\left(2\left(\frac{1}{2} + h_2\right)^2 + 2\cdot(1+2h_2)^2 - 2\left(\frac{1}{2} + h_2\right)\cdot(1+2h_2) - 6(1+2h_2)\right).$$

Получим $h_2 = 1/2$ и точку $x_2 = (1;2)$, которая является точкой минимума данной функции.

Таким образом, точка минимума найдена за две итерации (два шага). Сопряженные направления и путь к точке минимума показаны на рис. 14.12.

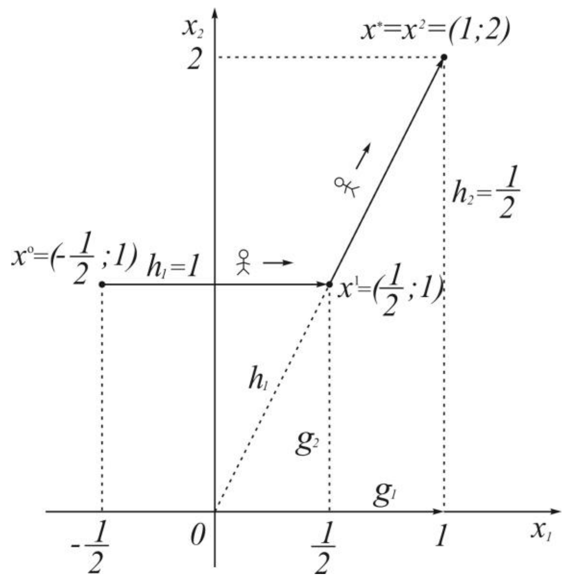

Рис. 14.12. Сопряженные направления

Нетрудно проверить, что, начиная с любой точки и минимизируя значение функции вдоль g_1 и g_2, точку минимума этой функции можно получить не более чем за два шага.

Известно, что если поиск ведется вдоль сопряженных относительно матрицы Гессе направлений, то точка минимума любой квадратичной функции может быть найдена не более чем за n шагов при произвольном выборе начальной точки x_0.

Поскольку произвольная функция может быть достаточно хорошо представлена в окрестности точки минимума ее квадратичной аппроксимацией, то понятие сопряженных направлений становится удобным инструментом для оптимизации и неквадратичных функций. Существует много различных методов построения сопряженных направлений. Одни из этих методов используют производные рассматриваемой функции, другие – только значения функции. Рассмотрим один из таких методов – метод сопряженных градиентов, относящийся к методам первого порядка, т.е. с использованием производных первого порядка.

Пусть требуется найти минимум дифференцируемой функции $f(x)$. Алгоритм сопряженных градиентов для минимизации дифференцируемой функции состоит в следующем.

1. На начальном этапе задается число $\varepsilon > 0$ для остановки алгоритма вычислений и начальная точка x_0. Вычисляем $grad\, f(x_0)$, полагаем $g_0 = -grad\, f(x_0)$ и переходим к основному этапу.
2. На основном этапе:
   Шаг 1. Проверяем условие $|grad\, f(x_k)| < \varepsilon$, если оно выполняется, то процесс останавливается и x_k принимается за точку минимума x^*, в противном случае переходим к шагу 2.
   Шаг 2. Как и в методе наискорейшего спуска находим число h_k, решая одномерную задачу минимизации $f(x_k + h_k g_k) = \min_h f(x_k + hg_k)$, и полагаем $x_{k+1} = x_k + h_k g_k$. Если $k < n$, то переходим к шагу 3, в противном случае – к шагу 4.

Шаг 3. Полагаем $g_{k+1} = -grad\, f(x_{k+1}) + \lambda_k g_k$, где $\lambda_k == \dfrac{|\, grad\, f(x_{k+1})\,|^2}{|\, grad\, f(x_k)\,|^2}$ и переходим к шагу 2, заменив k на $k+1$.

Шаг 4. Принимаем x_n за x_0, а $-grad\, f(x_n)$ за g_0 и переходим к шагу 1.

Метод сопряженных направлений отклоняет направление наискорейшего спуска путем добавления к нему с положительным коэффициентом λ_k направления g_k, используемого на предыдущем шаге.

Получение достаточно хорошего приближения к точке минимума в методе сопряженных градиентов гарантируется тем, что после каждого цикла расчетов, состоящего из n итераций, мы делаем обычный шаг наискорейшего спуска.

Пример 14.14. Найти минимум функции $z = f(x_1, x_2) = (x_1 - 2)^4 + (x_1 - 2x_2)^2$.

Решение. В данном примере $n = 2$. Следовательно, каждый цикл расчетов содержит две итерации. Выберем $\varepsilon = 0{,}1$ и $x_0 = (0;3)$. Результаты вычислений приведены в табл. 14.10.

Таблица 14.10

Номер цикла расчетов	Номер итерации k ($k \le 2$) в цикле	Точки $x_k = (x_1^k, x_2^k)$	Значение функции $f(x_k)$	Градиент функции	Числа λ_k
1	1	(0;3)	52,000	(−44; 24)	
	2	(2,7;1,51)	5 0,340	(0,73;1,28)	0,0009
2	1	(2,54;1,21)	5 0,100	(0,87; −0,48)	
	2	2,44;1,26)	5 0,040	(0,18;0,32)	0,1400
3	1	(2,25;1,1)	5 0,008	(0,16; −0,20)	
	2	(2,23;1,12)	5 0,003	(0,03;0,04)	0,0400

Продолжение таблицы 14.10

Номер цикла расчетов	Номер итерации k ($k \le 2$) в цикле	Направление g_k	Числа h_k	Точки x_{k+1}	Модуль градиента
1	1	(44; −24)	0,062	(2,70;1,51)	50,12
	2	(−0,69; −1,3)	0,230	(2,54;1,21)	5 1,47
2	1	(−0,87; 0,48)	0,110	(2,44;1,26)	5 0,99
	2	(−0,30; −0,25)	0,630	(2,25;1,1)	5 0,37
3	1	(−0,16; 0,20)	0,100	(2,23;1,12)	5 0,32
	2	(−0,030; −0,032)	1,020	(2,19;1,09)	5 0,05

Так как модуль градиента равен $0{,}05 < \varepsilon = 0{,}1$, то процесс вычислений останавливаем и точку (2,19;1,09) принимаем за точки минимума x^*, которая достаточно близка к точке $x^* = (2;1)$ минимума данной функции.

14.5. Задачи условной оптимизации

Задача оптимизации функции нескольких переменных на множестве K, являющемся подмножеством пространства R^n, называется задачей условной оптимизации. В общем виде ее можно представить так: найти $\min$ или $\max$ функции

$$z = f(x_1, x_2, \ldots, x_n) = f(x),$$

где $x = (x_1, x_2, \ldots, x_n) \in K, K \subset R^n.$

В дальнейшем будем рассматривать только задачу на минимум. Множество K в практических задачах описывается системой равенств и неравенств. Если K совпадает с R^n, то получаем задачу безусловной оптимизации, которая рассмотрена в предыдущем пункте. Здесь будем предполагать, что K – собственное подмножество пространства R^n.

Очевидно, что для задачи условной оптимизации имеют место классические необходимые и достаточные условия оптимальности (см. теоремы 14.1 и 14.2), если ее локальный минимум x^* является внутренней точкой множества K. Однако для многих задач условной оптимизации минимум целевой функции $z = f(x)$ достигается именно на границе множества K, в силу чего классические методы анализа не всегда применимы и тем самым все вопросы оптимизации для таких задач становятся более сложными.

В курсе высшей математики наряду с классическими методами решения задач безусловной оптимизации рассматривают и так называемые классические методы решения задач на условный экстремум, которые имеют вид

$$\min z = f_0(x),$$
$$f_i(x) = 0 \quad (i = \overline{1, m}),$$

т.е. множество K определяется как множество решений системы уравнений $f_i(x) = 0 \quad (i = \overline{1, m})$.

Заметим, что в принципе любую задачу нелинейного программирования можно преобразовать (см. п. 14.1) к задаче указанного здесь вида.

При исследовании этой классической задачи на условный экстремум важную роль играет ее функция Лагранжа:

$$L(x, u_0, u) = u_0 f_0(x) + \sum_{i=1}^{m} u_i f_i(x),$$

где $x = (x_1, x_2, \ldots, x_m) \in R^n$, u_0 – число, а $u = (u_1, u_2, \ldots, u_m) \in\in R^m$.

Частные производные функции Лагранжа по координатам x_j вектора x имеют вид

$$\frac{\partial L}{\partial x_j} = u_0 \frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} u_i \frac{\partial f_i}{\partial x_j} \quad (j = \overline{1, n}).$$

Необходимые условия локальной оптимальности содержатся в следующей теореме.

Теорема 14.3. *Пусть функции* $f_0(x), f_1(x), \ldots, f_m(x)$ *непрерывно дифференцируемы в окрестности точки* $x^* \in R^n$. *Если* x^* *– точка локального минимума, то существуют числа* $u_0^*, u_1^*, \ldots, u_m^*$, *не равные одновременно нулю, и такие, что*

$$\frac{\partial L}{\partial x_j} = u_0^* \frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} u_i^* \frac{\partial f_i}{\partial x_j} = 0 \quad (j = \overline{1, n})$$

в точке x^*. *Если при этом градиенты* $grad\, f_1(x^*), grad\, f_2(x^*), \ldots\ldots, grad\, f_m(x^*)$ *линейно* *независимы, то* $u_0 \neq 0$.

Линейная независимость градиентов функций $f_1(x),\ldots\ldots, f_m(x)$ в точке x^* называется условием регулярности. Условия $\frac{\partial L}{\partial x_j} = 0$ $(j=\overline{1,n})$ означают что градиенты функций $f_0(x)$ и $f_1(x), f_2(x),\ldots, f_m(x)$ в точке x^* линейно зависимы. В частности, если $m=1$, то $grad\, f_0(x^*)$ и $grad\, f_1(x^*)$ линейно зависимы (см. рис. 14.13).

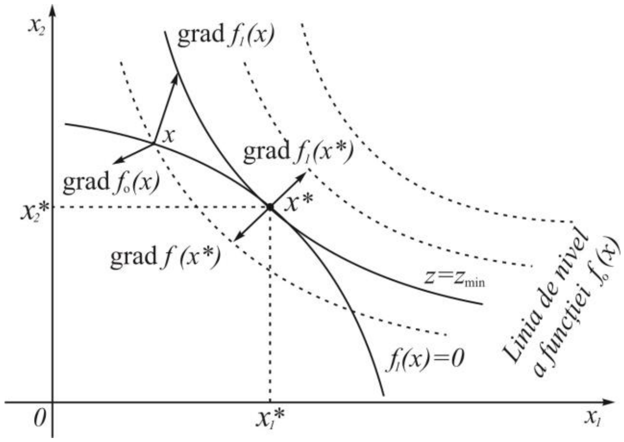

Рис. 14.13. Геометрическая интерпретация необходимого условия оптимальности в классической задаче на условный экстремум

На рис. 14.13 видно, что в точке x^* градиенты функций $f_0(x)$ и $f_1(x)$ коллинеарны, а в точке x нет, и эта точка x не может быть точкой минимума, так как из x можно сместиться, оставаясь на линии $f_1(x)=0$ так, что значение функции $f_0(x)$ уменьшится.

Числа $u_0^*, u_1^*, \ldots, u_m^*$ называются множителями Лагранжа. Любая допустимая точка x^*, удовлетворяющая необходимым условиям оптимальности задачи условной оптимизации, называется стационарной точкой этой задачи. Таким образом, стационарные точки определяются как решения системы из $n+m$ уравнений с $n+m+1$ неизвестными $x_1, x_2, \ldots, x_n$ и $u_0, u_1, u_2, \ldots, u_m$, т.е. системы

$$\frac{\partial L}{\partial x_j} = u_0 \frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} u_i \frac{\partial f_i}{\partial x_j} = 0 \quad (j=\overline{1,n}),$$

$$f_i(x) = 0 \quad (i=\overline{1,m}).$$

В случае $u_0^* \neq 0$ можно всегда считать $u_0^* = 1$. Для этого следует все множители Лагранжа $u_1^*, u_2^*, \ldots, u_m^*$ поделить на u_0^*. Таким образом, если выполняется условие регулярности, то функция Лагранжа имеет вид

$$L(x,u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x).$$

Отметим, что вышеуказанная система уравнений для определения стационарных точек и множителей Лагранжа может быть записана в виде

$$\frac{\partial L}{\partial x_j} = u_0 \frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} u_i \frac{\partial f_i}{\partial x_j} = 0 \quad (j = \overline{1,n}),$$

$$\frac{\partial L}{\partial u_i} = f_i(x) = 0 \quad (i = \overline{1,m}).$$

Во многих случаях решение этой системы уравнений в явном виде представляет собой весьма сложную задачу. Только в простейших случаях удается найти в явном виде решение указанной системы.

Однако не всегда стационарные точки обязаны быть решениями задачи. Для выбора из них оптимальных применяются достаточные условия с привлечением вторых производных. Эти условия сформулируем в виде следующей теоремы.

Теорема 14.4. *Пусть функции* $f_0(x), f_1(x), \ldots, f_m(x)$ *дважды дифференцируемые в стационарной точке* x^* *и пусть*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 L}{\partial x_i \partial x_j} dx_i\, dx_j > 0$$

при любых ненулевых приращениях dx_i, dx_j*, таких, что*

$$\sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} \cdot dx_j = 0 \quad (i = \overline{1,m}),$$

где $\frac{\partial^2 L}{\partial x_i \partial x_j}$ *и* $\frac{\partial f_i}{\partial x_j}$ *вычислены при* $x = x^*$ *и* $u_0^*, u_1^*, \ldots, u_m^*$. *Тогда* x^* *– точка строгого локального минимума рассматриваемой задачи.*

Пример 14.15. Найти точки минимума функции $z = x_1^2 + x_2^2$ при условии $x_1^3 + x_2^3 = 1$.

Решение. Условие регулярности здесь выполнено, так как имеется одно ограничение равенства и $grad\, z = (3x_1^2; 1) \neq (0;0)$ в любой точке R^2.

Выпишем систему уравнений

$$2pt\begin{cases} 2x_1 & + & 3x_1^2 u_1 & = & 0, \\ 2x_2 & + & 3x_2^2 u_1 & = & 0, \\ x_1^3 & + & 3x_2^3 & = & 1. \end{cases}$$

Она имеет три решения:

1) $x_1 = 0, x_2 = 1, u_1^* = -2/3$;

2) $x_1 = 1, x_2 = 0, u_1^* = -2/3$;

3) $x_1 = \frac{1}{\sqrt[3]{2}}, x_2 = \frac{1}{\sqrt[3]{2}}, u_1^* = -\frac{2\sqrt[3]{2}}{3}$.

Следовательно, в данной задаче мы имеем три стационарные точки.

Для нахождения точек минимума среди найденных стационарных точек составим матрицу из вторых частных производных по переменным x_1 и x_2. Получим

$$H = \begin{pmatrix} 2+6x_1u_1 & 0 \\ 0 & 2+6x_2u_1 \end{pmatrix},$$

а условие $\frac{\partial f_i}{\partial x_j} \cdot dx_j = 0$ выглядит как $3x_1 dx_1 + 3x_2 dx_2 = 0$.

Для стационарной точки $x_1 = 0, x_2 = 1, u_1^* = -\frac{2}{3}$ имеем $H = \begin{pmatrix} 2 & 0 \\ 0 & 6 \end{pmatrix}$ и $3dx_2 = 0$, откуда имеем $dx_2 = 0$, и тогда условие $\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 L}{\partial x_i \partial x_j} dx_i dx_j > 0$ имеет вид $2 \cdot (dx_1)^2 > 0$, что верно при любом $dx_1 \neq 0$. Следовательно, точка $(0;1)$ – точка локального минимума.

Аналогично нетрудно убедиться, что точка $(1;0)$ – также точка локального минимума.

Для третьей стационарной точки $x_1 = x_2 = \frac{1}{\sqrt[3]{2}}$ и $u_1^* = -\frac{2\sqrt[3]{2}}{3}$ получаем $H = \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix}$, и условие $3x_1^2 dx_1 + 3x_2^2 dx_2 = 0$ имеет вид $3 \cdot \frac{1}{\sqrt[3]{4}} dx_1 + 3 \cdot \frac{1}{\sqrt[3]{4}} dx_2 = 0$, или $dx_1 + dx_2 = 0$, откуда $dx_1 = -dx_2$. Условие $\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 L}{\partial x_i \partial x_j} dx_i dx_j > 0$ примет вид $-2(dx_1)^2 - 2(dx_2)^2 = -4(dx_1)^2 < 0$, и тогда условие теоремы 14.4 не выполняется. Следовательно, эта точка не является точкой локального минимума. Легко убедиться, что это точка является точкой локального максимума рассматриваемой функции, так как для точек максимума в теореме 14.4 условие $\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 L}{\partial x_i \partial x_j} dx_i dx_j > 0$ должно быть заменено на условие $\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 L}{\partial x_i \partial x_j} dx_i dx_j < 0$.

Таким образом, метод множителей Лагранжа включает в себя два этапа.

Этап 1. Составление и решение системы уравнений

$$\begin{cases} u_0 \frac{\partial f_0}{\partial x_j} + \sum_{i=1}^{m} u_i \frac{\partial f_i}{\partial x_j} = 0 \quad (j = \overline{1,n}), \\ f_i(x) = 0 \quad (i = \overline{1,m}) \end{cases}$$

для нахождения стационарных точек (необходимые условия оптимальности).

Этап 2. Исследование достаточных условий (см. теорему 14.4) для каждой найденной стационарной точки.

Метод множителей Лагранжа имеет ограниченное практическое применение в силу часто непреодолимых трудностей. Поэтому на практике используют приближенные численные методы решения задач условной оптимизации с учетом особенностей тех или иных типов задач условной оптимизации.

14.6. Задачи выпуклого программирования

Среди задач нелинейного программирования выделяют специальный тип задач условной оптимизации – задачи выпуклого программирования.

Задачу условной оптимизации

$$\min_{x \in K} z = f(x)$$

называют задачей выпуклого программирования, если:

– область K допустимых решений является выпуклой;

– целевая функция $f(x)$ является выпуклой.

Напомним, что множество K называется выпуклым, если вместе с любыми двумя его точками x_1 и x_2 оно содержит целиком и соединяющий их отрезок, т.е. для $\forall x_1, x_2 \in K$ $x = \alpha x_1 + (1-\alpha)x_2 \in K$, где $0 \le \alpha \le 1$.

Дадим определение выпуклых и вогнутых функций.

Определение 14.2. *Функция* $z = f(x)$, *определенная на непустом выпуклом множестве* $K \subset R^n$, *называется выпуклой, если для любых* $x_1, x_2 \in K$ *имеет место неравенство*

$$f(\alpha x_1 + (1-\alpha)x_2) \le \alpha f(x_1) + (1-\alpha)x_2 f(x_2), 0 \le \alpha \le 1.$$

Функция $z = f(x)$ *строго выпукла на* K, *если*

$$f(\alpha x_1 + (1-\alpha)x_2) < \alpha f(x_1) + (1-\alpha) f(x_2)$$

для любых $x_1, x_2 \in K$ *и* $\alpha \in [0;1]$.

Определение 14.3. *Функция* $z = f(x)$, *определенная на выпуклом множестве* K, *называется вогнутой (строго вогнутой), если функция* $-f(x)$ *выпукла (строго выпукла) на* K. *Для вогнутой функции имеем*

$$f(\alpha x_1 + (1-\alpha)x_2) \ge \alpha f(x_1) + (1-\alpha) f(x_2),$$

где $0 \le \alpha \le 1$.

Рассмотрим геометрическую интерпретацию сформулированных понятий. Пусть x_1 и x_2 – две различные точки из выпуклой области K. Рассмотрим точку $x = \alpha x_1 + (1-\alpha)x_2$, где $0 \le \alpha \le 1$, которая есть точка отрезка, соединяющего точки x_1 и x_2. Эта точка $x \in K$ в силу выпуклости множества K. Выражение $\alpha f(x_1) + (1-\alpha) f(x_2)$ дает средневзвешенное значение величин $f(x_1)$ и $f(x_2)$, а $f(\alpha x_1 + (1-\alpha)x_2)$ дает значение функции в точке x отрезка $[x_1, x_2]$. Для выпуклой функции значение $f(x)$ в точках отрезка $[x_1, x_2]$ не превосходит средневзвешенных с тем же α значений $f(x_1)$ и $f(x_2)$ функции на концах этого отрезка. Геометрически это означает, что точка $(x; f(x))$ графика выпуклой функции лежит ниже соответствующей точки отрезка, соединяющего точки $(x_1; f(x_1))$ и $(x_2, f(x_2))$ (рис. 14.14).

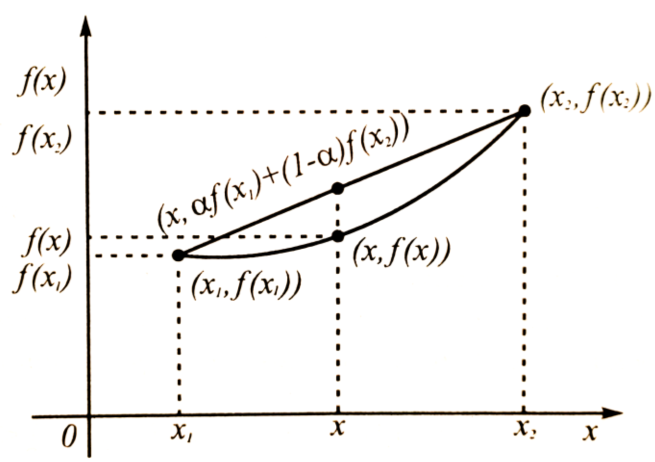

Рис. 14.14. Выпуклая функция одной переменной

На рис. 14.15 приведен график вогнутой функции одной переменной, а на рис. 14.16 – выпуклой и вогнутой функции двух переменных.

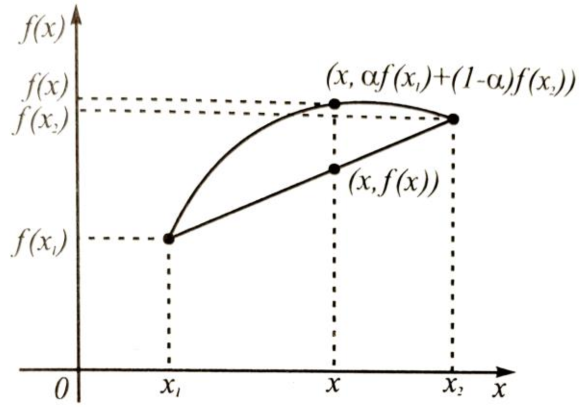

Рис. 14.15. Вогнутая функция одной переменной

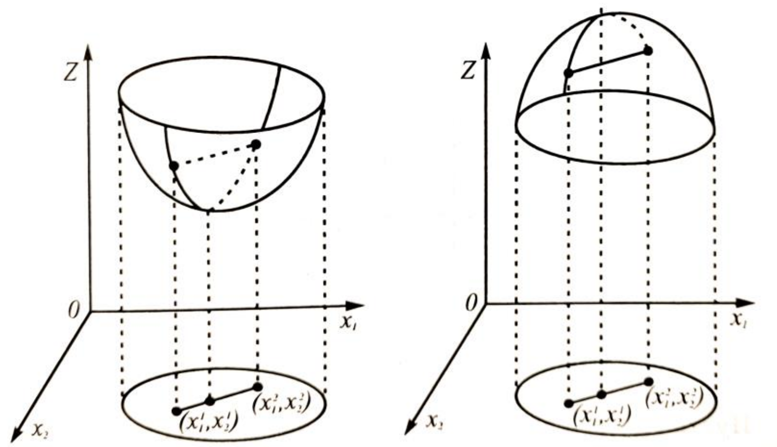

Рис. 14.16. Выпуклая и вогнутая функция двух переменных

На рис. 14.17 приведен график функции, не являющейся ни выпуклой ни вогнутой.

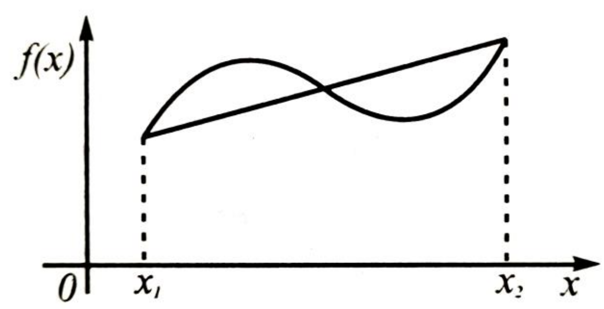

Рис. 14.17. Функция, не являющаяся ни выпуклой, ни вогнутой

Следующие функции являются выпуклыми всюду в R^n:

$$z = |x|; \quad z = x^2 - 2x; \quad z = f(x_1, x_2) = 2x_1^2 + x_2^2 - x_1 x_2;$$

$$z = f(x_1, x_2, x_3) = x_1^4 + 2x_2^2 + x_3^2 - 2x_1 - 2x_2 x_3.$$

Взяв эти функции с противоположными знаками, получим примеры вогнутых функций.

Нетрудно, привести примеры выпуклых функций не во всем пространстве, а только в некоторой области. Например, $f(x) = x^3$ выпуклая в области K, определенной неравенством $x \geq 0$, а во всем R^1 она не является выпуклой (рис. 14.18).

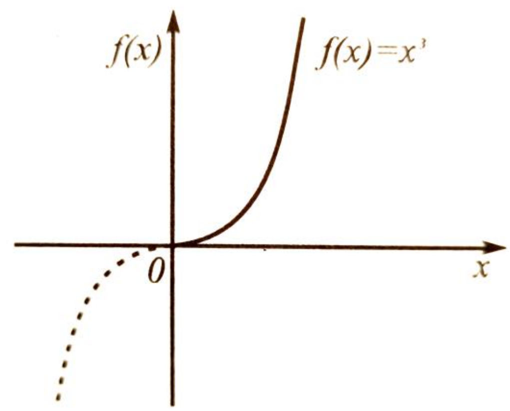

Рис. 14.18. График функции $f(x) = x^3$ при $x \geq 0$

Пусть K – непустое выпуклое множество в R^n и $f(x)$ – выпуклая функция на K. Тогда множество $S = \{x \in K; f(x) \leq c\}$, где c – константа (рис. 14.19) является выпуклым при любом действительном c. Напомним, что $f(x) = c$ есть поверхность уровня функции $f(x)$.

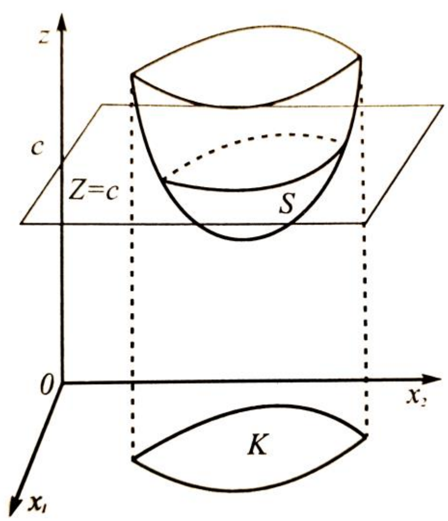

Рис. 14.19. Выпуклость множества $S = \{x \in K;\ f(x) \leq c\}$

Как известно, любая функция, определенная на K, может быть полностью описана множеством $G = \{x; f(x)\}$ пространства R^{n+1}, которое называется графиком функции $f(x)$. С графиком функции $f(x)$ связаны следующие два важных множества:

1) надграфик, или эпиграф функции $f(x)$, состоящий из всех точек, лежащих не ниже графика $f(x)$, и обозначаемый epi $f(x)$;

2) подграфик, или гиперграфик, состоящий из всех точек, лежащих выше графика $f(x)$, и обозначаемый hyp $f(x)$.

На рис. 14.20 изображены надграфик и подграфик выпуклой функции одной переменной.

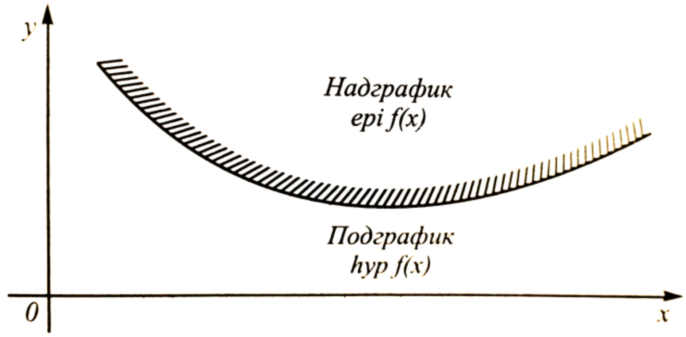

Рис. 14.20. Надграфик и подграфик выпуклой функции

Известно, что функция выпукла тогда и только тогда, когда ее надграфик является выпуклым множеством. Точно так же функция вогнута тогда и только тогда, когда ее подграфик является выпуклым множеством.

Поскольку надграфик выпуклой функции и подграфик вогнутой функции являются выпуклыми множествами, то в их граничных точках существуют опорные гиперплоскости, которые используются при определении понятия субградиента или обобщенного градиента, используемые при оптимизации недифференцируемых функций (см. гл. 15 настоящего учебного пособия). Для дифференцируемых выпуклых функций субградиент совпадает с ее градиентом.

Легко показать, что если $f(x) = \sum_{i=1}^{k} f_i(x)$, где $f_i(x)(i = \overline{1,k})$ – выпуклые (вогнутые) на некотором выпуклом множестве K функции, то $f(x)$ – тоже выпуклая (вогнутая) на K функция. Если выпуклая функция $f(x)$ дифференцируема, по крайней мере, один раз во внутренних точках множества K, и выпукла на нем, то можно получить следующий результат: из условия

$$f(\alpha x_1 + (1-\alpha)x_2) \le \alpha f(x_1) + (1-\alpha) f(x_2)$$

получаем

$$f(x_2 + \alpha(x_1 - x_2)) - f(x_2) \le \alpha(f(x_1) - f(x_2)),$$

откуда

$$\frac{f(x_2 + \alpha(x_1 - x_2)) - f(x_2)}{\alpha} \le f(x_1) - f(x_2).$$

По формуле Тейлора можно записать, что

$$f(x_2 + \alpha(x_1 - x_2)) = f(x_2) + grad\, f(x_2 + \alpha(x_1 - x_2)) \cdot \alpha(x_1 - x_2).$$

Тогда

$$\frac{f(x_2 + \alpha(x_1 - x_2)) - f(x_2)}{\alpha} =$$

$$= grad\, f(x_2 + \alpha(x_1 - x_2)) \cdot (x_1 - x_2) \le f(x_1) - f(x_2).$$

Переходя к пределу при $\alpha \to 0$, получаем

$$grad\, f(x_2) \cdot (x_1 - x_2) \le f(x_1) - f(x_2).$$

Это условие выполняется для любых внутренних точек x_1 и x_2 множества K. Оно является необходимым и достаточным условием выпуклости $f(x)$ на K.

Для задачи выпуклого программирования имеет место

Теорема 14.5. *Пусть* $z = f(x)$ *– выпуклая функция на выпуклом замкнутом множестве* K. *Тогда любой ее локальный минимум является глобальным.*

Доказательство. Пусть x^* – точка локального минимума функции $f(x)$, а x^{**} – точка глобального минимума, тогда $f(x^*) > f(x^{**})$. Соединим точки x^{**} и x^* отрезком.

Так как $f(x)$ – выпуклая функция, то для любого $0 \le \alpha \le 1$ справедливо неравенство

$$f(\alpha x^{**} + (1-\alpha)x^*) \le \alpha f(x^{**}) + (1-\alpha) f(x^*).$$

Множество K выпукло, поэтому точка $x = \alpha x^{**} + (1-\alpha)x^*$ $(0 \le \alpha \le 1)$ принадлежит этому множеству. Вышеуказанное неравенство можно усилить, заменив $f(x^{**})$ на $f(x^*)$. Тогда получим

$$f(x) = f(\alpha x^{**} + (1-\alpha)x^*) < f(x^*).$$

Значения α можно выбрать сколь угодно малыми и тогда точка $x = \alpha x^{**} + (1-\alpha)x^*$ будет расположена сколь угодно близко к точке x^*. Полученное неравенство $f(x) < f(x^*)$ противоречит тому, что x^* – точка локального минимума (в точке x, близкой к x^*, функция принимает меньшее значение, чем в точке x^*). Теорема доказана.

Из этой теоремы вытекают следующие следствия.

Следствие 14.1. *Если $f(x)$ – строго выпуклая функция на выпуклом множестве K, то ее глобальный минимум единственный.*

Следствие 14.2. *Если глобальный минимум выпуклой функции достигается в двух различных точках, то он достигается и в любой точке отрезка, соединяющего эти точки.*

Для выпуклой дифференцируемой функции $f(x)$ можно доказать, что если в точке x^* имеем $grad\, f(x^*) = 0$, то x^* есть точка ее локального, а следовательно, и глобального минимума.

Действительно, пусть x -- произвольная точка из окрестности точки x^* множества K, а x^* – точка, в которой $grad\, f(x^*) = 0$. Рассмотрим раннее полученное неравенство

$$grad\, f(x_2) \cdot (x_1 - x_2) \le f(x_1) - f(x_2).$$

Полагая $x_2 = x^*$, а $x_1 = x$, получаем

$$f(x_1) - f(x_2) \ge 0 \text{ или } f(x_1) \ge f(x_2) \quad (f(x^*) \le f(x)).$$

Следовательно, точка x^* – это точка локального, а следовательно, и глобального минимума выпуклой функции $f(x)$.

Очевидно, что если K – замкнутое ограниченное множество, то глобальный минимум выпуклой функции $f(x)$ достигается на нем в его крайних точках.

При вогнутой целевой функции $z = f(x)$ имеем задачу максимизации функции $z = f(x)$ на выпуклом множестве K. Рассмотренные результаты можно сформулировать следующим образом: любой локальный максимум вогнутой функции $f(x)$ на выпуклом замкнутом множестве K является и глобальным; если $f(x)$ строго вогнутая функция, то глобальный максимум единственный. Градиент вогнутой функции $f(x)$ в точках максимума равен нулю, если $f(x)$ дифференцируемая функция.

Отметим, что часто в практических задачах выпуклого программирования выпуклое множество K задается системой ограничений $f_i(x) \le b_i$ или $f_i(x) - b_i \le 0$ $(i = \overline{1,m})$, где функции $f_i(x)$ – выпуклые функции.

Множество S точек x, удовлетворяющих условию $f(x) \le b$, является выпуклым, если это множество не пусто. Таким образом, каждое неравенство $f_i(x) \le b_i \quad (i = \overline{1,m})$ задает выпуклое множество S_i, если $f_i(x)$ – выпуклая функция. Пересечение выпуклых множеств S_i выпукло. Таким образом, множество всех точек K (если K не пусто), удовлетворяющих условиям $f_i(x) \le b_i \quad (i = \overline{1,m})$, выпукло, если $f_i(x)$ – выпуклые функции. Если в задаче имеются и неравенства $x_j \ge 0$, то каждое такое неравенство задает полупространство, которое выпукло, и тогда множество всех допустимых решений K задачи выпуклого программирования также выпукло.

14.7. Теория двойственности в нелинейном программировании

Для любой задачи нелинейного программирования можно построить некоторую другую задачу нелинейной оптимизации, тесно связанную с исходной. Исходная задача называется прямой задачей, а вторая – двойственной к ней.

Рассмотрим задачу нелинейного программирования: найти

$$\min z = f_0(x),$$

при ограничениях

$$f_i(x) \le 0 \quad (i = \overline{1,m}),$$
$$x \ge 0.$$

Известны различные постановки задачи, называемой двойственной к данной. Среди них особое место занимает двойственная по Лагранжу задача. Она приводит к различным алгоритмам решения как линейных, так и нелинейных задач.

Рассмотрим функцию

$$L(x,u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x),$$

которая называется функцией Лагранжа для рассматриваемой задачи нелинейного программирования, где $u = (u_1, u_2, \ldots\ldots, u_m)$ – множители Лагранжа и $u_i \ge 0 \;\; (i = \overline{1,m})$.

Определим функции

$$F(x) = \max_{u \ge 0} L(x,u) \;\text{ и }\; G(u) = \min_{x \ge 0} L(x,u).$$

Тогда задачу отыскания минимума функции $F(x)$ при $x \ge 0$ называют прямой задачей, а задачу максимизации функции $G(u)$ при $u \ge 0$ – двойственной. Таким образом, прямая задача – это задача

$$\min_{x \ge 0} \max_{u \ge 0} L(x,u),$$

а двойственная – это задача

$$\max_{u \ge 0} \min_{x \ge 0} L(x,u).$$

Говорят, что точка (x^*, u^*) является седловой точкой функции Лагранжа $L(x,u)$ в области $x \ge 0, u \ge 0$, если

$$L(x^*,u) \le L(x^*,u^*) \le L(x,u^*)$$

для всех $x \ge 0$ и $u \ge 0$.

График функции двух переменных, имеющий седловую точку, изображен на рис. 14.21.

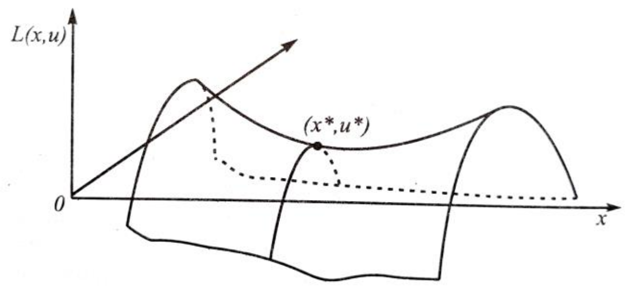

Рис. 14.21. График функции, имеющий седловую точку

Имеет место

Теорема 14.6. *Пусть* (x^*,u^*) *– седловая точка функции Лагранжа. Тогда* x^* *является решением рассматриваемой задачи нелинейного программирования, причем справедливо следующее условие дополняющей нежесткости:*

$$\sum_{i=1}^{m} u_i^* \cdot f_i(x^*) = 0.$$

Доказательство. По определению седловой точки имеем

$$f_0(x^*) + \sum_{i=1}^{m} u_i f_i(x^*) \le f_0(x^*) + \sum_{i=1}^{m} u_i^* f_i(x^*) \le f_0(x) + \sum_{i=1}^{m} u_i^* f_i(x).$$

Мы должны установить, что $x^* \in K$ (K – области допустимых решений) и $f_0(x^*) \le f_0(x)$ для всех $x \in K$.

Так как

$$f_0(x^*) + \sum_{i=1}^{m} u_i f_i(x^*) \le f_0(x^*) + \sum_{i=1}^{m} u_i^* f_i(x^*)$$

то

$$\sum_{i=1}^{m} u_i f_i(x^*) \le \sum_{i=1}^{m} u_i^* f_i(x^*)$$

при любых неотрицательных значениях $u_1, u_2, \ldots, u_m$. Отсюда получаем, что $f_i(x^*) \le 0$ для $i = \overline{1,m}$, т.е. $x^* \in K$. Если бы это было не так, т.е. нашлось бы такое ограничение k, что $f_k(x^*) >> 0$, то, взяв u_k большим положительным числом, а остальные $u_i = 0$, мы получили бы $u_k f_k(x^*) > \sum_{i=1}^{m} u_i^* f_i(x^*)$, что противоречит предыдущему неравенству. Таким образом, $x^* \in K$. Полагая все $u_i = 0$ $(i = \overline{1,m})$, получаем из неравенства $\sum_{i=1}^{m} u_i f_i(x^*) \le \sum_{i=1}^{m} u_i^* f_i(x^*)$, что $\sum_{i=1}^{m} u_i^* f_i(x^*) \ge 0$, а, с другой стороны, так как $u_i^* \ge 0$ $(i = \overline{1,m})$, а $f_i(x^*) \le 0$ $(i = \overline{1,m})$, то $\sum_{i=1}^{m} u_i^* f_i(x^*) \le 0$. Таким образом, $\sum_{i=1}^{m} u_i^* f_i(x^*) = 0$, т.е. условие дополняющей нежесткости выполнено.

Покажем, что $f_0(x^*) \le f_0(x)$. Из неравенства

$$f_0\left(x^*\right) + \sum_{i=1}^{m} u_i^* f_i\left(x^*\right) \le f_0(x) + \sum_{i=1}^{m} u_i^* f_i(x)$$

с учетом условия дополняющей нежесткости получаем

$$f_0(x^*) \le f_0(x) + \sum_{i=1}^{m} u_i^* f_i(x).$$

Но если $x \in K$, то $f_i(x^*) \le 0 \quad (i = \overline{1,m})$, и тогда $\sum_{i=1}^{m} u_i^* f_i(x) \le 0$, так как $u_i^* \ge 0 \quad (i = \overline{1,m})$. Отсюда, опуская слагаемое

$\sum_{i=1}^{m} u_i^* f_i(x)$, получаем, что $f_0(x^*) \le f_0(x)$ для всех $x \in K$. Теорема доказана.

Утверждение, обратное утверждению теоремы 14.6, имеет место лишь для задач выпуклого программирования, при дополнительном условии, что существует такое допустимое решение x^0, что $f_i(x^0) < 0$ (условие регулярности Слейтера).

Теорема 14.7 *(теорема Куна-Таккера). Точка x^* тогда и только тогда является оптимальным решением задачи вы пу кло го программирования*

$$\min z = f_0(x),$$
$$f_i(x) \le 0 \quad (i = \overline{1,m}),$$
$$x \ge 0,$$

когда существует такая точка $u^* \ge 0$, что (x^*, u^*) есть седловая точка функции Лагранжа, т.е.

$$L(x^*, u) \le L(x^*, u^*) \le L(x, u^*).$$

Доказательство этой теоремы опускаем.

Если $f_0(x)$ и $f_i(x)$ – дифференцируемые функции, то можно доказать, что условия теоремы 14.7 эквивалентны следующим условиям оптимальности Куна-Таккера:

1) $\dfrac{\partial L}{\partial x_j} \ge 0;$ 2) $x_j^* \cdot \dfrac{\partial L}{\partial x_j} = 0;$ 3) $x_j^* \ge 0 \ (j = \overline{1,n});$

4) $\dfrac{\partial L}{\partial u_i} \le 0;$ 5) $u_i^* \cdot \dfrac{\partial L}{\partial u_i} = 0;$ 6) $u_i^* \ge 0 \quad (i = \overline{1,m}),$

причем производные функции Лагранжа по x_j и u_i вычислены в точке (x^*, u^*).

Пример 14.16. Найти $\min z = x_1^2 + x_2^2$ при ограничениях

$$\begin{array}{ccc} 2x_1 + x_2 \ge 2, & & -2x_1 - x_2 + 2 \le 0, \\ 2x_1 + x_2 \le 8, & \text{или} & -2x_1 + x_2 - 8 \le 0, \\ 2x_1 + x_2 \le 6, & & -2x_1 + x_2 - 6 \le 0, \\ x_1 \ge 0,\ x_2 \ge 0, & & x_1 \ge 0,\ x_2 \ge 0. \end{array}$$

Решение. Графически нетрудно найти оптимальное решение: $x_1^* = 0{,}8,\ x_2^* = 0{,}4,\ z_{\min} = 0{,}8$. Очевидно, что условие Слейтера выполняется например для точки (1; 1) все ограничения строгие.

Покажем, что существует точка $u^* = (u_1^*, u_2^*, u_3^*)$ с неотрицательными координатами (множители Лагранжа), удовлетворяющая условиям Куна-Таккера для функции Лагранжа:

$$L(x,u) = x_1^2 + x_2^2 + u_1(2 - 2x_1 - x_2) + u_2(2x_1 + x_2 - 8) + u_3(x_1 + x_2 - 6).$$

Находим

$$\frac{\partial L}{\partial x_1} = 2x_1 - 2u_1 + 2u_2 + u_3,\ \frac{\partial L}{\partial x_2} = 2x_2 - u_1 + u_2 + u_3,$$

$$\frac{\partial L}{\partial u_1} = 2 - 2x_1 - x_2,\ \frac{\partial L}{\partial u_2} = 2x_1 + x_2 - 8,\ \frac{\partial L}{\partial u_3} = x_1 + x_2 - 6.$$

Так как $x_1^* = 0{,}8 > 0$, то $\frac{\partial L}{\partial x_1} = 0$, т.е. $2 \cdot 0{,}8 - 2u_1^* + 2u_2^* + u_3^* = 0$. Из того, что $x_2^* = 0{,}4 > 0$, следует, что $\frac{\partial L}{\partial x_2} = 0$, т.е. $2 \cdot 0{,}4 - u_1^* + u_2^* + u_3^* = 0$. Так как в точке (0,8; 0,4) $\frac{\partial L}{\partial u_2} = 2 \cdot 0{,}8 + 0{,}4 - 8 < 0$, то $u_2^* = 0$. Аналогично устанавливаем, что $u_3^* = 0$, и тогда условия $\frac{\partial L}{\partial x_1} = 0$ и $\frac{\partial L}{\partial x_2} = 0$ принимают вид

$$2 \cdot 0{,}8 - 2u_1^* = 0,$$
$$2 \cdot 0{,}4 - u_1^* = 0,$$

откуда получаем $u_1^* = 0{,}8$.

Итак, имеем точку $(x^*, u^*) = (0{,}8; 0{,}4; 0{,}8; 0; 0)$, которая есть седловая точка для функции Лагранжа. В этой точке (x^*, u^*) выполняются условия Куна-Таккера, и тогда $x^* = (0{,}8; 0{,}4)$ действительно является точкой ми ни му ма для рассматриваемой задачи нелинейного программирования.

Для более глубокого изучения теории двойственности в нелинейном программировании читателю рекомендуется обратиться к литературе по условной оптимизации.

Отметим, что прямая задача нахождения минимума функции $F(x) = \max\limits_{u \ge 0} L(x,u)$ равносильна исходной задаче нелинейного программирования, т.е. задаче

$$\min z = f_0(x),$$
$$f_i(x) \le 0 \quad (i = \overline{1,m}),$$
$$x \ge 0.$$

Действительно,

$$F(x) = \max_{u \ge 0} L(x,u) = \begin{cases} f_0(x), & \textit{если } x \in K \\ \infty, & \textit{если } x \notin K \end{cases}$$

Отсюда следует, что $\min F(x)$ находится в допустимой области K и совпадает с минимумом функции $f_0(x)$, т.е.

$$\min_{x \ge 0} \max_{u \ge 0} L(x,u) = \min_{x \in K} f_0(x).$$

Для пары двойственных задач имеют место следующие утверждения:

1) при любых $x \ge 0, u \ge 0$ имеет место неравенство

$$G(u) \le F(x);$$

2) если функция Лагранжа имеет седловую точку и выполнено условие Слейтера, то обе задачи имеют оптимальные решения и $\max\limits_{u \ge 0} G(u) = \min\limits_{x \ge 0} F(x)$.

Для задачи линейного программирования

$$\min z = \sum_{j=1}^{n} c_j x_j,$$

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

$$x_j \ge 0 \quad (j = \overline{1,n})$$

функция Лагранжа имеет вид

$$L(x,u) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} u_i (b_i - \sum_{j=1}^{n} a_{ij} x_j) =$$

$$= \sum_{j=1}^{n} (c_j - \sum_{i=1}^{m} a_{ij} u_i) x_j + \sum_{i=1}^{m} b_i u_i.$$

Тогда

$$G(u) = \min_{x \ge 0} L(x,u) = \min_{x \ge 0} \left[\sum_{j=1}^{n} \left(c_j - \sum_{i=1}^{m} a_{ij} u_i \right) x_j + \sum_{i=1}^{m} b_i u_i \right] =$$

$$= \begin{cases} \sum_{i=1}^{m} b_i u_i, \text{ если } \sum_{j=1}^{m} a_{ij} u_i \le c_j \quad (j = \overline{1,n}), \\ -\infty, \text{ в противном случае.} \end{cases}$$

Следовательно, $\max_{u \ge 0} G(u) = \max_{u \ge 0} \sum_{i=1}^{m} b_i u_i$ при ограничениях $\sum_{i=1}^{m} a_{ij} u_i \le c_j \quad (j = \overline{1,n})$, т.е. получаем известную в теории линейного программирования двойственную задачу. Таким образом, с помощью функции Лагранжа построена пара взаимно-двойственных задач линейного программирования (см. том I, гл. 2, п. 2.9).

Отметим, что в процессе формирования двойственных задач вместо условия $x \ge 0$ можно взять любое множество S, которое задается частью ограничений данной задачи нелинейного программирования, что приводит к неоднозначности полученных двойственных задач, т.е. двойственную задачу можно сформировать различными способами.

Изложенные идеи теории двойственности позволяют построить ряд важных численных методов решения задач нелинейного программирования.

Задача минимизации функции

$$F(x) = \max_{u \ge 0} L(x,u) \text{ при } x \in S,$$

эквивалентная задаче нелинейного программирования

$$\min z = f_0(x),$$

$$f_i(x) \le 0 \quad (i = \overline{1,m}),$$

$$x \in S,$$

кажется более привлекательной по сравнению с данной задачей нелинейного программирования, так как снимаются ограничения $f_i(x) \le 0$. Однако задача $\min F(x)$ может оказаться более сложной, так как операция взятия max для функции Лагранжа может привести к тому, что функция $F(x)$ будет недифференцируемой функцией, даже если функции $f_i(x)$ $(i = \overline{0,m})$ являются дифференцируемыми. Но, формируя множество S так, чтобы оно было простым и на S легко было бы осуществить необходимые операции, можно развить важные численные методы решения задач нелинейного программирования.

В заключении этого пункта приведем одно общее условие оптимальности в задачах математического программирования.

Рассмотрим общую задачу оптимизации

$$\min z = f(x),$$
$$x \in K.$$

Говорят, что вектор $g \in R^n$ задает направление убывания функции $f(x)$ в точке $x^* \in R^n$, если $f(x^* + hg) < f(x^*)$ при всех достаточно малых $h > 0$. Множество таких g обозначим $U(x^*, g)$. Вектор h задает возможное направление относительно множества K допустимых решений в точке $x^* \in K$, если $x^* + hg \in K$ при достаточно малых $h > 0$. Множество g возможных направлений обозначим $V(x^*, K)$.

Имеет место следующее необходимое условие локального минимума функции $f(x)$ на множестве K, не требующее никаких предположений относительно K и $f(x)$:

Теорема 14.8. *Если* x^* *– локальное решение задачи оптимизации, то* $U(x^*, g) \cap V(x^*, K) = .$

Доказательство. Допустим, что это условие не имеет места, тогда существует вектор $h \in R^n$, для которого $f(x^* + hg) \leq f(x^*)$ и $x^* + hg \in K$ при достаточно малых $h > 0$. Следовательно, в любой сколь угодно малой окрестности точки x^* найдется точка $x = x^* + hg \in K$, где $f(x) < f(x^*)$, а это противоречит тому, что x^* – точка локального минимума.

Это условие геометрически означает, что из точки x^* локального минимума нельзя осуществить сколь угодно малый сдвиг вдоль какого бы ни было луча так, чтобы уменьшить значение целевой функции, оставаясь при этом в множестве K.

Эта теорема лежит в основе ряда других более содержательных условий оптимальности при дополнительных предположениях относительно K и $f(x)$, часть из которых была изложена выше применительно к задаче выпуклого программирования. На основе понятия возможных направлений разработаны и численные методы решения задач условной оптимизации.

14.8. Задачи квадратичного программирования

Частным случаем задачи выпуклого программирования является задача квадратичного программирования, в которой ограничения $f_r(x_1, x_2, \ldots, x_n) \leq 0$ $(r = \overline{1, m})$ являются линейными $\sum_{j=1}^{n} a_{rj} x_j - b_r \leq 0$ $\left(\sum_{j=1}^{n} a_{rj} x_j \leq b_r\right)$, а целевая функция z представляет собой сумму линейной и квадратичной функции

$$z = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j \quad \left(d_{ij} = d_{ji}\right).$$

Квадратичная функция n переменных называется квадратичной формой и может быть записана в матрично-векторном виде:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_i x_j = x' D x,$$

где $x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$, $x' = (x_1, x_2, \dots x_n)$ и $D = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{pmatrix}$ – квадратная симметричная матрица $\left(d_{ij} = d_{ji}\right)$.

Если матрица D положительно определенная, то квадратичная форма $x'Dx$ представляет собой выпуклую функцию и целевая функция z является выпуклой функцией, так как она является суммой линейной функции (которая и выпукла и вогнута) и положительно определенной квадратичной формы. В этом случае мы и получаем частный случай задачи выпуклого программирования, для которой разработан ряд эффективных методов решения.

Запишем задачу квадратичного программирования в векторно-матричном виде:

$$\min z = c \cdot x + x'Dx,$$
$$Ax \le b,$$
$$x \ge 0,$$

где $c = (c_1, c_2, \dots, c_n)$, $x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$, $x' = (x_1, x_2, \dots, x_n)$, $b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}$, $A = \begin{pmatrix} a_{11} & a_{12} & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{mn} \end{pmatrix}$ и

$$D = \begin{pmatrix} d_{11} & d_{12} & d_{1n} \\ d_{21} & d_{22} & d_{2n} \\ \dots & \dots & \dots \\ d_{n1} & d_{n2} & d_{nn} \end{pmatrix}.$$

Для этой задачи квадратичного программирования функция Лагранжа имеет вид

$$L(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_m) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} x_i x_j + \sum_{r=1}^{m} u_r \left(\sum_{j=1}^{n} a_{rj} x_j - b_r \right),$$

или

$$L = cx + xDx + u \cdot (Ax - b).$$

Условия Куна-Таккера имеют вид

$$x_j \cdot \frac{\partial L}{\partial x_j} = c_j + \sum\nolimits_{i=1}^{n} d_{ij} x_i + u_r a_{rj} \ge 0 \qquad (j = \overline{1, n}),$$

$$x_j \cdot \frac{\partial L}{\partial x_j} = x_j \left(c_j + \sum\nolimits_{i=1}^{n} d_{ij} x_i + u_r a_{rj} \right) = 0 \qquad (j = \overline{1, n}),$$

$$x_j \cdot \frac{\partial L}{\partial u_r} = \sum\nolimits_{j=1}^{n} a_{rj} x_j - b_r \le 0 \qquad (r = \overline{1, m}),$$

$$u_r \cdot \frac{\partial L}{\partial u_r} = u_r \cdot \left(\sum\nolimits_{j=1}^{n} a_{rj} x_j - b_r \right) = 0 \qquad (r = \overline{1, m}).$$

Введем дополнительные переменные $v_j \ge 0$ так, чтобы $c_j + \sum\nolimits_{i=1}^{n} d_{ij} x_i + u_r a_{rj} - v_j = 0$, и $y_r \ge 0$

так, чтобы $\sum_{j=1}^{n} a_{rj}x_j - b_r + y_r = 0.$ Тогда условия Куна-Таккера можно записать в форме

$$x_j \cdot \frac{\partial L}{\partial x_j} = c_j + \sum_{i=1}^{n} d_{ij}x_i + a_{rj}u_r - v_j = 0 \qquad (j = \overline{1,n}).$$

$$x_j \cdot \frac{\partial L}{\partial x_j} = x_j \cdot v_j = 0 \qquad (j = \overline{1,n}),$$

$$x_j \cdot x_j \geq 0,\ v_j \geq 0 \qquad (j = \overline{1,n}),$$

$$x_j \cdot \frac{\partial L}{\partial u_r} = \sum_{j=1}^{n} a_{rj}x_j - b_r + y_r = 0 \qquad (r = \overline{1,m}),$$

$$u_r \cdot \frac{\partial L}{\partial u_r} = u_r \cdot y_r = 0 \qquad (r = \overline{1,m}),$$

$$x_j \cdot u_r \geq 0,\ y_r \geq 0 \qquad (r = \overline{1,m}).$$

Таким образом, чтобы найти решение задачи квадратичного программирования, нужно определить неотрицательное решение системы уравнений

$$\begin{cases} c_j + \sum\limits_{i=1}^{n} d_{ij}x_i + a_{rj}u_r - v_j = 0 \qquad (j = \overline{1,n}), \\ \sum\limits_{j=1}^{n} a_{rj}x_j + y_r = b_r \qquad (r = \overline{1,m}), \end{cases}$$

которое удовлетворяет условиям $x_j \cdot v_j = 0$ $(j = \overline{1,n})$ и $u_r \cdot y_r = 0$ $(r = \overline{1,m})$.

Это решение можно найти с помощью метода искусственного базиса задачи линейного программирования, примененного к полученной системе с учетом условий $x_j \cdot v_j = 0$ и $u_r \cdot y_r = 0$.

Процесс нахождения решения задачи квадратичного программирования с использованием условий Куна-Таккера включает следующие этапы:

- составление функции Лагранжа;
- запись необходимых и достаточных условий существования седловой точки для функции Лагранжа;
- решение задачи линейного программирования методом искусственного базиса и нахождение седловой точки функции Лагранжа, если она существует, или установление отсутствия такой точки;
- запись оптимального решения и нахождение оптимального значения целевой функции.

Рассмотрим пример решения задачи квадратичного программирования по указанной схеме.

Пример 14.17. Найти минимальное значение целевой функции

$z = x_1^2 + 2x_2^2 - 2x_1 - 4x_2$ при условиях

$$\begin{aligned} x_1 + 2x_2 &\leq 8, \\ 2x_1 - x_2 &\leq 12, \\ x_1 \geq 0,\ x_2 &\geq 0. \end{aligned}$$

Решение. Так как матрица $D = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ положительно определенная, то целевая функция является выпуклой, и мы имеем задачу квадратичного выпуклого программирования. Следовательно, можно воспользоваться теоремой Куна-Таккера.

Составим функцию Лагранжа

$$L = x_1^2 + 2x_2^2 - 2x_1 - 4x_2 + u_1(x_1 + 2x_2 - 8) + u_2(2x_1 - x_2 - 12).$$

Выпишем условие Куна-Таккера.

$$2pt\, x_j \cdot \begin{array}{l|l} \dfrac{\partial L}{\partial x_1} = 2x_1 - 2 + u_1 + 2u_2 \ge 0, & \dfrac{\partial L}{\partial u_1} = x_1 + 2x_2 - 8 \le 0, \\ \dfrac{\partial L}{\partial x_2} = 4x_2 - 4 + 2u_1 - u_2 \ge 0, & \dfrac{\partial L}{\partial u_2} = 2x_1 - x_2 - 12 \le 0, \end{array}$$

$$2pt \begin{array}{l|l} x_1 \cdot \dfrac{\partial L}{\partial x_1} = x_1 \cdot (2x_1 - 2 + u_1 + 2u_2) = 0, & u_1 \cdot \dfrac{\partial L}{\partial u_1} = u_1 \cdot (x_1 + 2x_2 - 8 = 0), \\ x_2 \cdot \dfrac{\partial L}{\partial x_2} = x_2 \cdot (4x_2 - 4 + 2u_1 - u_2) = 0, & u_2 \cdot \dfrac{\partial L}{\partial u_2} = u_2 \cdot (2x_1 - x_2 - 12) = 0, \end{array}$$

$$x_j \cdot x_1 \ge 0,\ x_2 \ge 0,\ u_1 \ge 0,\ u_2 \ge 0.$$

Систему линейных неравенств (первые четыре условия) перепишем в следующем виде:

$$2pt \begin{array}{ccccccccccc} 2x_1 & & & + & u_1 & + & 2u_2 & \ge & 2, \\ & & 4x_2 & + & 2u_1 & - & u_2 & \ge & 4, \\ x_1 & + & 2x_2 & & & & & \le & 8, \\ 2x_1 & - & x_2 & & & & & \le & 12. \end{array}$$

Вводим дополнительные неотрицательные переменные v_1, v_2, y_1, y_2 так, чтобы эти неравенства преобразовать в равенства. Получим систему

$$2x_1 + u_1 + 2u_2 - v_1 = 2,$$
$$4x_2 + 2u_1 - u_2 - v_2 = 4,$$
$$x_1 + 2x_2 + y_1 = 8,$$
$$2x_1 - x_2 + y_1 + y_2 = 12,$$
$$x_1 \ge 0, x_2 \ge 0, u_1 \ge 0, u_2 \ge 0, v_1 \ge 0, v_2 \ge 0, y_1 \ge 0, y_2 \ge 0$$

и условия $v_1 \cdot x_1 = 0, v_2 \cdot x_2 = 0, y_1 \cdot u_1 = 0, y_2 \cdot u_2 = 0.$

В первое и второе уравнение введем искусственные переменные w_1 и w_2, и рассмотрим задачу линейного программирования

$$\max T = -Mw_1 - Mw_2$$

при условиях

$$2x_1 + u_1 + 2u_2 - v_1 + w_1 = 2,$$
$$4x_2 + 2u_1 - u_2 - v_2 + w_2 = 4,$$

$$2x_1 + 2x_2 + y_1 = 8,$$
$$2x_1 - x_2 + y_1 + y_2 = 12,$$

$$x_1 \ge 0,\ x_2 \ge 0,\ u_1 \ge 0,\ u_2 \ge 0,\ v_1 \ge 0,\ v_2 \ge 0,\ y_1 \ge 0,\ y_2 \ge 0,\ w_1 \ge 0,\ w_2 \ge 0.$$

В результате решения этой задачи и получаем допустимое базисное решение системы линейных уравнений, причем при проведении симплексных пре об ра зо ва ний учитываем условия

$$v_1 \cdot x_1 = 0, v_2 \cdot x_2 = 0, y_1 \cdot u_1 = 0, y_2 \cdot u_2 = 0.$$

Расчеты по нахождению решения задачи квадратичного программирования приведены в табл. 14.11

Таблица 14.11

Базисные переменные	x_1	x_2	u_1	u_2	v_1	v_2	w_1	w_2	y_1	y_2	b
w_1	2	–0	–1	–2	–1	–0	1	–0	0	0	2
$\leftarrow w_2$	0	4	–2	–1	–0	–1	0	–1	0	0	2
y_1	1	–2	–0	–0	–0	–0	0	–0	1	0	4
y_2	2	–1	–0	–0	–0	–0	0	–0	0	1	12
T	$-2M$	$-4M_{\uparrow}$	$-3M$	$-M$	$-M$	$-M$	0	–0	0	0	$-6M$
w_1	2	–0	–1	–2	–1	–0	1	–0	0	0	2
x_2	0	–1	$-1/2$	$-1/4$	–0	$-1/4$	0	$-1/4$	0	0	1
y_1	1	–0	–1	$-1/2$	–0	$-1/2$	0	$-1/2$	1	0	6
y_2	2	–0	$-1/2$	$-1/4$	–0	$-1/4$	0	$-1/4$	0	1	13
T	$-2M$	–0	$-M$	$-2M$	$-M$	–0	0	$-M$	0	0	$-2M$
x_1	1	–0							0	0	1
x_2	0	–1							0	0	1
y_1	0	–0							1	0	5
y_2	0	–0							0	1	11
T	0	–0	0	0	0	0	$2M$	M	0	0	0

В результате получено допустимое базисное решение системы линейных уравнений $x_1^* = 1, x_2^* = 1, u_1^* = 0, u_2^* = 0, v_1^* = 0, v_2^* = 0, y_1^* = 5, y_2^* = 11.$

Так как $x_1^* \cdot u_1^* = 0, x_2^* \cdot u_2^* = 0, v_1^* \cdot y_1^* = 0, v_2^* \cdot y_2^* = 0$, то $(x^*, u^*) = (1;1;0;0)$ является седловой точкой функции Лагранжа для рассматриваемой задачи квадратичного программирования. Следовательно, $x_1^* = 1$ и $x_2^* = 1$ – оптимальное решение задачи, $z_{\min} = -3$.

Для решения задачи квадратичного программирования разработаны и другие методы решения, являющиеся обобщением симплексного метода решения задач линейного программирования.

14.9. Метод проекции градиента решения задач условной оптимизации

Рассмотрим общую задачу нелинейной оптимизации

$$\min z = f(x),$$
$$x \in K.$$

Если для ее решения использовать градиентные методы, то как было отмечено выше, направлением наискорейшего спуска является антиградиент целевой функции. Однако при наличии ограничений движение вдоль направления наискорейшего спуска по формуле

$$x_{k+1} = x_k - h_k grad\, f(x_k)$$

может привести в недопустимые точки. В методе проекции градиента, предложенного Розеном в 1960 году, антиградиент проектируется таким образом, что значение целевой функции улучшается и в тоже время сохраняется допустимость точек.

В методе проекции градиента в качестве очередной точки приближения к решению задачи выбирается проекция на множество K той точки, которая получается по градиентному методу

$$x_{k+1} = \prod_K \left(x_k - h_k grad\, f(x_k) \right).$$

Коэффициенты $h_k \geq 0$ выбираются на основе тех же правил, о которых шла речь в задачах безусловной оптимизации.

Введем понятие проекции точки a на множество X. Проекцией точки a на множество X называется точка $\prod_X(a)$, ближайшая к a среди всех точек из X. Иными словами $\prod_X(a)$ является решением так называемой задачи проектирования:

$$\min \varphi(x) = \| x - a \|^2 \text{ для } x \in X,$$

$$\text{где } \| x - a \|^2 = \sum_{j=1}^{n} (x_j - a_j)^2.$$

Если множество X замкнутое и выпуклое, то проекция любой точки a существует и она единственна (рис. 14.22).

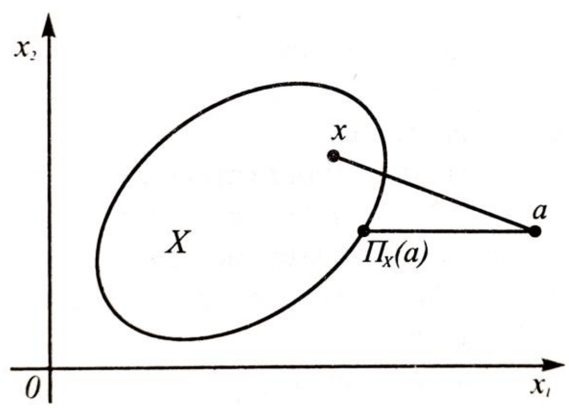

Рис. 14.22. Проекция точки на множество

В методе проекции градиента на каждой k-й итерации требуется производить проекцию точки $x_k - h_k grad\, f(x_k)$ на множество K, т.е. решать задачу проектирования, которая ничуть не проще исходной задачи нелинейного программирования, если множество K задается с помощью сложной системы равенств и неравенств. Но в некоторых случаях, если K – просто устроенное множество (шар, неотрицательный ортант, полупространство или многогранник), удается указать явный вид проекции. В сложных задачах условной оптимизации прибегают к модификациям метода проекции градиента, в которых в окрестности очередной точки проектирование на множество K заменяют проектированием на многогранники, аппроксимирующие множество K.

Отметим также, что, используя идею проектирования можно модифицировать и другие методы безусловной оптимизации, например метод сопряженных градиентов.

Рассмотрим метод проекции градиента в случае линейных ограничений. Прежде всего приведем определение матрицы проектирования.

Определение. *Квадратная матрица* P *порядка* $n \times n$ *называется матрицей проектирования, если* $P = P'$ *и* $P \cdot P = P$, *где* P' *– транспонированная матрица*.

Имеют место следующие утверждения:

1. Если P – матрица проектирования, то она положительно полуопределенная матрица.

2. Чтобы P была матрицей проектирования, необходимо и достаточно, чтобы и $E - P$ была матрицей проектирования, где E – единичная матрица.

3. Пусть P – матрица проектирования и $Q = E - P$. Тогда подпространства $L_1 = \left\{Px : x \in E^n\right\}$ и $L_2 = \left\{Qx : x \in E^n\right\}$ являются ортогональными подпространствами, скалярное произведение любых векторов p и q равно нулю, где $p \in L_1$ и $q \in L_2$. Кроме того, любой вектор $x \in E^n$ может быть представлен однозначно в виде $p + q$ ($x = p + q$), где $p \in L_1$ и $q \in L_2$ (рис. 14.23).

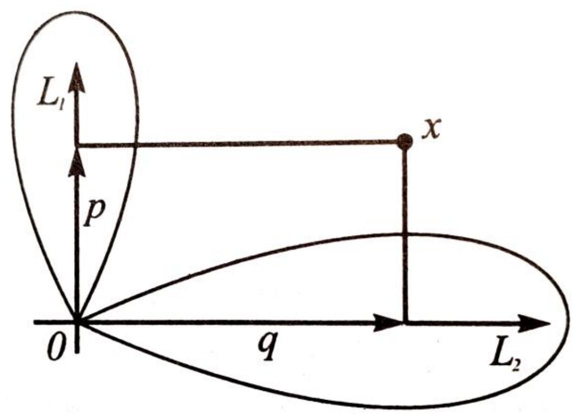

Рис. 14.23. Ортогональные подпространства L_1 и L_2 для трехмерного пространства E^3

Рассмотрим задачу нелинейного программирования с линейными ограничениями, т.е. найти

$$\min z = f(x)$$

при условиях

$$Ax \le b,$$

$$Cx = d,$$

где A – матрица $m \times n$, а C – матрица $l \times n$, b – m-мерный вектор, а d – l-мерный вектор, а функция $f(x)$ выпуклая дифференцируемая функция. В заданной допустимой точке x направлением наискорейшего спуска является вектор антиградиент, т.е. $-grad\, f(x)$. Однако движение вдоль антиградиента может нарушить допустимость. Поэтому следует спроектировать вектор $-grad\, f(x)$ так, чтобы можно было двигаться вдоль направления $g = -P \cdot grad\, f(x)$, где P – соответствующая матрица проектирования, которая строится по специальному правилу.

Не вдаваясь в подробности обоснования метода проекции градиента для рассматриваемого типа задач, приведем лишь общую схему алгоритма метода проекции градиента.

Начальный этап. Выбрать допустимое решение x_1, т.е. такую точку x_1, для которой $Ax_1 \le b$ и $Cx_1 = d$. Представить матрицу A в виде $\begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ и вектор b в виде $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ соответственно, где $A_1 x_1 = b_1$, $A_2 x_1 < b_2$, т.е. выделить активные ограничения $A_1 x_1 = b_1$ и неактивные $A_2 x_1 < b_2$. Положить $k = 1$ и перейти к основному этапу, состоящему из двух шагов (процедур).

Основной этап. Шаг 1. Положить $M = \begin{pmatrix} A_1 \\ C \end{pmatrix}$. Если M пуста, т.е. не содержит ни одного столбца, то положить матрицу проектирования $P = E$. В противном случае

$$P = E - M' \cdot (MM')^{-1} \cdot M.$$

Вычислить вектор $g_k = -P \cdot grad\, f(x_k)$. Если $g_k \ne 0$, то перейти к шагу 2. Если же $g_k = 0$ и M пуста, то остановиться. Если же $g_k = 0$, но M непуста, то вычислить вектор $w = -(MM')^{-1} M\, grad\, f(x_k)$. Пусть $w = \begin{pmatrix} u \\ v \end{pmatrix}$, где размерность u соответствует размерности вектора b_1 активных ограничений. Если $u \ge 0$, то x_k – оптимальное решение, в противном случае следует выбрать отрицательную компоненту этого вектора и вычеркнуть соответствующую строку матрицы A_1, получив новую матрицу, которая принимается за матрицу A_1 (аналогично поступаем с вектором b_1), и повторить шаг 1.

Шаг 2. Взять в качестве h_k оптимальное решение задачи

$$\min_h f(x_k + h \cdot g_k)$$

при условиях $0 \le h \le h_{\max}$, где $h_{\max}$ определяется условиями исходной задачи, т.е. необходи мо решить систему линейных неравенств относительно переменной h

$$A(x_k + hg_k) \le b,$$
$$C(x_k + hg_k) = d.$$

Положить $x_{k+1} = x_k + h_k g_k$ и представить A и b в виде $\begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ и $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, так что $A_1 x_{k+1} = b_1$ и $A_2 x_{k+1} < b_2$. Заменить k на $k+1$ и перейти к шагу 1.

Пример 14.18. Найти

$$\min z = 2x_1^2 + 2x_2^2 - 2x_1 x_2 - 4x_1 - 6x_2$$

при условиях

$$\begin{array}{rcrcl} x_1 & + & x_2 & \le & 2, \\ x_1 & + & 5x_2 & \le & 5, \\ -x_1 & & & \le & 0, \\ & - & x_2 & \le & 0. \end{array}$$

Решение. Для функции z имеем $grad\, z(x) = \begin{pmatrix} 4x_1 - 2x_2 - 4 \\ -2x_1 + 4x_2 - 6 \end{pmatrix}$.

Начальный этап. В качестве начального допустимого решения выберем точку $x^1 = \begin{pmatrix} x_1^1 \\ x_2^1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. На каждой итерации сначала найдем на прав ле ние движения, т.е. вектор q_1 (шаг 1), а затем проведем линейный поиск минимума функции z вдоль этого направления.

Итерация 1 $(k=1)$. *Поиск направления* (*шаг 1*). В точке $x^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ имеем $grad\, z(x^1) = \begin{pmatrix} -4 \\ -6 \end{pmatrix}$. В этой точке активными ограничениями являются ограничения неотрицательности, т.е. $-x_1 \le 0$ и $-x_2 \le 0$. Таким образом, матрица $A_1 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ и $A_2 = \begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix}$. Матрица $M = A_1$.

Тогда $P = E - A_1' \cdot (AA_1') \cdot A_1 =$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \left[\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \right]^{-1} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

и вектор $g_1 = -P \cdot grad\, z(x^1) = -\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ -6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, т.е. вектор $g_1 = 0$.

Вычислим вектор $w = u = -(A_1 A_1')^{-1} A_1 \cdot grad\, z(x^1) =$

$$= \left[\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \right]^{-1} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ -6 \end{pmatrix} =$$

$$= -\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ -6 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ -6 \end{pmatrix} = \begin{pmatrix} -4 \\ -6 \end{pmatrix}.$$

Отметим, что вектор w совпадает с вектором u, так как ограничения равенства в исходной задаче отсутствуют. Вектор w имеет отрицательные координаты. Выберем одну из них, а именно $u_2 = -6$. Тогда в матрице $A = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ вычеркиваем вторую строку и получаем новую матрицу $A_1 = (-1;0)$. Преобразованная матрица проектирования имеет вид

$$P = E - A_1'(A_1 A_1')^{-1} A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Новый вектор $g_1 = -P\,grad\,z(x^1) = -\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ -6 \end{pmatrix} = -\begin{pmatrix} 0 \\ -6 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}$ отличен от нуля и, следовательно, найдено направление движения.

Линейный поиск (*шаг 2*). Любая точка x^2 полученная из x^1 по направлению g_1 имеет вид $x^2 = x^1 + hg_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + h\begin{pmatrix} 0 \\ 6 \end{pmatrix} = \begin{pmatrix} 0 \\ 6h \end{pmatrix}$.

Значение целевой функции

$$z(x^2) = 2 \cdot 36h^2 - 6 \cdot 6h = 72h^2 - 36h.$$

Максимальное значение h, для которого точка $x^2 = \begin{pmatrix} 0 \\ 6h \end{pmatrix}$ допустима, определяется из условий $6h \le 2,\ 5 \cdot 6h \le 5,\ -6h \le 0$. Отсюда $h_{\max} = \min\left\{\frac{2}{6}; \frac{5}{30}\right\} = \frac{1}{6}$. Следовательно, $h_1 = \frac{1}{6}$ и $x^2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Итерация 2. Поиск направления (*шаг 1*). В точке $x^2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ $grad\,z(x^2) = \begin{pmatrix} -6 \\ -2 \end{pmatrix}$. В этой точке активными является второе ограничение $x_1 + 5x_2 \le 5$ и третье ограничение $-x_1 \le 0$. Следовательно, матрица $A_1 = \begin{pmatrix} 15 \\ -10 \end{pmatrix}$ и $A_2 = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}$.

Определяем матрицу P.

$$P = E - A_1' \cdot (A_1 A_1')^{-1} \cdot A_1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

и направление $g_2 = -P \cdot grad\,z(x^2) = (0,0)$. Вычислим вектор

$$w = u = -(A_1 A_1')^{-1} \cdot A_1 \cdot grad\,z(x^2) =$$

$$= -\left[\begin{pmatrix} 15 \\ -10 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 \\ 5 & 0 \end{pmatrix}\right]^{-1} \cdot \begin{pmatrix} 15 \\ -10 \end{pmatrix} \cdot \begin{pmatrix} -6 \\ -2 \end{pmatrix} =$$

$$= \begin{pmatrix} 26 & -1 \\ -1 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 15 \\ -10 \end{pmatrix} \cdot \begin{pmatrix} -6 \\ -2 \end{pmatrix} = \begin{pmatrix} \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{26}{25} \end{pmatrix} \cdot \begin{pmatrix} 15 \\ -10 \end{pmatrix} \cdot \begin{pmatrix} -6 \\ -2 \end{pmatrix} =$$

$$= \begin{pmatrix} 0 & \frac{1}{25} \\ -1 & \frac{1}{5} \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 2 \end{pmatrix} = \begin{pmatrix} \frac{2}{5} \\ -\frac{28}{5} \end{pmatrix}.$$

Так как вектор u имеет отрицательную координату $-\frac{28}{5}$, то в матрице A_1 вычеркиваем строку $(-1;0)$ и получаем матрицу $A_1 = (1;5)$. Находим матрицу проектирования и соответствующее направление:

$$P = E - A'_1(A_1 A'_1)^{-1} \cdot A_1 = \begin{pmatrix} \frac{25}{26} & -\frac{5}{26} \\ -\frac{5}{26} & \frac{1}{25} \end{pmatrix},$$

$$g_2 = -P\,grad\,z(x^2) = \begin{pmatrix} \frac{70}{13} \\ -\frac{14}{13} \end{pmatrix}.$$

Так длина вектора g_2 не играет никакой роли, а важно лишь направление, то рассмотрим вектор $\begin{pmatrix} 5 \\ -1 \end{pmatrix}$, колинеарный вектору g_2. Окончательно полагаем $g_2 = \begin{pmatrix} 5 \\ -1 \end{pmatrix}$.

Линейный поиск (*шаг 2*). Рассматриваем лишь точку $x^3 = x^2 + hg_2 = (5h; 1-h)$. Значение функции $z(x^3) = 62h^2 - 28h - 4$.

Максимальное значение h определяется из условий

$$5h + 1 - h \le 2, \quad 5h + 5(1-h) \le 5, \text{ откуда получаем } h_{\max} = \frac{1}{4}.$$

Решаем задачу

$$\min z(h) = 62h^2 - 28h - 4,$$

$$\text{при условии} \quad 0 \le h \le \frac{1}{4}.$$

Оптимальным решением является $h_2 = \frac{7}{31}$, так что $x^3 = \left(\frac{35}{31}; \frac{24}{31}\right)$.

Итерация 3. Поиск направления. В точке $x^3 = \left(\frac{35}{31}; \frac{24}{31}\right)$ $grad\,z(x^3) = \begin{pmatrix} -\frac{32}{31} \\ -\frac{160}{31} \end{pmatrix}$. В этой точке активным является второе ограничение и тогда $A_1 = (1;5)$ и $A_2 = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}$.

Находим

$$P = E - A'_1 \cdot (A_1 A'_1)^{-1} \cdot A_1 = \begin{pmatrix} \frac{25}{26} & -\frac{5}{26} \\ -\frac{5}{26} & \frac{1}{26} \end{pmatrix},$$

и $g_3 = -P grad\,z(x^3) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Вычисляем $w = u = (A_1 A'_1)^{-1} A_1 \cdot grad\,z(x^3) = \frac{32}{31} > 0$. Следовательно, точка $x^3 = \left(\frac{35}{31}; \frac{24}{31}\right)$ является оптимальным решением. Так функция z в этом примере выпукла, то точка x^3

является точкой глобального минимума задачи и $z_{\min} = -7{,}16$.

Геометрическая интерпретация решения задачи приведена на рис. 14.24.

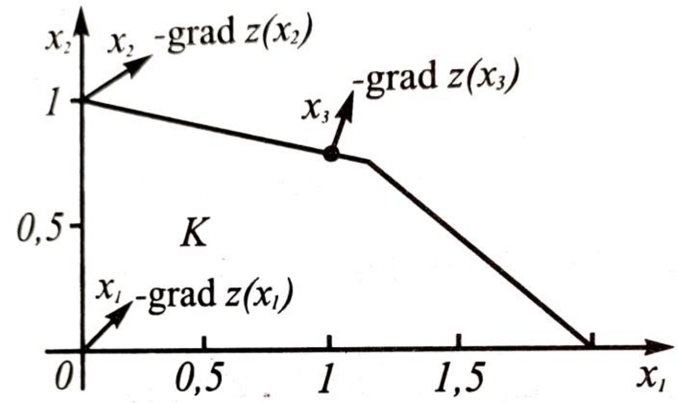

Рис. 14.24. Метод проекции градиента

Отметим, что метод проекции градиента был распространен и при наличии нелинейных ограничений.

Имеются и другие процедуры для построения возможного направления спуска, например метод приведенного градиента решения задач нелинейного программирования с линейными ограничениями.

14.10. Метод штрафных функций

Метод штрафных функций основан на преобразовании исходной задачи с ограничениями в одну или в последовательность задач безусловной оптимизации. С помощью функций, задающих ограничения, строится так называемая штрафная функция, которая добавляется к целевой функции исходной задачи так, что нарушение какого-либо ограничения становится невыгодным при решении полученной задачи безусловной оптимизации. Рассмотрим введение штрафной функции для задачи с одним ограничением равенством, т.е. для задачи

$$\min z = f(x)$$

при условиях

$$g(x) = 0,$$

$$x \in E^n.$$

Преобразуем ее в задачу безусловной оптимизации

$$\min \Phi(x) = f(x) + \mu \cdot g^2(x)$$

при $x \in E^n$, где $\mu > 0$ – большое число. Ясно, что на оптимальном решении x^* последней задачи $g^2(x^*)$ должно быть близким к нулю.

Для задачи с единственным ограничением неравенством $g(x) \le 0$ нецелесообразно вводить штраф в виде $\mu g^2(x)$, так как при $g(x) \ne 0$ штраф будет взиматься независимо от знака $g(x)$; штраф должен взиматься лишь в точках, в которых $g(x) > 0$. Поэтому в этом случае приемлемой задачей безусловной оптимизации является задача

$$\min \Phi(x) = f(x) + \mu \cdot \max\{0; g(x)\}$$

$$\text{при } x \in E^n.$$

В этом случае при $g(x) = 0$ имеем $\max\{0; g(x)\} = 0$, и штраф не будет взиматься; если же $g(x) > 0$, то $\max\{0; g(x)\} > 0$, и взимается штраф величиной $\mu g(x)$. Введенные функции $g^2(x)$ и $\max\{0; g(x)\}$ являются примерами штрафных функций.

Таким образом, вводимая штрафная функция должна определять положительный штраф в недопустимых точках и не штрафовать допустимые точки.

В общей задаче нелинейного программирования

$$\min z = f(x)$$

при ограничениях

$$g_i(x) \le 0 \quad (i = \overline{1, m}),$$

$$h_k(x) = 0 \quad (k = \overline{1, l}),$$

штрафная функция имеет вид

$$\alpha(x) = \sum_{i=1}^{m} \varphi(g_i(x)) + \sum_{k=1}^{l} \psi(h_i(x)),$$

где φ и ψ – неопределенные функции одной переменной, удовлетворяющие условием

$$\varphi(y) = 0, \text{ если } y \le 0 \text{ и } \varphi(y) > 0, \text{ если } y > 0;$$

$$\psi(y) = 0, \text{ если } y = 0 \text{ и } \psi(y) > 0, \text{ если } y \ne 0.$$

Типичными такими функциями φ и ψ являются функции вида $\varphi(y) = (\max\{0; y\})^p$ и $\psi(y) = |y|^p$, где p – целое положительное число. В этом случае штрафная функция имеет вид

$$\alpha(x) = \sum_{i=1}^{m} \left(\max\{0; g_i(x)\}\right)^p + \sum_{k=1}^{l} |h_i(x)(x)|^p,$$

и тогда задача безусловной оптимизации для общей задачи нелинейного программирования имеет вид

$$\min \Phi(x) = f(x) + \mu \cdot \alpha(x)$$

$$\text{при } x \in E^n,$$

где $\alpha(x)$ – штрафная функция.

Пример 14.19. Найти $\min z = x$ при условии $-x + 3 \le 0$.

Решение. Введем штрафную функцию

$$\alpha(x) = \begin{cases} 0, \text{ если } x \ge 3, \\ (-x+3)^2, \text{ если } x < 3, \end{cases}$$

тогда задача безусловной оптимизации будет иметь вид

$$\min \Phi(x) = x + \mu \cdot \alpha(x).$$

На рис. 14.25 изображен график штрафной функции $\alpha(x)$, а на рис. 14.26 – график функции $\Phi(x)$ при $\mu = 1$.

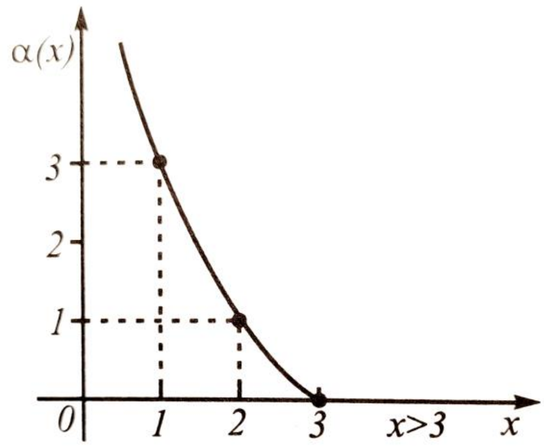

Рис. 14.25. График штрафной функции $\alpha(x)$ задачи безусловной оптимизации

Отметим, что минимум функции $\Phi(x)$ достигается в точках $x^* = 3 - -\dfrac{1}{2\mu}$. При $\mu \to \infty$ последовательность таких точек стремится к $x^* = 3$, которая является точкой минимума целевой функции исходной задачи, в чем нетрудно убедиться решая исходную задачу графически (см. рис. 14.27).

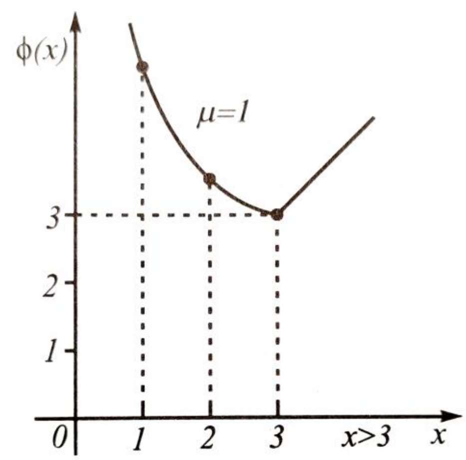

Рис. 14.26. График функции $\Phi(x)$ задачи безусловной оптимизации

Заметим, что последовательность точек при возрастании параметра μ ($\mu > 0$) стремится к точке $x^* = 3$ из вне допустимой области, т.е. при $x < 3$.

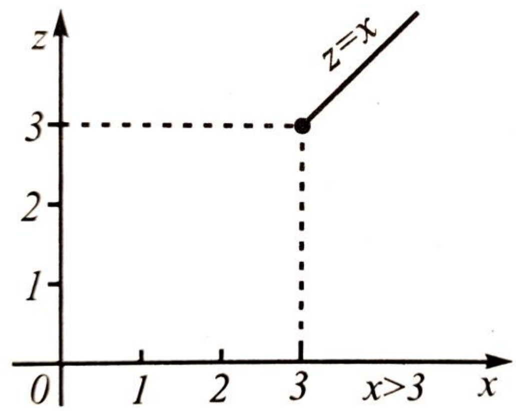

Рис. 14.27. Графическое решение исходной задачи

Пример 14.20. Найти $\min z = x_1^2 + x_2^2$ при условии $x_1 + x_2 = 2$.

Решение. Нетрудно убедиться, что минимум целевой функции достигается в точке $x^* = (1;1)$ и равен 2.

Построим задачу безусловной оптимизации со штрафом при достаточно большом μ:

$$\min \Phi(x) = x_1^2 + x_2^2 + \mu(x_1 + x_2 - 2)^2.$$

При любом $\mu \geq 0$ функция $\Phi(x)$ является выпуклой и, следовательно, необходимым и достаточным условием оптимальности является равенство нулю градиента функции $\Phi(x)$:

$$2x_1 + 2\mu(x_1 + x_2 - 2) = 0,$$
$$2x_2 + 2\mu(x_1 + x_2 - 2) = 0.$$

Решая эту систему уравнений, получаем $x_1 = x_2 = \dfrac{2\mu}{1+2\mu}$. При $\mu \to \infty$ получаем точку минимума $x^* = (1;1)$. Следовательно, решение задачи безусловной оптимизации может быть приведено как угодно близко к решению исходной задачи при выборе достаточного большого μ.

Приведенные примеры показывают, что можно сколь угодно близко подойти к оптимальному значению целевой функции исходной задачи, вычисляя значение функции $\Phi(x)$ при достаточно больших μ.

Приведем общую схему алгоритма метода штрафных функций для решения задачи минимизации функции $z = f(x)$ при условиях $g_i(x) \leq 0$ $(i = \overline{1,m})$, $h_k(x) = 0$ $(k = \overline{1,l})$. Используется штрафная функция вида

$$\alpha(x) = \sum_{i=1}^{m} \left(\max\{0, g_i(x)\}\right)^2 + \sum_{k=1}^{l} h_i^2(x).$$

Этот метод не накладывает каких-либо ограничений на функции $f(x), g_i(x)$ и $h_k(x)$, помимо их непрерывности.

Начальный этап. Выбрать $\varepsilon > 0$ в качестве критерия остановки процесса вычислений. Выбрать начальную точку x_1, величину штрафного параметра μ_1 и число $\beta > 1$. Положить $k = 1$ и перейти к основному этапу.

Основной этап. Шаг 1. При начальной точке x_k решить задачу безусловной оптимизации

$$\min \Phi(x) = f(x) + \mu_k \alpha(x).$$

Положить, что x_{k+1} равен оптимальному решению этой задачи, и перейти к шагу 2.

Шаг 2. Если $\mu_k \alpha(x_{k+1}) < \varepsilon$, то остановиться; в противном случае положить $\mu_{k+1} = \beta\mu_k$. Заменить k на $k+1$ и перейти к шагу 1.

Кроме метода штрафных функций, для решения задач нелинейного программирования с ограничениями-неравенствами используют метод барьеров, суть которого заключается в преобразовании задачи с ограничениями в задачу безусловной оптимизации с помощью введения специальных барьерных функций, которые как бы препятствуют выходу рассматриваемой точки из допустимой области. Рассмотрим, как исходная задача преобразуется в задачу безусловной оптимизации с помощью введения барьерной функции.

Пусть дана задача

$$\min z = f(x),$$

при

$$g_i(x) \le 0 \quad (i = \overline{1, m}),$$

где функции f и $g_1, g_2, \ldots, g_m$ – непрерывные функции n переменных. Введем барьерную функцию $B(x)$, которая должна обладать свойствами неотрицательности и непрерывности внутри допустимой области и стремящаяся к бесконечности при приближении изнутри к границе области K. Типичным примером барьерной функции является функция

$$B(x) = \sum_{i=1}^{m} \frac{-1}{g_i(x)}.$$

Тогда исходная задача сводится к задаче безусловной оптимизации функции

$$\varphi(x) = f(x) + \mu B(x),$$

где $\mu > 0$.

Пример 14.21. Так, для задачи

$$\min z = x$$

при условии

$$x \ge 3 \quad (-x + 3 \le 0)$$

барьерной функцией будет функция $B(x) = \dfrac{-1}{-x+3}$, для $x \ne 3$. Для любого заданного $\mu > 0$ задача безусловной оптимизации заключается в минимизации функции $\varphi(x) = x - \dfrac{\mu}{-x+3} = x + \dfrac{\mu}{x-3}$. Любой метод безусловной оптимизации для минимизации функции $\varphi(x) = x + \dfrac{\mu}{x-3}$ при начальной точке $x_1 > 3$ приведет к оптимальному решению $x_\mu^* = 3 + \sqrt{\mu}$. Очевидно, что при $\mu \to 0$ $x_\mu^* \to x^* = 3$, т.е. к точке минимума исходной задачи.

Схема алгоритма решения задачи нелинейного программирования методом барьеров аналогична схеме штрафных функций, но начальная точка x^1 должна быть внутренней точкой допустимой области K.

* * *

В заключении этой главы отметим, что решение той или иной задачи нелинейного программирования представляет собой достаточно сложный процесс, требующий учета особенностей данной задачи и выбора метода решения (точного или приближенного). Отметим, что каждый алгоритм решения задачи характеризуется областью его применения, вычислительными особенностями и сходимостью последовательности приближений к оптимальному решению. В данной главе рассмотрены лишь основы нелинейного программирования без подробного обоснования рассмотренных методов. Совсем не затронуты важные, но достаточно трудные, вопросы сходимости приближенных методов решения нелинейных задач.

Все эти сложные вопросы читатель при необходимости сможет уяснить, обратившись к специальной литературе по нелинейному программированию.