

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

**Донбасская государственная машиностроительная академия**

# **МЕТОДЫ ОПТИМИЗАЦИИ И ИССЛЕДОВАНИЯ ОПЕРАЦИЙ**

(МОДУЛИ 3-4)

## **КОНСПЕКТ ЛЕКЦИЙ**

(для студентов направления «Системный анализ» всех форм обучения)

**Краматорск 2014**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

**Донбасская государственная машиностроительная академия**

# **МЕТОДЫ ОПТИМИЗАЦИИ И ИССЛЕДОВАНИЯ ОПЕРАЦИЙ**

(МОДУЛИ 3-4)

## **КОНСПЕКТ ЛЕКЦИЙ**

(для студентов направления «Системный анализ» всех форм обучения)

**Краматорск 2014**

Конспект лекций по курсу «Методы оптимизации и исследования операций» (для направления 6.040303 «Системный анализ») всех форм обучения / Сост.: Гитис В.Б. – Краматорск: ДГМА, 2014. – 60 с.

В конспекте лекций приводятся теоретические сведения по курсу «Методы оптимизации и исследования операций» (модули 3-4).

Составитель

В.Б. Гитис, к.т.н., доцент каф. ИСПР

Отв. за выпуск

В.Б.Гитис, к.т.н., доцент каф. ИСПР

# СОДЕРЖАНИЕ

1	Методы исследования операций	6
2	Оптимизация математических моделей	6
2.1	Постановка задачи оптимизации	6
2.2	Понятие экстремума. Виды экстремумов	7
2.3	Системы ограничений в математических моделях	8
2.4	Классификация методов оптимизации	10
2.5	Необходимое и достаточное условие экстремума. Понятие выпуклости и вогнутости	13
3	Одномерная оптимизация	17
3.1	Численные методы поиска экстремума функции одной переменной	17
3.2	Метод сканирования (метод перебора)	18
3.3	Методы последовательного поиска нулевого порядка	19
3.3.1	Стратегия последовательного поиска	19
3.3.2	Метод половинного деления (метод дихотомии)	21
3.3.3	Метод золотого сечения	22
3.4	Методы поиска с использованием полиномиальной аппроксимации	24
3.5	Методы поиска с использованием производных	26
3.5.1	Метод средней точки (метод Больцано)	26
3.5.2	Метод секущих (метод хорд)	26
3.5.3	Метод Ньютона-Рафсона	28
4	Многомерная безусловная оптимизация	28
4.1	Общая стратегия многомерной оптимизации	28
4.2	Методы оптимизации нулевого порядка	30
4.2.1	Метод покоординатного спуска (метод Гаусса-Зейделя)	30
4.2.2	Метод вращающихся координат (метод Розенброка)	31
4.2.3	Метод параллельных касательных (метод Пауэлла)	33
4.3	Методы оптимизации первого порядка (градиентные методы)	34
4.3.1	Принцип градиентного спуска	34
4.3.2	Численное дифференцирование функций	35
4.3.3	Градиентный метод с переменным шагом	37

4.3.4	Метод наискорейшего спуска (метод Коши)	38
4.3.5	Партан-метод	40
4.3.6	Метод тяжелого шарика	40
4.3.7	Методы сопряженных градиентов	41
4.3.8	Метод Левенберга-Марквардта	43
4.4	Методы оптимизации второго порядка	44
4.4.1	Метод Ньютона	45
4.4.2	Метод Ньютона-Рафсона	45
4.4.3	Квазиньютоновские методы	47
5	Многомерная условная оптимизация	49
5.1	Метод множителей Лагранжа	49
5.2	Условия Куна-Таккера	52
5.3	Методы штрафных функций	52
5.4	Методы случайного поиска	54
5.4.1	Метод случайного поиска с пересчетом	55
5.4.2	Метод случайного поиска по наилучшей пробе (поиск по статистическому градиенту)	56
6	Методы глобальной оптимизации	57
	Список рекомендуемой литературы	59

# 1 МЕТОДЫ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

Первая часть курса «Методы оптимизации и исследования операций» – «Методы исследования операций» была изложена в модулях 1 и 2.

## 2 ОПТИМИЗАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

### 2.1 ПОСТАНОВКА ЗАДАЧИ ОПТИМИЗАЦИИ

В процессе решения задачи оптимизации обычно необходимо найти оптимальные значения вектора параметров  $X = (x_1, x_2, \dots, x_n)$ , определяющих данную задачу. Число параметров характеризует размерность (и степень сложности) задачи оптимизации.

Под *оптимизацией* понимают процесс выбора наилучшего варианта из всех возможных.

Система, для которой показатель её качества имеет экстремальное значение, называется *оптимальной*.

Выбор оптимального решения или сравнение двух альтернативных решений проводится с помощью некоторой зависимой величины (функции), определяемой проектными параметрами. Эта величина называется *целевой функцией*. Целевая функция определяет правило предпочтения одного варианта другому. В процессе решения задачи оптимизации должны быть найдены такие значения проектных параметров, при которых целевая функция имеет минимум (или максимум). Таким образом, *целевая функция* – это глобальный критерий оптимальности в математических моделях, с помощью которых описываются инженерные или экономические задачи. Целевую функцию можно записать в виде  $u = f(x_1, x_2, \dots, x_n)$ .

В случае одного проектного параметра ( $n = 1$ ) целевая функция является функцией одной переменной, и ее график – некоторая кривая на плоскости. При ( $n = 2$ ) целевая функция является функцией двух переменных, и ее графиком является поверхность. При  $n > 2$  – гиперповерхность.

Целевая функция должна быть однозначной функцией проектных параметров.

## 2.2 ПОНЯТИЕ ЭКСТРЕМУМА. ВИДЫ ЭКСТРЕМУМОВ

Функция  $f(x)$  достигает *локального* минимума или максимума в точке  $X^*$ , если для всех точек  $x$ , лежащих в малой окрестности точки  $X^*$ , имеют место неравенства:

$$\begin{cases} f(X^*) \leq f(x) & \text{для минимума, } x \in \delta X^*; \\ f(X^*) \geq f(x) & \text{для максимума, } x \in \delta X^*. \end{cases}$$

*Глобальным* экстремумом называется точка  $X^*$ , в которой минимизируемая (максимизируемая) функция  $f(x)$  имеет наименьшее (наибольшее) значение среди всех локальных экстремумов (рисунок 2.1).

$$\begin{cases} f(X^*) \leq f(x) & \text{для минимума, } x \in D_x; \\ f(X^*) \geq f(x) & \text{для максимума, } x \in D_x. \end{cases}$$

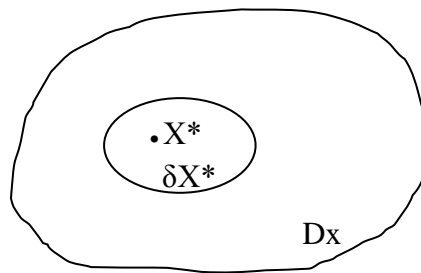


Рисунок 2.1 – Экстремум  $X^*$  в области определения  $D_x$

Функция  $f(x)$  называется *одноэкстремальной* (унимодальной), если она имеет один экстремум, и *многоэкстремальной* (мультимодальной), если она имеет более одного минимума или максимума.

Задача нахождения максимума функции сводится к отысканию минимума заменой исходной функции  $f(x)$  на  $-f(x)$ , т.к. выполняется равенство  $\max f(x) = -\min(-f(x))$ . Графически идентичность решения задач максимизации и минимизации показана на рисунке 2.2. Поэтому в дальнейшем будут рассматриваться только задачи минимизации. Тогда задача отыскания оптимальных параметров записывается в виде  $\min f(x) = f(X^*) = f^*$ .

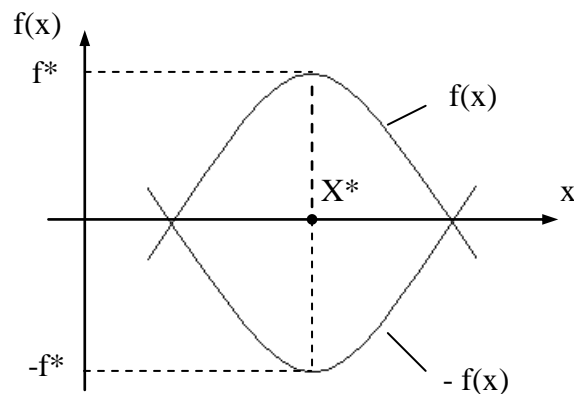


Рисунок 2.2 – Идентичность решения задач максимизации и минимизации

## 2.3 СИСТЕМЫ ОГРАНИЧЕНИЙ В МАТЕМАТИЧЕСКИХ МОДЕЛЯХ

В задачах оптимизации чаще всего присутствуют ограничения, которые сужают область поиска  $Dx$ . Эти ограничения задаются совокупностью некоторых функций, удовлетворяющих уравнениям или неравенствам.

В результате ограничений область проектирования  $Dx$ , определяемая всеми  $n$  проектными параметрами, может быть существенно уменьшена.

Различают ограничения *прямые* и *функциональные*. *Прямые* имеют вид

$$X_{\min i} \leq X_i \leq X_{\max i},$$

где  $i = 1 \dots n$ .

*Функциональные* ограничения представляют собой условия работоспособности выходных параметров, которые не вошли в целевую функцию. Функциональные ограничения имеют вид

$$G(x) \geq 0, H(x) = 0,$$

где  $G$  и  $H$  – векторы функций.

$$\begin{cases} G(x) = (g_1(x), g_2(x), \dots, g_k(x)); \\ H(x) = (h_1(x), h_2(x), \dots, h_m(x)). \end{cases}$$

Например, если известно, что  $y(x) \leq 1$  или  $y(x) = 10$ , то функциональные ограничения соответственно запишутся в виде



$$g(x) = 1 - y(x) \geq 0 \text{ и } h(x) = y(x) - 10 = 0.$$

Прямые ограничения можно рассматривать как частный случай функциональных.

Ограничения формируют допустимую область поиска – область работоспособности:

$$Dx = \{x | G(x) \geq 0, H(x) = 0\}.$$

Любая из точек  $x \in Dx$  называется *допустимым решением задачи оптимизации* (или *планом*).

Допустимое решение, которое оптимизирует целевую функцию, называется *оптимальным решением* (или *оптимальным планом*).

Теория и методы решения задач оптимизации при наличии ограничений составляют предмет исследования одного из разделов математики – *математического программирования*.

Таким образом, задача управления, формализованная в виде задачи математического программирования, имеет вид

$$\begin{cases} f(x) \rightarrow \min (\max), x \in Dx; \\ G(x) \geq 0; \\ H(x) = 0; \\ x_{\min i} \leq x_i \leq x_{\max i}. \end{cases}$$

Особенностью отыскания решения при наличии ограничений является то, что оптимальное решение здесь может соответствовать либо локальному экстремуму (максимуму или минимуму) внутри области проектирования, либо значению целевой функции на границе области. Если же ограничения отсутствуют, то ищется оптимальное решение на всей области проектирования, т. е. глобальный экстремум.

Функция  $f(x)$  называется *ограниченной сверху (снизу)* на множестве  $Dx$ , если существует число  $B$  такое, что  $f(x)$  меньше (больше)  $B$  при всех  $x \in Dx$ . Функция  $f(x)$  называется *ограниченной* на множестве  $Dx$ , если она ограничена на  $Dx$  сверху и снизу.

Примером ограниченной сверху функции может служить функция  $f(x)$ , представленная на рисунке 2.2. Тогда  $-f(x)$  – функция, ограниченная снизу. Классический пример ограниченной функции – гармоническая функция.

## 2.4 КЛАССИФИКАЦИЯ МЕТОДОВ ОПТИМИЗАЦИИ

Практически все основные задачи математического программирования в соответствии с типом оптимизируемых параметров, а также видом функциональных зависимостей, входящих в математическую модель, могут быть представлены определенными классами задач, в каждом из которых целесообразно применять те или иные методы поиска решения (рисунок 2.3).

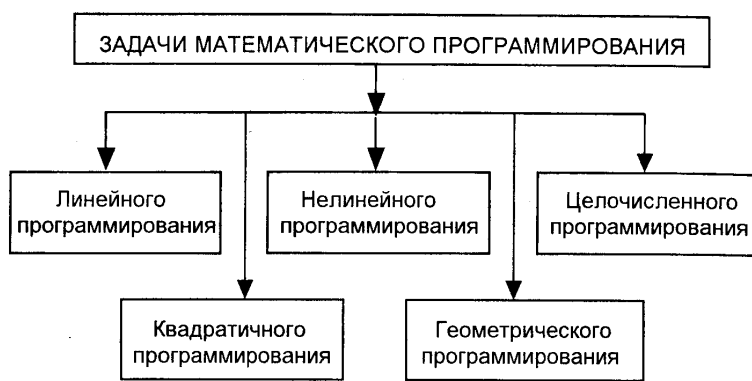


Рисунок 2.3 – Классификация задач математического программирования

К задачам *линейного* программирования относятся те, в которых в математическую модель входят только линейные зависимости (и критерий, и ограничения).

Задачи *нелинейного* программирования содержат в математической модели хотя бы одну нелинейную зависимость.

В задачах *целочисленного* программирования переменные или какая-то их часть должны принимать целые значения.

Кроме задач целочисленного программирования, еще два типа задач имеют специальную структуру. Это *квадратичное* программирование, к которому относят задачи с квадратичной целевой функцией и линейными ограничениями, и *геометрическое* программирование, исследующее оптимизационные задачи, в которых целевая функция и левые части ограниче-

ний представляют собой обобщенные многочлены, а переменные должны быть положительными.

По виду искомого экстремума поисковые методы оптимизации могут быть классифицированы так, как показано на рисунке 2.4.

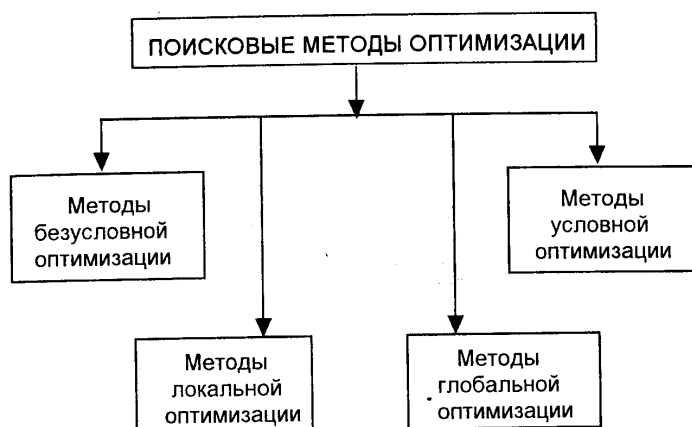


Рисунок 2.4 – Классификация методов по виду искомого экстремума

Методы *безусловной* оптимизации предназначены для нахождения минимума или максимума функций, на переменные которых не накладываются никакие ограничения.

Методы *условной* оптимизации предназначены для решения оптимизационных задач с ограничениями, накладываемыми либо на сами переменные, либо на их связи.

Методы *локальной* оптимизации предназначены для нахождения локальных минимальных или максимальных значений целевой функции (практически все методы оптимизации относятся к локальным).

Методы *глобальной* оптимизации – это те методы и практические приемы, которые позволяют отыскивать глобальный оптимум (наилучшую среди локальных оптимальных точек).

При решении оптимизационных задач большое значение имеют методы безусловной оптимизации, поскольку часто задачи условной оптимизации (задачи с ограничениями) сводятся различными способами к задачам безусловной оптимизации.

Локальные методы безусловной оптимизации делятся на методы *нулевого, первого и второго* порядка (по порядку используемой производной).

Методы нулевого порядка в качестве исходной информации используют лишь значения целевой функции (нулевая производная). Реализация методов первого порядка требует знания первых производных функций. При использовании методов второго порядка необходимо вычислять, кроме значений целевой функции и ограничений, еще и значения первых и вторых производных всех функций, входящих в модель.

Эффективность методов обычно возрастает с ростом порядка производной (методы второго порядка эффективнее методов первого и нулевого порядка). Однако эти методы требуют большего объема вычислений и накладывают дополнительные математические ограничения на входящие в модель функции. В частности, для обеспечения сходимости в ряде задач необходимо иметь возможность вычислять производные функций с очень высокой точностью, что не всегда можно реализовать на практике.

Оценка методов оптимизации:

1. По *скорости сходимости* – показывает, насколько быстро убывает значение целевой функции за единицу времени;
2. По количеству итераций, необходимому для нахождения экстремума;
3. По точности, достигаемой при заданном количестве итераций;
4. По объему вычислений, проводимых на каждой итерации;
5. По трудоемкости программной реализации.

Необходимость использования различных методов связана с тем, что ни один метод или класс методов не являются одинаково эффективными при решении разных типов оптимизационных задач. В частности, возможны случаи, когда происходит переполнение памяти ЭВМ; в других случаях вычисление значений целевой функции требует чрезмерных затрат времени; в некоторых задачах требуется получить решение с очень высокой степенью точности; в ряде задач либо невозможно, либо весьма затруднительно найти аналитические выражения для производных целевой функции. Таким образом, пользователь вынужден приспособлять применяемый метод к конкретным характеристикам решаемой задачи.

## 2.5 НЕОБХОДИМОЕ И ДОСТАТОЧНОЕ УСЛОВИЕ ЭКСТРЕМУМА. ПОНЯТИЕ ВЫПУКЛОСТИ И ВОГНУТОСТИ

Чтобы точка  $X^*$  была точкой безусловного локального экстремума функции  $f(x)$ , необходимо и достаточно, чтобы все её частные производные по элементам  $x_i$  были равны нулю (выполнялось *условие стационарности*):

$$\left. \frac{\partial f(x)}{\partial x_i} \right|_{x=X^*} = 0,$$

где  $i = 1 \dots n$  ( $n$  – число переменных),

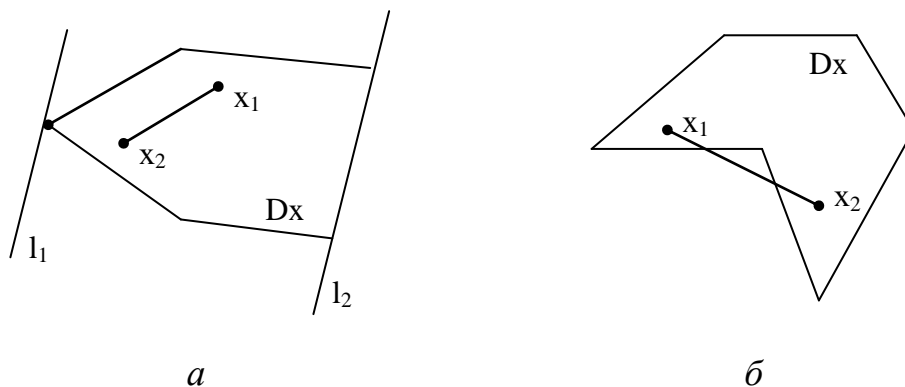
или

$$\text{grad} f(x^*) = \nabla f(x^*) = \left( \frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right) = (\nabla_1 f(x), \nabla_2 f(x), \dots, \nabla_n f(x)) = 0,$$

где  $\nabla$  – обозначение градиента (*набла*).

Характер экстремума функции в стационарной точке  $X^*$  определяется её поведением в этой точке.

Множество  $Dx$  называется *выпуклым*, если для любых двух точек  $x_1$  и  $x_2$  из множества  $Dx$  все точки вида  $\alpha x_1 + (1 - \alpha)x_2$  при  $0 \leq \alpha \leq 1$  также принадлежат множеству  $Dx$  (иначе, если произвольные две его точки можно соединить отрезком, целиком принадлежащим данному множеству). В противном случае множество называется *невыпуклым* (см. рисунок 2.5).



а – выпуклое множество; б – невыпуклое множество

Рисунок 2.5 – Выпуклое и невыпуклое множества

*Опорной* прямой называется прямая, которая имеет с множеством, по крайней мере, одну общую точку, при этом вся область расположена по одну сторону от этой прямой (прямые  $l_1$  и  $l_2$  на рисунке 2.5).

Функция  $f(x)$ , определённая на выпуклом множестве  $Dx$ , называется *выпуклой* (рисунок 2.6), если для любых двух точек  $x_1$  и  $x_2$  из множества  $Dx$  выполняется неравенство

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2) \text{ при } 0 \leq \alpha \leq 1. \quad (2.1)$$

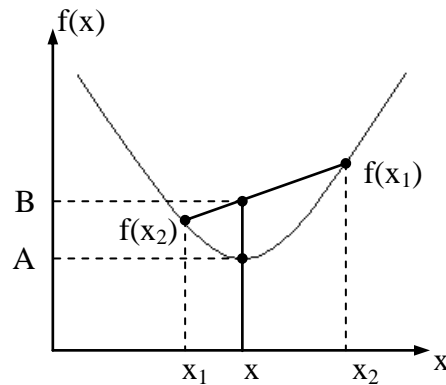


Рисунок 2.6 – Выпуклая функция

Если неравенство (2.1) имеет обратный знак, то функция называется *вогнутой* (рисунок 2.7).

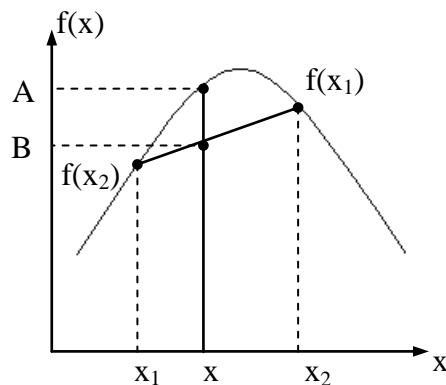


Рисунок 2.7 – Вогнутая функция

На рисунках 2.6 и 2.7 значения  $A$  и  $B$  соответствуют левой и правой частям неравенства (2.1). Т. к. для функции, представленной на рисунке 2.6, неравенство (2.1) выполняется ( $A < B$ ), то функция считается выпуклой.

Для функции же, приведенной на рисунке 2.7, неравенство (2.1) не выполняется ( $A > B$ ), и функция считается вогнутой.

Если неравенство (2.1) выполняется как строгое (например, для функций, представленных на рисунках 2.6 и 2.7), то функция называется *строго выпуклой* (*строго вогнутой*). В противном случае – *нестрого выпуклой* (*нестрого вогнутой*) (рис.2.8).

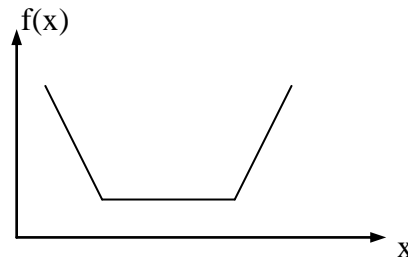


Рисунок 2.8 – Нестрого выпуклая функция

Выпуклость (вогнутость) функции многих переменных в окрестности точки  $X^*$  определяется с помощью определителей, построенных на вторых производных функции  $f(x)$ :

$$f_{11}(X^*); \quad \begin{vmatrix} f_{11}(X^*) & f_{12}(X^*) \\ f_{21}(X^*) & f_{22}(X^*) \end{vmatrix}; \quad \begin{vmatrix} f_{11}(X^*) & f_{12}(X^*) & f_{13}(X^*) \\ f_{21}(X^*) & f_{22}(X^*) & f_{23}(X^*) \\ f_{31}(X^*) & f_{32}(X^*) & f_{33}(X^*) \end{vmatrix};$$

$$H(X^*) = \begin{vmatrix} f_{11}(X^*) & f_{12}(X^*) & \dots & f_{1n}(X^*) \\ f_{21}(X^*) & f_{22}(X^*) & \dots & f_{2n}(X^*) \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1}(X^*) & f_{n2}(X^*) & \dots & f_{nn}(X^*) \end{vmatrix}, \quad (2.2)$$

$$\text{где } f_{ij}(X^*) = \frac{\partial^2 f(X^*)}{\partial x_i \partial x_j}.$$

Матрица  $n$ -го порядка  $H(x)$  (2.2) называется *матрица Гессе* (или *гессиан*).

Если знаки определителей чередуются, то матрица Гессе называется *отрицательно определённой*. В этом случае функция является строго во-

гнутой, и в точке  $X^*$  достигается максимум. Если все определители положительные, то матрица Гессе называется *положительно определённой*, функция  $f(x)$  – строго выпуклая, а точка  $X^*$  является точкой минимума. Если закономерности нет, то функция  $f(x)$  в окрестности точки  $X^*$  называется *седловой* (рисунок 2.9).

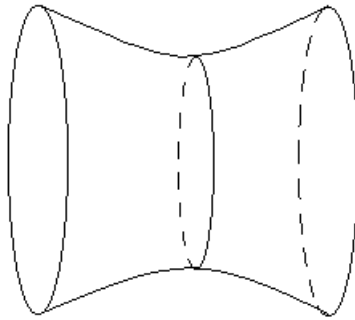
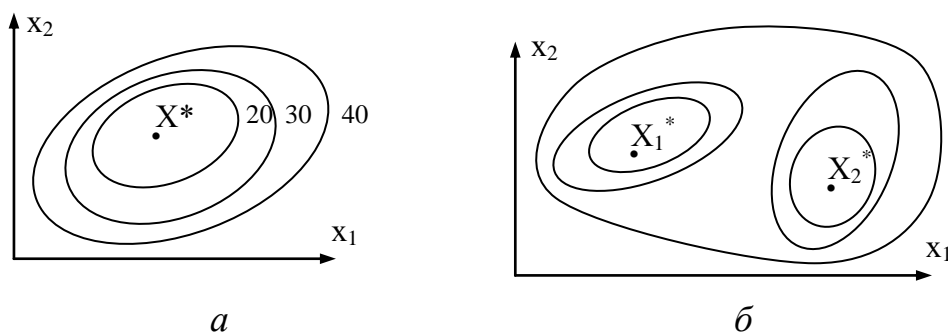


Рисунок 2.9 – Седловая функция

График многомерной функции  $f(x)$  представляется в виде *поверхности отклика*. Поверхность отклика обычно изображают в виде *линий равного выхода* (*линий уровня, изолиний*). Линия уровня имеет уравнение  $f(x) = a$ , где  $a - \text{const}$ , т. е. это геометрическое место точек пространства управляемых параметров  $x_i$ , в которых значения целевой функции равны **a** (рис. 2.10 и 2.11).



а – один экстремум; б – два экстремума

Рисунок 2.10 – Линии уровня одно- и многоэкстремальной функции



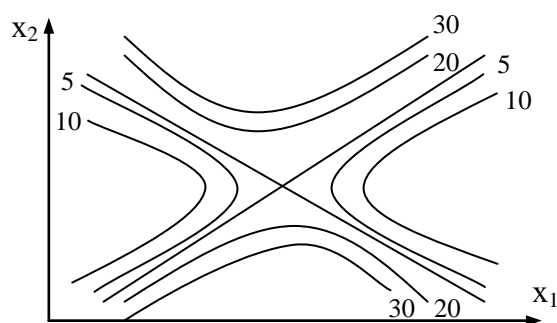


Рисунок 2.11 – Линии уровня седловой функции

### 3 ОДНОМЕРНАЯ ОПТИМИЗАЦИЯ

#### 3.1 ЧИСЛЕННЫЕ МЕТОДЫ ПОИСКА ЭКСТРЕМУМА ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ

Одномерная задача оптимизации в общем случае формулируется следующим образом. Найти наименьшее (или наибольшее) значение целевой функции  $y = f(x)$ , заданной на множестве  $Dx$ , и определить значение проектного параметра  $x \in Dx$ , при котором целевая функция принимает экстремальное значение. Существование решения поставленной задачи вытекает из следующей теоремы.

*Теорема Вейерштрасса.* Всякая функция  $f(x)$ , непрерывная на отрезке  $[a, b]$ , принимает на этом отрезке наименьшее и наибольшее значения, т. е. на отрезке  $[a, b]$  существуют такие точки  $x_1$  и  $x_2$ , что для любого  $x \in [a, b]$  имеют место неравенства

$$f(x_1) \leq f(x) \leq f(x_2).$$

Эта теорема не доказывает единственности решения. Не исключена возможность, когда равные экстремальные значения достигаются сразу в нескольких точках данного отрезка. В частности, такая ситуация имеет место для периодической функции, рассматриваемой на отрезке, содержащем несколько периодов.

Рассмотрим методы оптимизации для разных классов целевых функций. Простейшим из них является случай дифференцируемой функции  $f(x)$

на отрезке  $[a, b]$ , причем функция задана в виде аналитической зависимости  $y = f(x)$ , и может быть найдено явное выражение для ее производной  $f'(x)$ . Нахождение экстремумов таких функций можно проводить методами дифференциального исчисления.

Процесс решения задачи численными методами поиска состоит в последовательном сужении интервала изменения проектного параметра, называемого *интервалом неопределенности*. В начале процесса оптимизации его длина равна  $b - a$ , а к концу она должна стать менее заданного допустимого значения  $\varepsilon$ , т. е. оптимальное значение проектного параметра должно находиться в интервале неопределенности – отрезке  $[x_n, x_{n+1}]$ , причем  $x_n - x_{n+1} < \varepsilon$ .

### 3.2 МЕТОД СКАНИРОВАНИЯ (МЕТОД ПЕРЕБОРА)

Рассмотрим задачу нахождения минимума функции  $y=f(x)$  на интервале  $[a, b]$ .

Наиболее простым способом сужения интервала неопределенности является деление его на некоторое число равных частей с последующим вычислением значений целевой функции в точках разбиения. Пусть  $n$  – число элементарных отрезков,  $h = (b - a)/n$  – шаг разбиения. Вычисляем значения целевой функции  $y_k = f_k(x)$  в узлах  $x_k = a + kh$  ( $k = 0, 1, \dots, n$ ). Сравнивая полученные значения  $f(x_k)$ , найдем среди них наименьшее  $y_{\min} = f_k(x)$ .

$$y_{\min} = \begin{cases} y_k, & y_k < y_{\min} \\ y_{\min}, & y_k \geq y_{\min} \end{cases}.$$

Число  $y_{\min} = y_k$  можно приближенно принять за наименьшее значение целевой функции  $f(x)$  на отрезке  $[a, b]$ . Близость  $y_{\min}$  к минимуму зависит от числа точек, т. е. с увеличением числа точек разбиения погрешность в определении минимума стремится к нулю.

В данном методе основная трудность состоит в выборе  $n$  и оценке погрешности.

### 3.3 МЕТОДЫ ПОСЛЕДОВАТЕЛЬНОГО ПОИСКА НУЛЕВОГО ПОРЯДКА

#### 3.3.1 Стратегия последовательного поиска

Методы *последовательного поиска* являются более экономичным способом уточнения оптимального параметра. Методы последовательного поиска используются для унимодальных целевых функций. Свойство унимодальности позволяет построить процесс сужения интервала неопределенности, при котором любая пара вычислений значения функции уменьшает интервал поиска. При применении методов последовательного поиска к мультимодальным функциям, найденный экстремум может быть не глобальным.

Пусть среди всех значений унимодальной функции  $y=f(x)$ , вычисленных в узлах  $X_k$  ( $k = 0, 1, \dots, n$ ), наименьшим оказалось  $y_i$ . Это означает, что оптимальное значение проектного параметра находится на отрезке  $[x_{i-1}, x_{i+1}]$ , т. е. интервал неопределенности сузился до длины двух шагов (рисунок 3.1).

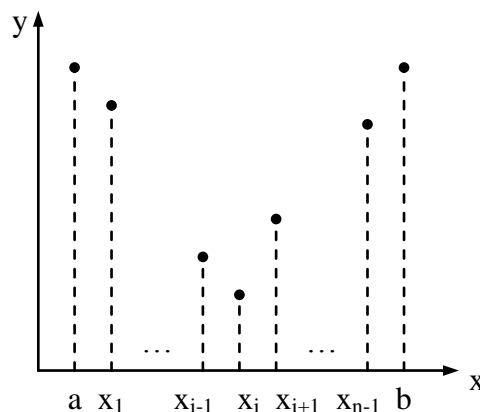


Рисунок 3.1 – Схема последовательного поиска

Если размер интервала недостаточен для удовлетворения заданной погрешности, т. е.  $x_{i-1} - x_{i+1} > \varepsilon$ , то его снова можно уменьшить путем нового разбиения. Получится интервал, равный двум длинам нового шага разбиения, и т. д. Процесс оптимизации продолжается до достижения заданного размера интервала неопределенности.

При построении процесса оптимизации стараются сократить объем вычислений и время поиска. Этого достигают обычно путем сокращения количества вычислений (или измерений при проведении эксперимента) зна-

При разбиении интервала неопределённости двумя точками возможны 3 варианта соотношения значений функции в точках разбиения:

- 
- The left graph shows a function  $f(x)$  on the interval  $[a, b]$ . Points  $x_1$  and  $x_2$  are marked on the x-axis, with  $x_1 < x_2$ . The corresponding function values  $f(x_1)$  and  $f(x_2)$  are marked on the y-axis. The right graph shows the same function  $f(x)$  on the interval  $[a, b]$ . Points  $x_1$  and  $x_2$  are marked on the x-axis, with  $x_1 < x_2$ . The corresponding function values  $f(x_1)$  and  $f(x_2)$  are marked on the y-axis. In this case,  $f(x_1) = f(x_2)$ , and the tangent line at  $x_1$  is horizontal, indicating a local extremum.

6

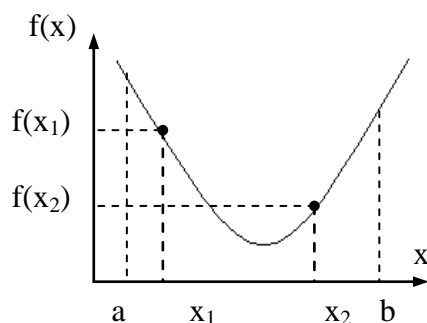


Рисунок 3.2 – Сужение интервала неопределённости методами  
последовательного поиска

20

### 3.3.2 Метод половинного деления (метод дихотомии)

Идея метода *дихотомии* заключается в том, чтобы на каждом шаге вдвое уменьшать интервал поиска.

Выбирается пара точек  $x_1$  и  $x_2$  на расстоянии  $\delta > 0$  от середины текущего интервала неопределенности (рисунок 3.3),  $\delta$  – *параметр метода*.  $\delta$  принимается равным  $\frac{\varepsilon}{2}$ , где  $\varepsilon$  – заданная погрешность.

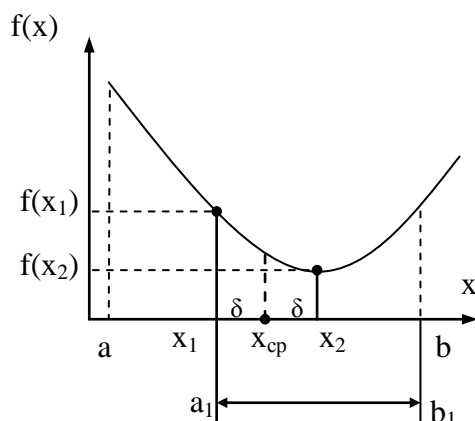


Рисунок 3.3 – Сужение интервала неопределённости с помощью метода дихотомии

Точки  $x_1$  и  $x_2$  делят интервал  $[a; b]$  на три отрезка:  $[a; x_1]$ ,  $[x_1; x_2]$  и  $[x_2; b]$ . Правило исключения интервалов, основанное на свойстве унимодальности функции, позволяет исключить какой-либо из интервалов,  $[a; x_2]$  или  $[x_1; b]$ , в котором определено не может быть точки минимума. А поскольку точки  $x_1$  и  $x_2$  находятся от середины отрезка на расстоянии  $\delta$ , составляющем половину заданной точности вычислений, то какой-либо из указанных отброшенных интервалов практически равен его половине. Таким образом, на каждом шаге данный метод позволяет исключить половину рассматриваемого интервала.

#### Алгоритм метода дихотомии

- 1 Вычисляется срединная точка интервала  $x_{cp} = (a + b)/2$ .
- 2 Определяются координаты двух точек: справа и слева вблизи точки  $x_{cp}$ :

$$x_1 = x_{cp} - \delta; x_2 = x_{cp} + \delta;$$

3 Вычисляются значения функции в этих точках  $f(x_1)$  и  $f(x_2)$ ;

4 Формируется новый интервал неопределенности:

- если  $f(x_1) \leq f(x_2)$ , то  $a_{k+1} = a_k$ ,  $b_{k+1} = x_2$ ;
- если  $f(x_1) > f(x_2)$ , то  $a_{k+1} = x_1$ ,  $b_{k+1} = b_k$ .

5 Если  $(b - a) < \varepsilon$ , то поиск прекращается, если  $(b - a) > \varepsilon$ , то следует перейти на этап 1 и повторить процедуру вычислений на следующей итерации.

После  $k$  вычислений функции экстремум окажется в интервале

$$l = b_k - a_k = \frac{b - a - \delta}{2^{\frac{k}{2}}} + \delta.$$

В качестве оптимальной точки принимается

$$x^* = \frac{a_k + b_k}{2}.$$

Преимуществом метода дихотомии является то, что на каждой итерации алгоритма исключается практически половина интервала поиска. Недостатком является необходимость вычислений на каждом шаге значений функции в двух точках.

### 3.3.3 Метод золотого сечения

*Метод золотого сечения* является одним из наиболее эффективных методов, в которых при ограниченном количестве вычислений  $f(x)$  достигается наилучшая точность. Он состоит в построении последовательности отрезков  $[a_0, b_0]$ ,  $[a_1, b_1]$ , ..., стягивающихся к точке минимума функции  $f(x)$ . На каждом шаге, за исключением первого, вычисление значения функции  $f(x)$  проводится лишь один раз в определенной точке. Эта точка, называемая золотым сечением, выбирается специальным образом.

*Золотым сечением* отрезка называют деление отрезка на две неравные части так, чтобы отношение длины всего отрезка к длине большей части было равно отношению длины большей части к меньшей (рисунок 3.4, а). Т. е. выполняется соотношение

$$\frac{b-a}{b-x} = \frac{b-x}{x-a} \approx 1,618. \quad (3.1)$$



*Рисунок 3.4 – Золотые сечения отрезков*

Всего на отрезке можно отложить 2 золотых сечения (рисунок 3.4, б).

Преобразовав выражение (3.1), получим:

$$x_1 = a + 0,382(b - a) = 0,618a + 0,382b;$$

$$x_2 = a + 0,618(b - a) = 0,382a + 0,618b.$$

Точки  $x_1$  и  $x_2$  располагаются симметрично относительно середины отрезка. Точка  $x_1$  дает золотое сечение отрезка  $[a, x_2]$ , а точка  $x_2$  дает золотое сечение отрезка  $[x_1, b]$ .

### ***Алгоритм метода золотого сечения***

- 1 Проводятся два первых вычисления точек:

$$x_1 = 0,618a + 0,382b;$$

$$x_2 = 0,382a + 0,618b.$$

- 2 Вычисляются значения функции в этих точках  $f(x_1)$  и  $f(x_2)$ .

- 3 Формируется новый интервал неопределенности:

– если  $f(x_1) \leq f(x_2)$ , то  $a_{k+1} = a_k$ ,  $b_{k+1} = x_2$ ;

– если  $f(x_1) > f(x_2)$ , то  $a_{k+1} = x_1$ ,  $b_{k+1} = b_k$ .

- 4 Так как оставшаяся точка дает золотое сечение нового отрезка, то на каждой итерации выбор новой точки  $x^H$  и вычисление значения функции производится 1 раз:

– если  $f(x_1) \leq f(x_2)$ , то  $x_1^H = 0,618a_k + 0,382b_k$ ;  $x_2^H = x_1$ ;

– если  $f(x_1) > f(x_2)$ , то  $x_1^H = x_2$ ;  $x_2^H = 0,382a_k + 0,618b_k$ .

- 5 Вычисление значения функции  $f(x_1^H)$  или  $f(x_2^H)$ .

6 Если  $(b - a) < \varepsilon$ , то поиск прекращается, иначе осуществляется переход на этап 3.

После  $k$  вычислений функции экстремум окажется в интервале

$$l = b_k - a_k = 0,618^k(b - a).$$

В качестве оптимальной точки принимается

$$x^* = \frac{a_k + b_k}{2}.$$

Метод золотого сечения является более эффективным по сравнению с методом дихотомии (и других алгоритмов, использующих процедуру деления интервала на отрезки), поскольку он требует наименьшего числа вычислений значения функции для достижения одной и той же точности.

В то же время уже через небольшое число итераций быстро возрастает погрешность вычисления золотых сечений интервала неопределённости, связанная с ошибками округления. Поэтому через 5 – 10 итераций необходимо заново вычислить 2 золотых сечения, т. е. обнулить метод.

### **3.4 МЕТОДЫ ПОИСКА С ИСПОЛЬЗОВАНИЕМ ПОЛИНОМИАЛЬНОЙ АППРОКСИМАЦИИ**

Основная идея методов связана с возможностью аппроксимации гладкой функции полиномом и последующего использования аппроксимирующего полинома для оценки координаты точки оптимума. Необходимыми условиями эффективной реализации такого подхода является унимодальность и непрерывность исследуемой функции. В соответствии с теоремой Вейерштрасса об аппроксимации, если функция непрерывна на некотором интервале, то ее с любой степенью точности можно аппроксимировать полиномом достаточно высокого порядка.

Следовательно, если функция унимодальна и найден полином, который достаточно точно ее аппроксимирует, то координату точки оптимума функции можно оценить путем вычисления координаты точки оптимума полинома. В соответствии с теоремой Вейерштрасса, качество оценок координаты точки оптимума, получаемых с помощью аппроксимирующего полинома, можно повысить двумя способами: использованием полинома бо-



лее высокого порядка и уменьшением интервала аппроксимации. Вторым способ лучше, поскольку построение аппроксимирующего полинома выше третьего порядка сложнее, чем уменьшение интервала в условиях, когда выполняется предположение об унимодальности функции.

### *Метод оценки с использованием квадратичной аппроксимации*

Метод основан на том, что функция, которая принимает минимальное значение во внутренней точке интервала, должна быть, по крайней мере, квадратичной. При реализации метода предполагается, что на ограниченном интервале можно аппроксимировать функцию квадратичным полиномом, а потом использовать построенную аппроксимационную схему для оценки координаты точки истинного минимума функции.

Если задана последовательность точек  $x_1, x_2, x_3$  и известны соответствующие этим точкам значения функции  $f(x_1), f(x_2), f(x_3)$ , то можно определить постоянные величины  $a_0, a_1, a_2$  таким образом, что значения квадратичной функции  $q(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$  совпадут со значениями  $f(x)$  в трех указанных точках. Перейдем к вычислению  $q(x)$  в каждой из трех заданных точек.

1) При  $x = x_1$   $f(x_1) = q(x_1) = a_0$ , т. е.  $a_0 = f(x_1)$ ;

2) При  $x = x_2$   $f(x_2) = q(x_2) = f(x_1) + a_1(x_2 - x_1)$ .  $a_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$ ;

3) При  $x = x_3$   $f(x_3) = q(x_3) = f(x_1) + \frac{(f(x_2) - f(x_1))(x_3 - x_1)}{x_2 - x_1} + a_2(x_3 - x_1)(x_3 - x_2)$ .

$$a_2 = \frac{1}{x_3 - x_2} \left( \frac{f(x_3) - f(x_1)}{x_3 - x_1} - \frac{f(x_2) - f(x_1)}{x_2 - x_1} \right)$$

Для стационарной точки функции  $q(x)$   $\frac{dq(x)}{dx} = a_1 + 2a_2x - a_2x_1 - a_2x_2 = 0$ .

$$\text{Откуда } \bar{x} = \frac{x_2 + x_1}{2} - \frac{a_1}{2a_2}.$$

Величина  $\bar{x}$  является оценкой координаты точки истинного оптимума  $X^*$ .

### 3.5 МЕТОДЫ ПОИСКА С ИСПОЛЬЗОВАНИЕМ ПРОИЗВОДНЫХ

Методы последовательного поиска и точечной оценки основываются на предположениях об унимодальности и в ряде случаев о непрерывности исследуемой целевой функции. Если в дополнение к условию непрерывности ввести требование дифференцируемости функции, то эффективность поисковых процедур можно существенно повысить.

Однако если функция  $f(x)$  имеет сложную форму или содержит члены, которые включают  $x$  в третьей и более высоких степенях, то непосредственное получение аналитического решения уравнения  $f'(x) = 0$  может оказаться сложным. В таких случаях используются приближенные численные методы его решения.

#### 3.5.1 Метод средней точки (метод Больцано)

Структура поиска основана на исключения интервалов на основании исследования знака производной функции независимо от значений, которые эта производная принимает.

На каждой итерации рассматривается лишь одна пробная точка. Если в пробной точке  $z$  выполняется неравенство  $f'(z) < 0$ , то с учетом предположения об унимодальности, точка минимума не может находиться левее точки  $z$  и интервал  $x \leq z$  подлежит исключению. Если  $f'(z) > 0$ , то точка минимума не может находиться правее  $z$  и исключается интервал  $x \geq z$ .

##### Алгоритм метода

Пусть есть ограниченный интервал  $a \leq x \leq b$  и задан параметр сходимости  $\varepsilon$ . Причем  $f'(a) < 0$  и  $f'(b) > 0$ .

Шаг 1. Вычислить  $z = (a + b)/2$  и  $f'(z)$ .

Шаг 2. Если  $|f'(z)| < \varepsilon$ , закончить поиск. В противном случае, если  $f'(z) < 0$ , принять  $a = z$ , если  $f'(z) > 0$ , принять  $b = z$ . Переход к шагу 1.

#### 3.5.2 Метод секущих (метод хорд)

Пусть в процессе поиска стационарной точки функции  $f(x)$  в интервале  $[a, b]$  выявлено, что в точках  $a$  и  $b$  знаки производной разные. В этом случае алгоритм метода секущих позволяет аппроксимировать функцию  $f$

$f'(x)$  «секущей прямой» (прямой линией, которая соединяет две точки) и найти точку, в которой секущая графика  $f'(x)$  пересекает ось абсцисс. Приближение к стационарной точке  $X^*$  определяется по формуле:

$$z = b - \frac{f'(b)(b-a)}{f'(b) - f'(a)}$$

Если  $|f'(z)| < \varepsilon$ , поиск заканчивается. В противном случае необходимо выбрать одну из точек  $a$  или  $b$  таким образом, чтобы знаки производной в этой точке и точке  $z$  были разными, а потом повторить основной шаг алгоритма.

В отличие от метода средней точки метод секущих основан на исследовании не только знака, но и значений производной в пробных точках и потому в ряде случаев позволяет исключить больше половины интервала поиска.

Пример: найти минимум функции  $f(x) = 2x^2 + 16/x$  в диапазоне  $[1;5]$ .

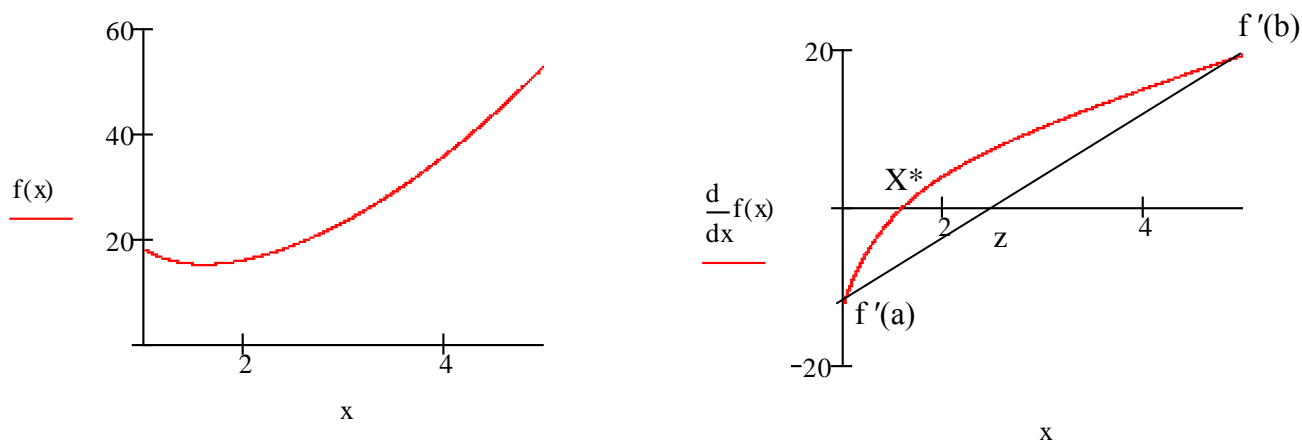


Рисунок 3.4 – График функции и ее производной

Как видно из рисунка (3.4 б), после первой итерации точка  $b$  будет смещена в точку  $z$ , длина дуги уменьшится, и последующая ее аппроксимация прямой будет более точной. Поэтому с каждой итерацией точка  $z$  будет все ближе смещаться к  $X^*$ .

### 3.5.3 Метод Ньютона-Рафсона

Метод Ньютона-Рафсона позволяет улучшить относительно грубую аппроксимацию, чтобы получить корень уравнения  $f'(x) = 0$ .

Спуск производится по формуле

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

где  $k = 0, 1, 2, \dots$  – номер итерации;

$\gamma_k$  – шаг на  $k$ -й итерации (задаваемая константа).

Итерации продолжаются до достижения необходимой точности  $\varepsilon$ , когда  $|f'(x_k)| < \varepsilon$ .

Для квадратичной функции экстремум достигается за 1 итерацию.

Пример: найти экстремум функции  $f(x) = 2x^2 + 8x + 4$ .

$$(f'(x) = 4x + 8. X^* = -2; f(X^*) = -4)$$

По методу Ньютона-Рафсона:

Зададим начальную точку  $X_0$  ( $k = 0$ ). Например,  $X_0 = 10$ .

$$f'(X_0) = 4 \cdot 10 + 8 = 48;$$

$$f''(X_0) = 4.$$

Тогда при  $\gamma_0 = 1$  получим  $X_1 = 10 - 48/4 = -2$ .

## 4 МНОГОМЕРНАЯ БЕЗУСЛОВНАЯ ОПТИМИЗАЦИЯ

### 4.1 ОБЩАЯ СТРАТЕГИЯ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ

Все методы многомерной оптимизации стремятся построить такую последовательность точек  $x_0, x_1, x_2, \dots$ , при которой  $f(x_0) > f(x_1) > f(x_2) > \dots$ . Перед началом поиска выбирается некоторая исходная точка  $x_0$  из области  $D_x$ . Затем необходимо выбрать *направление движения*  $P_k$  и *шаг*  $\gamma_k$  (рис. 4.1).

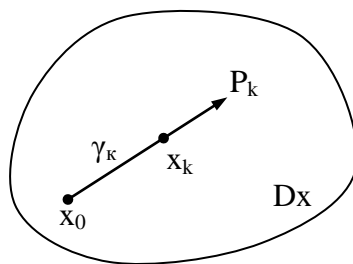


Рисунок 4.1 – Схема многомерной оптимизации

В итоге получаем формулу для нахождения следующей точки:

$$x_{k+1} = x_k + \gamma_k P_k,$$

где  $k = 0, 1, 2, \dots$  – номер итерации.

Общий вид процедуры поиска экстремума функции многих переменных приведен на рисунке 4.2.

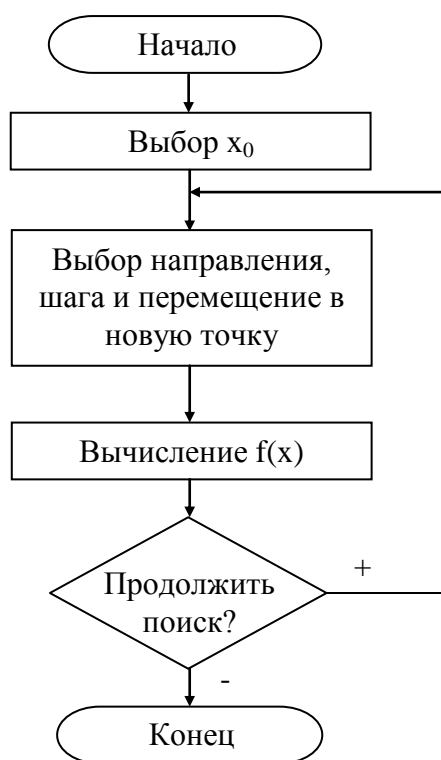


Рисунок 4.2 – Блок-схема алгоритма поиска экстремума функции многих переменных

Последовательность отображающих точек  $x_k$ , соединенных отрезками, называется *траекторией поиска* (рисунок 4.3).

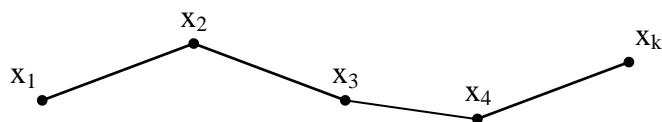


Рисунок 4.3 – Траектория поиска

## 4.2 МЕТОДЫ ОПТИМИЗАЦИИ НУЛЕВОГО ПОРЯДКА

### 4.2.1 Метод покоординатного спуска (метод Гаусса-Зейделя)

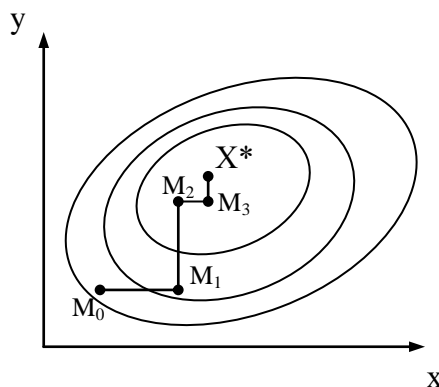
В методах оптимизации нулевого порядка минимизация осуществляется только с помощью вычислений целевой функции.

В качестве начального приближения в  $n$ -мерном пространстве выбирается некоторая точка  $M_0$  с координатами  $x_1^{(0)}$ ,  $x_2^{(0)}$ , ...,  $x_n^{(0)}$ . Фиксируются все координаты функции, кроме первой. Тогда  $f(x_1, x_2^{(0)}, \dots, x_n^{(0)})$  – функция одной переменной  $x_1$ . Решая одномерную задачу оптимизации для этой функции, от точки  $M_0$  переходим к точке  $M_1(x_1^{(1)}, x_2^{(0)}, \dots, x_n^{(0)})$ , в которой функция принимает наименьшее значение по координате  $x_1$  при фиксированных остальных координатах. В этом состоит первый шаг процесса оптимизации, состоящий в спуске по координате  $x_1$ .

Затем фиксируются все координаты, кроме  $x_2$ , и рассматривается функция этой переменной  $f(x_1^{(1)}, x_2, x_3^{(0)}, \dots, x_n^{(0)})$ . Снова решая одномерную задачу оптимизации, находят наименьшее значение при  $x_2 = x_2^{(1)}$ , т. е. в точке  $M_2(x_1^{(1)}, x_2^{(1)}, x_3^{(0)}, \dots, x_n^{(0)})$ . Аналогично проводится спуск по координатам  $x_3, x_4, \dots, x_n$ , а затем процедура снова повторяется от  $x_1$  до  $x_n$  и т. д. В результате этого процесса получается последовательность точек  $M_0, M_1, M_2, \dots$ , в которых значения целевой функции составляют монотонно убывающую последовательность  $f(M_0) > f(M_1) > f(M_2) > \dots$ . Когда ни по одной из осей невозможно перемещение с шагом  $\gamma_k > \gamma_{\min}$ , поиск прекращается, и последняя точка принимается в качестве наименьшего значения целевой функции в рассматриваемой области.

Таким образом, метод покоординатного спуска сводит задачу о нахождении наименьшего значения функции многих переменных к многократному решению одномерных задач оптимизации по каждому проектно-му параметру.

Траектория поиска экстремума методом покоординатного спуска приведена на рисунке 4.4.



*Рисунок 4.4 – Траектория поиска экстремума методом покоординатного спуска*

Сходимость метода зависит от вида функции. Метод не применим в случае наличия изломов в линиях уровня. Это соответствует так называемому оврагу на поверхности. Здесь возможен случай, когда спуск по одной координате приводит на «дно» оврага. Тогда любое движение вдоль другой координаты ведет к возрастанию функции, соответствующему подъему на «берег» оврага. Такое явление называется *эффектом оврага*, а соответствующие функции – *овражными*.

К достоинствам метода покоординатного спуска относится возможность использования простых алгоритмов одномерной оптимизации.

#### **4.2.2 Метод вращающихся координат (метод Розенброка)**

Метод Розенброка направлен на ликвидацию одного из недостатков метода Гаусса-Зейделя – высокую чувствительность эффективности к выбору системы координат. Метод сводится к отысканию "удачной" системы координат путем поворота исходных осей координат в соответствии с изменением скорости убывания целевой функции. Новое направление координатных осей строится таким образом, чтобы одна из координат соответ-

ствовала направлению наиболее быстрого убывания функции, а остальные направления были к ней ортогональны.

#### Алгоритм метода

1. Первый цикл поиска выполняется по методу Гаусса-Зейделя;
2. Выбираются новые координатные оси  $p_1, p_2, \dots, p_n$ . В качестве  $p_1$  выбирают  $x_{k+1} - x_k$ ;
3. Производится ортогонализация остальных осей с помощью процедуры Грама-Шмидта:  $(p_i, p_j) = 0, i \neq j$ ;
4. Проверка условия окончания поиска: если  $|x_{k+1} - x_k| > \varepsilon$ , то переход на этап 1, иначе – на этап 5;
5.  $x^* = x_{k+1}, f^* = f(x^*)$ .

На рис. 4.5. показана траектория поиска экстремума методом вращающихся координат.

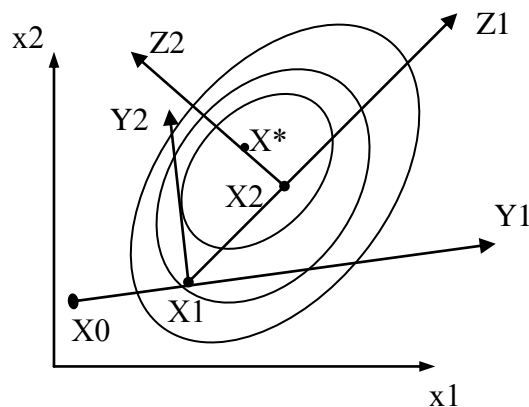


Рисунок 4.5 – Траектория поиска экстремума методом вращающихся координат

Метод Розенброка на каждом цикле обеспечивает такое изменение координатных осей, при котором направление первого спуска очередного цикла стремится к оптимальному. Метод эффективен при минимизации овражных функций, т. к. результирующее направление ориентируется вдоль оврага. Особенно метод эффективен для квадратичных функций, где оптимум может быть найден после одного преобразования осей.



### 4.2.3 Метод параллельных касательных (метод Пауэлла)

Метод использует свойство квадратичной функции, которое заключается в том, что любая прямая, проходящая через  $X^*$  пересекает под равными углами касательные к поверхностям равного уровня в точке пересечения.

Алгоритм метода:

1. Из точки  $X_0$  выполняется одномерный поиск вдоль произвольного направления, которое приводит в точку  $X_1$ .
2. Затем из точки  $X_2$  не лежащей на прямой  $X_0X_1$  опять выполняется одномерный поиск вдоль прямой параллельной  $X_0X_1$ . Получаем точку  $X_3$ .
3. Направление  $X_1X_3$  определяет в результате одномерного поиска точку  $X^*$ .

На рис. 4.6. показана траектория поиска экстремума методом параллельных касательных.

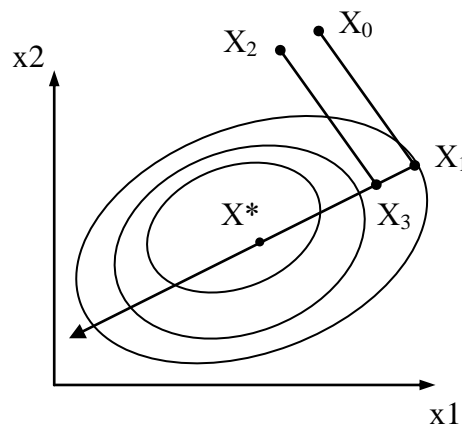


Рисунок 4.6 – Траектория поиска экстремума методом параллельных касательных

Для квадратичной функции минимум находится за  $n$  итераций. Для неквадратичных – это итерационная процедура, сходящаяся к решению тем быстрее, чем ближе минимизируемая функция к квадратичной.

Метод применяется для функций невысокой размерности, близких к квадратичным.

## 4.3 МЕТОДЫ ОПТИМИЗАЦИИ ПЕРВОГО ПОРЯДКА (ГРАДИЕНТНЫЕ МЕТОДЫ)

### 4.3.1 Принцип градиентного спуска

Иногда при использовании методов нулевого порядка для получения решения требуется чрезвычайно большое количество вычислений значений функции. Это обстоятельство, наряду со стремлением реализовать возможности нахождения стационарных точек (то есть точек, удовлетворяющих необходимому условию первого порядка), приводит к необходимости рассмотрения методов, основанных на использовании градиента функции. Указанные методы носят итеративный характер, так как компоненты градиента являются нелинейными функциями оптимизируемых параметров.

В природе часто наблюдаются явления, сходные с решением задачи на нахождение минимума. К ним относится, в частности, стекание воды с берега котлована на дно. Если берега котлована «унимодалыны», т. е. они гладкие и не содержат локальных углублений или выступов, то вода устремится вниз в направлении наибольшей крутизны берега в каждой точке. То есть направление наискорейшего спуска соответствует направлению наибольшего убывания функции.

Известно, что градиент функции  $\nabla f(x)$  ортогонален к поверхности отклика  $f(x)$  в точке  $X$  и его направление совпадает с направлением наибоыстрейшего возрастания функции. Тогда *антиградиент* ( $-\nabla f(x)$ ) направлен в сторону наибоыстрейшего убывания функции  $f(x)$  (рисунок 4.7).

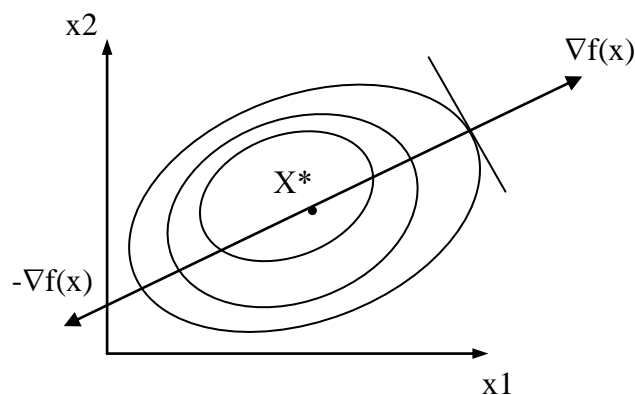


Рисунок 4.7 – Градиент и антиградиент функции  $f(x)$

Методы, основанные на выборе пути оптимизации с помощью градиента, называются *градиентными*.

Градиентные методы применяются для целевых функций, у которых первые производные по всем переменным существуют и непрерывны. Предполагается, что компоненты градиента могут быть записаны в аналитическом виде или с достаточно высокой точностью вычислены при помощи численных методов.

Выбрав в качестве направления  $P_k = -\nabla f(x_k)$ , получаем градиентный метод в векторной форме:

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k),$$

где  $\gamma_k$  – шаг на  $k$ -й итерации ( $\gamma_k > 0$ );

$k = 0, 1, 2, \dots$  – номер итерации.

В координатной форме

$$x_{i\ k+1} = x_{i\ k} - \gamma_k \nabla_i f(x_k),$$

где  $i = 1, 2, \dots, n$  – номер переменной.

Часто в качестве вектора  $P_k$  в градиентных методах выбирают вектор единичной длины

$$P_k = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|},$$

где  $\|\nabla f(x_k)\| = \sqrt{\sum_{i=1}^n \nabla_i^2 f(x_k)}$  – длина вектора.

Тогда получаем следующую форму метода:

$$x_{k+1} = x_k - \gamma_k \frac{\nabla f(x_k)}{\sqrt{\sum_{i=1}^n \nabla_i^2 f(x_k)}}.$$

### 4.3.2 Численное дифференцирование функций

В градиентных методах на каждом шаге оптимизации требуется вычислять градиент целевой функции. При этом бывает трудно получить в яв-

ном виде формулы для частных производных. В этом случае производные вычисляются с помощью разностных методов:

### 1. Алгоритм с центральной пробой

$$\frac{\partial f}{\partial x_i} \approx \frac{1}{\Delta x_i} [f(x_1, x_2, \dots, x_i + \Delta x_i, \dots, x_n) - f(x_1, x_2, \dots, x_i, \dots, x_n)],$$

где  $i = 1, 2, \dots, n$  – номер переменной;

$\Delta x_i = 10^{-4} \dots 10^{-6}$  – шаг аппроксимации (достаточно малая положительная величина).

### 2. Алгоритм с парной пробой

$$\frac{\partial f}{\partial x_i} \approx \frac{1}{2\Delta x_i} [f(x_1, x_2, \dots, x_i + \Delta x_i, \dots, x_n) - f(x_1, x_2, \dots, x_i - \Delta x_i, \dots, x_n)]$$

Первый алгоритм требует меньших вычислительных затрат, но его точность ниже. При этом погрешность во втором алгоритме не зависит от величины  $\Delta x_i$ .

Вторая производная по координате  $x_i$  вычисляется с помощью следующей формулы:

$$\frac{\partial^2 f(x)}{\partial x_i^2} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h_i^2},$$

где  $h_i$  – шаг аппроксимации;

$$f_{i,j} = f(x_1, x_2, \dots, x_n);$$

$$f_{i+1,j} = f(x_1, x_2, \dots, x_{i-1}, x_i + h_i, x_{i+1}, \dots, x_n);$$

$$f_{i-1,j} = f(x_1, x_2, \dots, x_{i-1}, x_i - h_i, x_{i+1}, \dots, x_n).$$

Смешанная производная по координатам  $x_i$  и  $x_j$  вычисляется с помощью следующей формулы:

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4h_i h_j},$$

где  $f_{i\pm 1, j\pm 1} = f(x_1, x_2, \dots, x_i \pm h_i, \dots, x_j \pm h_j, \dots, x_n)$ .

Следует отметить, что применение разностных уравнений дает погрешность в вычислении производных.

### 4.3.3 Градиентный метод с переменным шагом

Одним из наиболее простых методов первого порядка является *градиентный метод с переменным шагом*.

#### *Алгоритм метода*

- 1    Задается начальная точка  $x_0$  и начальный шаг  $\gamma$  при  $k = 0$ .
- 2    Вычисляется  $\nabla f(x_k)$ .
- 3    Вычисляются координаты новой точки

$$x_{k+1} = x_k - \gamma_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|};$$

- 4    Если  $f(x_{k+1}) < f(x_k)$ , то осуществляется переход на этап 5, иначе – переход на этап 6.
- 5    Полагаем, что  $k = k + 1$ , переход на этап 2.
- 6    Уменьшаем шаг:  $\gamma = \gamma/2$ .
- 7    Проверка условия окончания поиска: если  $\gamma > \gamma_{\min}$ , то переход на этап 3; иначе – переход на этап 8.
- 8    Оптимальная точка  $x^* = x_k$ ,  $f^* = f(x^*)$ .

Траектория поиска экстремума методом покоординатного спуска приведена на рисунке 4.8.

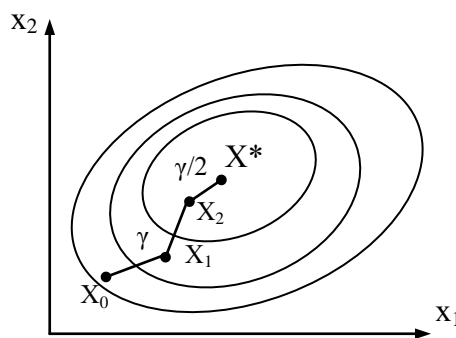


Рисунок 4.8 – Траектория поиска экстремума градиентным методом с переменным шагом

#### 4.3.4 Метод наискорейшего спуска (метод Коши)

При использовании градиентного спуска в задачах оптимизации основной объем вычислений приходится обычно на вычисление градиента целевой функции в каждой точке траектории спуска. Поэтому целесообразно уменьшить количество таких точек без ущерба для самого решения. Это достигается в некоторых методах, являющихся модификациями градиентного спуска. Одним из них является *метод наискорейшего спуска*. Согласно этому методу, после определения в начальной точке направления, противоположного градиенту целевой функции, в этом направлении делают не один шаг, а двигаются до тех пор, пока целевая функция убывает, достигая таким образом минимума в некоторой точке. В этой точке снова определяют направление спуска (с помощью градиента) и ищут новую точку минимума целевой функции и т. д. В этом методе спуск происходит гораздо более крупными шагами, и градиент функции вычисляется в меньшем числе точек.

Метод наискорейшего спуска сводит многомерную задачу оптимизации к последовательности одномерных задач на каждом шаге оптимизации, т. к. величина шага оптимизируется с помощью одного из методов одномерной оптимизации (рисунок 4.9):

$$f(x_k - \gamma_k \nabla f(x_k)) = \min_{\gamma > 0} f(x_k - \gamma \nabla f(x_k))$$

или

$$f(x_k - \gamma_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}) = \min_{\gamma > 0} f(x_k - \gamma \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}).$$

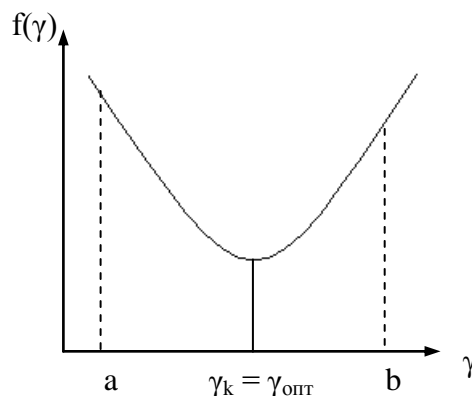


Рисунок 4.9 – Оптимизация шага  $\gamma_k$

### *Алгоритм метода*

- 1 Задание начальной точки  $x_0$ ,  $k = 0$ .
- 2 Вычисление градиента функции в точке  $x_k$ .
- 3 Нахождение одним из методов одномерной оптимизации шага  $\gamma_k$  вдоль антиградиентного направления.
- 4 Проверка условия окончания поиска. Поиск прекращается при выполнении одного из условий:

- 1) достижение минимального размера шага:  $\gamma_k < \gamma_{\min}$ ;
- 2) достижение величиной градиента некоторого минимального значения  $\|\nabla f(x_k)\| < |\nabla|_{\min} \approx 10^{-5}$ ;
- 3) достижение максимального числа итераций:  $k > k_{\max}$ .

При выполнении одного из условий осуществляется переход на этап 7.

В противном случае – переход на этап 5.

- 5 Перемещение в новую точку

$$x_{k+1} = x_k - \gamma_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}.$$

- 6 Полагаем, что  $k = k + 1$ , переход на этап 2.

- 7 Оптимальная точка  $x^* = x_k$ ,  $f^* = f(x^*)$ .

Поиск вдоль прямой с оптимальным шагом обеспечивает более высокую надежность метода наискорейшего спуска по сравнению с градиентным методом с переменным шагом. Однако скорость его сходимости при решении ряда практических задач остается низкой. Это объясняется тем, что изменения переменных непосредственно зависят от величины градиента, которая стремится к нулю в окрестности точки минимума, и отсутствует механизм ускорения движения к точке минимума на последних итерациях. Также метод обладает низкой скоростью сходимости при оптимизации овражных функций.

Одно из главных преимуществ метода наискорейшего спуска связано с его устойчивостью. Метод обладает важным свойством, которое заключается в том, что при достаточно малой длине шага итерации обеспечивается убывание функции.

### 4.3.5 Партан-метод

Одним из простейших антиовражных методов является метод ParTan. Идея метода состоит в том, чтобы запомнить начальную точку, затем выполнить  $k$  шагов оптимизации по методу наискорейшего спуска и сделать шаг оптимизации по направлению из начальной точки в конечную. Для двумерного случая после шага вдоль направления из первой точки в третью траектория спуска приводит в минимум (см. рис. 4.10).

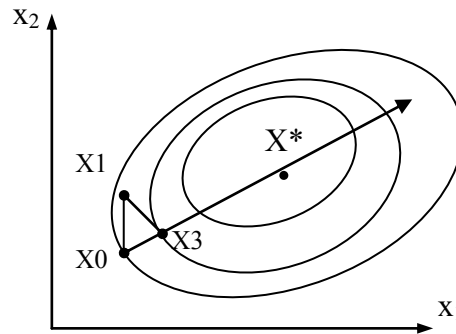


Рисунок 4.10 – Движение в направлении ParTan

В многомерном случае направление ParTan не ведет прямо в точку минимума, но спуск в этом направлении приводит в окрестность минимума меньшего радиуса, чем при еще одном шаге метода наискорейшего спуска. Кроме того, для выполнения третьего шага не требуется вычислять градиент, что экономит время при численной оптимизации.

### 4.3.6 Метод тяжелого шарика

Метод базируется на аналогии с движением «тяжелого» материального шарика по наклонной поверхности. Скорость шарика при движении вниз будет возрастать, и он будет стремиться занять нижнее положение, т.е. точку минимума.

Траектория поиска описывается формулой:

$$x_{k+1} = x_k + \alpha(x_k - x_{k-1}) - \gamma \nabla f(x_k).$$

При  $\alpha = 0$  метод превращается в обыкновенный градиентный. При  $\alpha = 1$  поиск не затухает, следовательно, при  $0 < \alpha < 1$  можно получать различную



эффективность метода, которая будет зависеть и от  $\gamma$ . Вдали от оптимума поиск будет ускоряться, а вблизи возможны колебания около точки  $\min f(x)$ .

К недостаткам метода, помимо колебаний, относится необходимость задания сразу двух неформальных параметров ( $\alpha$  и  $\gamma$ ), определяющих эффективность поиска. К достоинствам метода, помимо ускорения движения вдали от оптимума, относится возможность «проскока» мелких локальных минимумов за счет «инерционности шарика», т.е. можно решать и задачу глобальной оптимизации для функции с одним явно выраженным минимумом и многими «мелкими».

#### 4.3.7 Методы сопряженных градиентов

В методах сопряженных градиентов шаги итерационной процедуры выполняются в сопряженных направлениях.

Два  $n$ -мерных вектора  $\bar{X}$  и  $\bar{Y}$  называются *сопряженными* по отношению к матрице  $H$ , если скалярное произведение  $(\bar{X}, H\bar{Y}) = 0$ .  $H$  – это симметричная, положительно определенная матрица  $n$ -го порядка.

Методы сопряженных градиентов на каждом шаге преобразуют антиградиент  $(-\nabla f(x_k))$  в направление  $P_k$ , которое является  $H$ -сопряженным с ранее найденными направлениями  $P_0, P_1, P_2, \dots, P_{k-1}$ , где  $H$  – матрица Гессе функции  $f(x)$ . Направление  $P_k$  выбирается по следующим формулам:

$$\begin{cases} P_0 = -\nabla f(x_0); \\ P_k = -\nabla f(x_k) + \beta_k P_{k-1}; k \geq 1. \end{cases}$$

Коэффициент  $\beta_k$  выбирается таким, чтобы выполнялось отношение  $(P_k, H P_{k-1}) = 0$ , т. е. чтобы  $P_k$  и  $P_{k-1}$  были  $H$ -сопряженными.

$$\text{Для метода Флетчера-Ривса } \beta_k = \frac{(\nabla f(x_k), \nabla f(x_k))}{(\nabla f(x_{k-1}), \nabla f(x_{k-1}))}.$$

$$\text{Для метода Полака-Рибьера } \beta_k = \frac{(\nabla f(x_{k-1}) - \nabla f(x_k), \nabla f(x_k))}{(\nabla f(x_{k-1}), \nabla f(x_{k-1}))}.$$

Тогда итерационная процедура будет иметь следующий вид:

$$x_{k+1} = x_k + \gamma_k P_k.$$

Величина шага  $\gamma_k$  выбирается оптимальной в направлении  $P_k$  с помощью одного из методов одномерной оптимизации.

Для квадратичной функции

$$\gamma_k = \frac{(\nabla f(x_k), P_k)}{(P_k, HP_k)}.$$

### *Алгоритм метода*

1 Задание начальной точки  $x_0$ ,  $k = 0$ . Вычисление направления  $P_0 = -\nabla f(x_0)$ .

2 Нахождение минимума функции  $f(x)$  одним из методов одномерной оптимизации по  $\gamma$  в направлении  $P_k$  (определение оптимального шага  $\gamma_k$ ).  
Определение новой точки  $x_{k+1} = x_k + \gamma_k P_k$ .

3 Определение  $f(x_{k+1})$  и  $\nabla f(x_{k+1})$ .

4 Проверка условия окончания поиска. Поиск прекращается при выполнении одного из условий:

- 1) достижение минимального размера шага:  $\gamma_k < \gamma_{\min}$ ;
- 2) достижение величиной градиента некоторого минимального значения  $\|\nabla f(x_k)\| < |\nabla|_{\min} \approx 10^{-5}$ ;
- 3) достижение максимального числа итераций:  $k > k_{\max}$ .

При выполнении одного из условий осуществляется переход на этап 7.

В противном случае – переход на этап 5.

5 Определение нового направления

$$P_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} P_k.$$

6 Полагаем, что  $k = k + 1$ . Переход на этап 2.

7 Оптимальная точка  $x^* = x_{k+1}$ ,  $f^* = f(x^*)$ .

Методы сопряженных градиентов обладают более высокой скоростью сходимости для квадратичных функций по сравнению с методом наискорейшего спуска. Минимум квадратичной функции может быть найден за  $n$

шагов (где  $n$  – число переменных). При этом любая гладкая функция в окрестности точки минимума может быть достаточно хорошо аппроксимирована квадратичной функцией. Поэтому метод сопряженных градиентов применяется и для неквадратичных функций.

Метод чувствителен к погрешностям округления. На практике даже для квадратичных функций нет сходимости за  $n$  итераций. Поэтому после  $n + 1$  шага этапы 1...5 должны циклически повторяться с заменой  $x_0 = x_{n+1}$ , т. е. метод обнуляется.

#### 4.3.8 Метод Левенберга-Марквардта

Метод разработан для решения задач нелинейной регрессии наименьших квадратов, где он является более эффективным, чем большинство общих алгоритмов оптимизации. Алгоритм используется для нахождения параметров  $x_i$  ( $i = 1 \dots n$ ), минимизирующих выражение:

$$e(x) = \sum_{j=1}^m e_j(x) = \sum_{j=1}^m (Y_j - \varphi_j(x))^2,$$

где  $e(x)$  – функция ошибки;

$m$  – число компонент вектора ошибки;

$Y_j$  – константы (значения аппроксимируемой функции);

$\varphi(x)$  – аппроксимирующая функция.

Метод основан на идее доверительных областей. Метод предполагает, что исходное отображение является локально линейным. Вблизи точки минимума это предположение выполняется с большой точностью, т. к. здесь поверхность имеет параболоидную форму. В этом случае метод делает скачок сразу в точку минимума.

Вдали от минимума предположение о линейности может быть неправильным. Поэтому метод находит компромисс между линейной моделью и градиентным спуском. Скачок делается только в том случае, если он уменьшает ошибку, а там, где это необходимо для обеспечения продвижения, используется градиентный спуск. Причем в случае успешного скачка баланс смещается в сторону предположения линейности. В случае неудачи алгоритм идет более осторожно вниз по склону. Таким образом, метод все

время меняет схему действия и работает очень быстро.

Итерации метода Левенберга-Марквардта проводятся по формуле

$$x_{k+1} = x_k - (J^T J + \lambda I)^{-1} J^T \bar{e},$$

где  $\bar{e}$  – вектор ошибок на всех наблюдениях;

$\lambda$  – скаляр;

$I$  – единичная матрица;

$J$  – якобиан. В данном случае – матрица частных производных от каждой ошибки по переменным:

$$J = \begin{vmatrix} \frac{\partial e_1(x)}{\partial x_1} & \frac{\partial e_1(x)}{\partial x_2} & \dots & \frac{\partial e_1(x)}{\partial x_n} \\ \frac{\partial e_2(x)}{\partial x_1} & \frac{\partial e_2(x)}{\partial x_2} & \dots & \frac{\partial e_2(x)}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_m(x)}{\partial x_1} & \frac{\partial e_m(x)}{\partial x_2} & \dots & \frac{\partial e_m(x)}{\partial x_n} \end{vmatrix}$$

Первый член в формуле Левенберга-Марквардта соответствует линейной модели, а второй - градиентному спуску. Управляющий параметр  $\lambda$  задает относительную значимость этих двух подходов. Всякий раз, когда удастся уменьшить ошибку, управляющий параметр уменьшается в  $\lambda^-$  (10) раз, усиливая тем самым роль линейных предположений и стремления шагнуть сразу в точку минимума. Когда ошибку уменьшить не удастся, управляющий параметр увеличивается в  $\lambda^+$  (10) раз, придавая тем самым большее значение градиентному спуску и уменьшая величину шага.

#### 4.4 МЕТОДЫ ОПТИМИЗАЦИИ ВТОРОГО ПОРЯДКА

Методы второго порядка применяются для дважды непрерывно дифференцируемых функций, производные которых вычисляются достаточно просто. Точность и скорость сходимости методов второго порядка выше, чем у методов первого порядка, использующих только первую производную (градиентные методы).

#### 4.4.1 Метод Ньютона

Метод Ньютона применяется для оптимизации выпуклых функций.

Новая точка находится из выражения  $x_{k+1} = x_k + \gamma_k P_k$ . Причем  $\gamma_k = 1$ ,  $P_k = -H^{-1}(x_k)\nabla f(x_k)$ , где  $H$  – матрица Гессе. Тогда итерационная процедура примет следующий вид:

$$x_{k+1} = x_k - H^{-1}(x_k)\nabla f(x_k).$$

Если матрица Гессе функции  $f(x)$  положительно определена (т. е. функция выпуклая), то последовательность точек  $\{x_k\}$  стремится к точке минимума  $X^*$ . Если функция также является квадратичной, то метод сходится за одну итерацию независимо от выбора начальной точки  $X_0$  и степени овражности функции. Если функция не является выпуклой, то метод Ньютона гарантирует монотонное убывание функции от итерации к итерации. Сходимость метода в этом случае зависит от удаленности начальной точки  $X_0$  от точки экстремума. Если начальная точка выбирается слишком далеко от экстремума, то метод может расходиться.

#### 4.4.2 Метод Ньютона-Рафсона

Эффективность метода Ньютона можно повысить с помощью регуляции шага. Тогда итерационная процедура будет иметь вид:

$$x_{k+1} = x_k - \gamma_k H^{-1}(x_k)\nabla f(x_k).$$

Шаг  $\gamma_k$  выбирается из условия минимума функции в направлении спуска с помощью одномерной оптимизации.

Алгоритм метода

1. Задание  $X_0$ ,  $k = 0$ , вычисление  $\nabla f(x_k)$ ;
2. Вычисление  $H(x_k)$  и  $H^{-1}(x_k)$ ;
3. Вычисление направления  $P_k$ ;
4. Нахождение одним из методов одномерной оптимизации шага  $\gamma_k$  вдоль  $P_k$ ;
5. Вычисление  $x_{k+1}$  и  $\nabla f(x_{k+1})$ ;

6. Проверка условия окончания поиска. Поиск прекращается если

1)  $\gamma_k < \gamma_{\min}$

2)  $\|\nabla f(x_k)\| < \nabla_{\min}$

3) при достижении максимального числа итераций:  $k > k_{\max}$

При выполнении одного из условий осуществляется переход на этап 8.

В противном случае – переход на этап 7;

7. Полагаем  $k = k + 1$ . Переход на этап 2;

8.  $x^* = x_{k+1}$ ,  $f^* = f(x^*)$ .

Сложность методов второго порядка заключается в нахождении матрицы Гессе для функции  $f(x)$ , т. к. элементами матрицы являются вторые частные производные функции  $f(x)$ . При расчетах на ЭВМ для вычисления производных используют разностное исчисление.

После вычисления элементов матрицы Гессе находят обратную матрицу –  $H^{-1}$ . Матрица  $H^{-1}$  называется *обратной* по отношению к матрице  $H$ , если их произведение равно *единичной* матрице:  $HH^{-1} = E$ . Всякая невырожденная матрица  $H$  (т. е. с отличным от нуля определителем  $|H|$ ) имеет обратную матрицу. При этом  $|H^{-1}| = \frac{1}{|H|}$ . Вычисление обратной матрицы Гессе также производится численными методами.

Из-за накопления ошибки вычисления матрицы Гессе и ее обращения на некоторой итерации матрица Гессе может оказаться отрицательно определенной или ее нельзя будет обратить. В этом случае необходимо принять, что  $H(x_k) = E$ , т. е. в этом случае итерация будет выполняться по методу наискорейшего спуска.

Преимущество метода в том, что для получения одинаковой точности требуется на порядок меньше итераций, чем в градиентных методах. Однако это достигается за счет увеличения объема вычислений на каждой итерации.

В задачах большой размерности выигрыш в сокращении числа итераций не оправдывается. Поэтому целесообразно применять комбинацию градиентного метода и метода Ньютона-Рафсона. В начале процесса оптимизации, когда точка  $x_k$  находится далеко от области экстремума, применяют

градиентный метод, а затем, при уменьшении сходимости, переходят на метод Ньютона-Рафсона.

#### 4.4.3 Квазиньютоновские методы

Эти методы по скорости сходимости близки к методам 2-го порядка, но не требуют вычисления матрицы Гессе и ее обращения, то есть строго говоря относятся к методам 1-го порядка. Итерационная процедура задается выражением:

$$x_{k+1} = x_k - \gamma_k A(x_k) \nabla f(x_k),$$

где  $A(x_k)$  – матрица  $n$ -го порядка, которая называется *метрикой*.

Метрика служит для аппроксимации гессиана. Сначала аппроксимация следует линии наискорейшего спуска, а по мере приближения к минимуму используется более точная оценка гессиана. На квадратичных поверхностях это гарантирует сходимость к истинному обратному гессиану за  $n$  шагов. Для неквадратичных функций в пределе матрица  $A$  становится равной обратному гессиану. Методы поиска вдоль направлений, обусловленных этой формулой, также называются *методами с переменной метрикой*, поскольку матрица  $A$  изменяется на каждой итерации по формуле:

$$A(x_{k+1}) = A(x_k) + \Delta A(x_k),$$

где  $\Delta A(x_k)$  – матрица поправок.

В качестве  $A(x_0)$  выбирается любая симметричная положительно определенная матрица (обычно единичная матрица  $E$ ).

Матрица поправок рассчитывается по различным методам.

Пусть  $\Delta x_k = x_{k+1} - x_k$ ;  $\Delta g_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ ;  $A_k = A(x_k)$ .

##### Метод Давидона-Флетчера-Пауэлла (ДФП)

$$\Delta A(x_k) = \frac{\Delta x_k (\Delta x_k)^T}{(\Delta g_k)^T \Delta g_k} - \frac{(A_k \Delta g_k)(A_k \Delta g_k)^T}{(\Delta g_k)^T A_k \Delta g_k}$$

Метод ДФП – мощная оптимизационная процедура, очень эффективная при оптимизации большинства функций независимо от того, квадратичны они или нет.

### Метод Бroyдена-Флетчера-Гольдфарба-Шенно (BFGS)

$$\Delta A(x_k) = \left[ E - \frac{(\Delta g_k)^T A_k \Delta g_k}{(\Delta x_k)^T \Delta g_k} \right] \frac{\Delta x_k (\Delta x_k)^T}{(\Delta x_k)^T \Delta g_k} - \frac{\Delta x_k (\Delta g_k)^T A_k + A_k \Delta g_k (\Delta x_k)^T}{(\Delta x_k)^T \Delta g_k}$$

К числу главных преимуществ этого метода следует отнести не всегда обязательную необходимость возврата к начальной итерации алгоритма и относительно слабую зависимость от точности вычислений при проведении одномерного поиска.

### Метод Бroyдена

$$\Delta A(x_k) = \frac{(\Delta x_k - A_k \Delta g_k)(\Delta x_k - A_k \Delta g_k)^T}{(\Delta x_k - A_k \Delta g_k)^T \Delta g_k}$$

### Метод Пирсона № 2

$$\Delta A(x_k) = \frac{(\Delta x_k - A_k \Delta g_k)(\Delta x_k)^T}{(\Delta x_k)^T \Delta g_k}$$

### Метод Пирсона № 3

$$\Delta A(x_k) = \frac{(\Delta x_k - A_k \Delta g_k)(A_k \Delta g_k)^T}{(\Delta g_k)^T A_k \Delta g_k}$$

### Проективный метод Ньютона-Рафсона

$$\Delta A(x_k) = \frac{(A_k \Delta g_k)(A_k \Delta g_k)^T}{(\Delta g_k)^T A_k \Delta g_k}$$

### Метод Заутендайка (метод проекций)

$$\Delta A(x_k) = A_k \Delta g_k [\Delta g_k^T A_k \Delta g_k]^{-1} \Delta g_k A_k$$

### Алгоритм квазиньютоновских методов

1. Задание  $X_0$ ,  $k = 0$ . Определение  $\nabla f(x_k)$ ,  $A(x_k) = E$ ;
2. Определение направления спуска:  $P_k = -A(x_k)\nabla f(x_k)$ ;



3. Вычисление одним из методов одномерной оптимизации шага  $\gamma_k$  в направлении  $P_k$ ;
4. Перемещение в новую точку:  $x_{k+1} = x_k + \gamma_k P_k$ . Определение  $\nabla f(x_{k+1})$ ;
5. Вычисление  $\Delta x_k$  и  $\Delta g_k$ ;
6. Проверка условий окончания поиска. При  $\Delta x_k < \varepsilon$  или  $\nabla f(x_{k+1}) < \nabla \min$  или  $k > k_{\max}$  переход на этап 10;
7. Вычисление поправок  $\Delta A(x_k)$ ;
8. Преобразование матрицы  $A(x_k)$  в  $A(x_{k+1})$ ;
9.  $k = k + 1$ , переход на этап 2;
10.  $x^* = x_{k+1}$ ,  $f^* = f(x^*)$

На практике алгоритм может расходиться из-за накопления ошибок вычислений. Поэтому рекомендуется через каждые  $n$  итераций присвоить  $A(x_k) = E$ , т. е. перезапустить алгоритм.

Методы переменной метрики, как и методы 2-го порядка, целесообразно применять в окрестности экстремума.

## 5 МНОГОМЕРНАЯ УСЛОВНАЯ ОПТИМИЗАЦИЯ

### 5.1 МЕТОД МНОЖИТЕЛЕЙ ЛАГРАНЖА

Решение задач математического программирования значительно более трудоемко по сравнению с задачами безусловной оптимизации. Ограничения типа равенств или неравенств требуют их учета на каждом шаге оптимизации. Одним из направлений в методах решения задач математического программирования является сведение их к последовательности задач безусловной минимизации. К этому направлению относится, в частности, *метод множителей Лагранжа*.

Метод множителей Лагранжа применяется для решения нелинейных задач с ограничениями, которые содержат только ограничения в виде равенств. С помощью метода множителей Лагранжа устанавливаются необходимые условия, позволяющие идентифицировать точки оптимума в задачах оптимизации с ограничениями-равенствами. При этом задача с ограниче-

ниями преобразуется в эквивалентную задачу безусловной оптимизации, в которой фигурируют некоторые неизвестные параметры, называемые *множителями Лагранжа*.

Общая задача условной оптимизации с ограничениями-равенствами в виде  $f(x_1, x_2, \dots, x_n) \rightarrow \min$  при ограничениях  $h_k(x_1, x_2, \dots, x_n) = 0$ , где  $k = 1, 2, \dots, K$  сводится к задаче безусловной минимизации с использованием *функции Лагранжа*, которая записывается в виде

$$F(x_1, x_2, \dots, x_n; \lambda_1, \lambda_2, \dots, \lambda_K) = f(x_1, x_2, \dots, x_n) + \sum_{k=1}^K \lambda_k h_k,$$

где  $\lambda_k$  – множители Лагранжа.

Таким образом, задача сводится к отысканию минимума функции  $F(x, \lambda_k)$ . Количество ограничений  $K$  должно быть меньше числа переменных  $n$  (т. е. число связей переменных должно быть меньше их количества). Необходимым условием минимума функции  $F(x, \lambda_k)$  является равенство нулю ее градиента, что приводит к системе уравнений

$$\frac{\partial F(x_j; \lambda_k)}{\partial x_j} = 0,$$

где  $j = 1, 2, \dots, n$  – номер переменной, содержащей  $n$  уравнений с  $(n + K)$  переменными ( $n$  оптимизируемых параметров и  $K$  вспомогательных переменных – множителей Лагранжа).

Дополнив полученную систему уравнений системой ограничений-равенств, получим

$$\begin{cases} \frac{\partial F(x_j; \lambda_k)}{\partial x_j} = 0; \\ h_k(x_j) = 0, \end{cases}$$

где  $k = 1, 2, \dots, K$  – номер ограничения.

Эта система состоит из  $(n + K)$  уравнений и содержит  $(n+K)$  переменных. Таким образом, ее решение позволяет найти оптимальные значения  $n$  оптимизируемых параметров и  $K$  значений множителей Лагранжа в оптимальной точке.

### Пример

Пусть необходимо сконструировать бак заданной емкости  $W$  при минимальном расходе материала. Расход материала определяется площадью боковой поверхности, суммируемой с площадью днища (для бака без крышки). Эскиз бака приведен на рисунке 5.1.

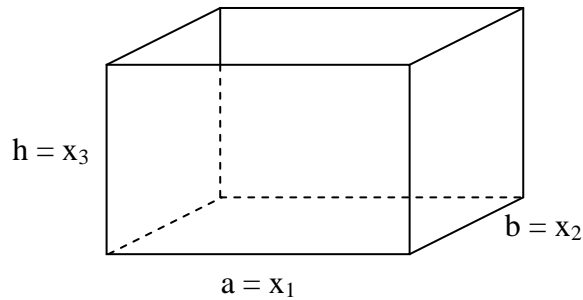


Рисунок 5.1 – Эскиз бака

Формируя целевую функцию, приходим к следующей задаче оптимизации:

$$\begin{cases} S = 2(x_1x_3 + x_2x_3) + x_1x_2 \rightarrow \min; \\ x_1x_2x_3 = W. \end{cases}$$

Второе уравнение представляет собой ограничение в виде равенства. Составим функцию Лагранжа:

$$F(x; \lambda_1) = 2(x_1x_3 + x_2x_3) + x_1x_2 + \lambda_1(x_1x_2x_3 - W).$$

Необходимые условия минимума функции Лагранжа:

$$\begin{cases} \frac{\partial F(x; \lambda_1)}{\partial x_1} = x_2 + 2x_3 + \lambda_1x_2x_3 = 0; \\ \frac{\partial F(x; \lambda_1)}{\partial x_2} = x_1 + 2x_3 + \lambda_1x_1x_3 = 0; \\ \frac{\partial F(x; \lambda_1)}{\partial x_3} = 2x_1 + 2x_2 + \lambda_1x_1x_2 = 0; \\ x_1x_2x_3 = W. \end{cases}$$

Решение этой системы дает:

$$x_1 = x_2 = \sqrt[3]{2W}; \quad x_3 = x_1/2.$$

## 5.2 УСЛОВИЯ КУНА-ТАККЕРА

Множители Лагранжа можно использовать при построении критериев оптимальности для задач оптимизации с ограничениями в виде равенств. Кун и Таккер обобщили этот подход на случай общей задачи нелинейного программирования с ограничениями как в виде равенств, так и в виде неравенств.

Пусть необходимо минимизировать функцию  $f(x_1, x_2, \dots, x_n)$  при ограничениях  $h_k(x_1, x_2, \dots, x_n) = 0$ , где  $k = 1, 2, \dots, K$  и  $g_j(x_1, x_2, \dots, x_n) \geq 0$ , где  $j = 1, 2, \dots, J$ .

Кун и Таккер построили необходимые и достаточные условия оптимальности для задач нелинейного программирования, исходя из предположения о дифференцируемости функций  $f$ ,  $g_j$  и  $h_k$ .

Эти условия оптимальности, известные как *условия Куна-Таккера*, можно сформулировать в виде задачи нахождения решения некоторой системы нелинейных уравнений и неравенств – *задачи Куна-Таккера*.

Найти векторы  $x$ ,  $u$  и  $\lambda$ , удовлетворяющие следующим условиям:

$$\begin{cases} \nabla f(x) - \sum_{j=1}^J u_j \nabla g_j(x) - \sum_{k=1}^K \lambda_k \nabla h_k(x) = 0; \\ g_j(x) \geq 0; \\ h_k = 0; \\ u_j g_j(x) = 0; \\ u_j \geq 0. \end{cases}$$

Решая полученную систему уравнений и неравенств, находим как искомые переменные  $x$ , так и вспомогательные переменные – множители Лагранжа  $u_j$  и  $\lambda_k$ .

## 5.3 МЕТОДЫ ШТРАФНЫХ ФУНКЦИЙ

*Методы штрафных функций* также сводят задачу условной оптимизации к задаче безусловной оптимизации. Использование этих методов поз-

воляет удерживать траекторию поиска внутри допустимой области. В этих методах оптимизируется вспомогательная целевая функция

$$F(x, \tau) = f(x) + \varphi(x, \tau),$$

где  $f(x)$  – исходная целевая функция;

$\varphi(x, \tau)$  – *штрафная* (или *барьерная*) функция;

$\tau$  – *коэффициент штрафа*.

Штрафная функция  $\varphi(x, \tau)$  выбирается таким образом, чтобы при  $\tau \rightarrow 0$  минимум вспомогательной функции совпадал с минимумом исходной.

Штрафная функция должна учитывать ограничения, которые задаются при постановке задачи оптимизации.

Если заданы ограничения в виде  $h_k(x) = 0$ , где  $k = 1, 2, \dots, K$ , и  $g_j(x) \geq 0$ , где  $j = 1, 2, \dots, J$ , то в качестве функции штрафа можно выбрать, например,

$$\varphi(x, \tau) = \frac{1}{\tau} \left\{ \sum_{k=1}^K h_k^2(x) + \sum_{j=1}^J g_j^2(x) [1 - \text{sign}(g_j(x))] \right\},$$

где  $\tau > 0$ ;

$\text{sign}(g_j(x))$  – *сигнум-функция* (или *сигнальная функция*).

$$\text{sign}(g_j(x)) = \begin{cases} 1, & g_j(x) \geq 0; \\ -1, & g_j(x) < 0. \end{cases}$$

При выполнении ограничений штрафная функция обращается в нуль. Если же эти условия нарушены (т.е.  $h_k(x) \neq 0$ ,  $g_j(x) < 0$  и  $\text{sign}(g_j(x)) = -1$ ), то штрафная функция положительна. Она увеличивает целевую функцию  $f(x)$  тем больше, чем больше нарушаются ограничения.

При малых значениях параметра  $\tau$  вне области  $D_x$  функция  $F(x, \tau)$  сильно возрастает. Поэтому ее минимум может быть либо внутри  $D_x$ , либо снаружи вблизи границ области. В первом случае минимумы функций

$F(x, \tau)$  и  $f(x)$  совпадают, поскольку  $\varphi(x, \tau)$  равна нулю. Если минимум функции  $F(x, \tau)$  находится вне  $D_x$ , то минимум целевой функции  $f(x)$  лежит на границе  $D_x$ . Можно при этом построить последовательность  $\tau \rightarrow 0$  такую, что соответствующая последовательность минимумов функции  $F(x, \tau)$  будет стремиться к минимуму функции  $f(x)$ .

Таким образом, задача условной оптимизации для целевой функции  $f(x)$  сводится к последовательности задач безусловной оптимизации для вспомогательной функции  $F(x, \tau)$ , решение которых может быть проведено с помощью методов спуска.

### *Алгоритм метода*

- 1 Задание начальной точки  $x_0$ ;
- 2 Задание коэффициента штрафа ( $\tau = 0,001 \dots 0,01$ );
- 3 Нахождение одним из методов безусловной оптимизации минимума функции  $F(x, \tau)$ ;

4 Проверка условия окончания поиска. Поиск прекращается при выполнении одного из условий:

- 1)  $|x_k - x_{k-1}| < \varepsilon$ ;  $\varepsilon = 10^{-5} \dots 10^{-4}$ ;
- 2)  $|f(x_k) - f(x_{k-1})| < \delta$ .  $\delta = 10^{-3} \dots 10^{-2}$ ;
- 3)  $|F(x_k; \tau_k) - F(x_{k-1}; \tau_{k-1})| < \delta$ .

При выполнении одного из условий осуществляется переход на этап 7.

В противном случае – переход на этап 5.

- 5 Уменьшение коэффициента штрафа  $\tau$  в  $r$  раз ( $r = 5 \dots 10$ );
- 6 Полагаем, что  $k = k + 1$ . Переход на этап 3.
- 7 Оптимальная точка  $x^* = x_{k+1}$ ,  $f^* = f(x^*)$ .

Недостатком метода является сложность вспомогательной функции  $F(x, \tau)$ , которая имеет овражную структуру. Причём с уменьшением  $\tau$  увеличивается степень овражности.

## 5.4 МЕТОДЫ СЛУЧАЙНОГО ПОИСКА

В основе всех методов случайного поиска лежит внесение элементов случайности в процедуру формирования пробных точек, которые используются для определения направления поиска.

### 5.4.1 Метод случайного поиска с пересчетом

Пусть заданы прямые ограничения  $x_{\min i} \leq x_i \leq x_{\max i}$ , где  $i = 1 \dots n$ .

Итерационная процедура минимизации функции  $f(x)$  задается выражением

$$X_{k+1} = X_k + \begin{cases} \gamma_k S \xi; f(X_k + \gamma_k S \xi) < f(X_k) \\ 0; f(X_k + \gamma_k S \xi) \geq f(X_k), \end{cases}$$

где  $\bar{S} = (S_1, S_2, \dots, S_n)$  – вектор масштабных коэффициентов;

$\bar{\xi} = (\xi_1, \xi_2, \dots, \xi_n)$  – случайный вектор с равномерным законом распределения в интервале  $[-1, 1]$ .

В координатной форме процедура минимизации задается выражением

$$x_{i\ k+1} = x_{i\ k} + \begin{cases} \gamma_k S_i \xi_i; f(X_k + \gamma_k S \xi) < f(X_k) \\ 0; f(X_k + \gamma_k S \xi) \geq f(X_k). \end{cases}$$

Значение  $S_i$  определяется по формуле

$$S_i = \frac{x_{\max i} - x_{\min i}}{\max_i (x_{\max i} - x_{\min i})}.$$

Значение  $S_i$  находится в диапазоне  $(0, 1]$ .

Если какой-либо из компонентов  $x_i$  выходит за пределы области поиска, то этому компоненту присваивается граничное значение:

$$x_i = \begin{cases} x_{\max i}, x_i > x_{\max i} \\ x_{\min i}, x_i < x_{\min i}. \end{cases}$$

Благодаря этому условию обеспечивается поиск вдоль границы области допустимых значений при нарушении ограничений. Если при оптимизации  $X_{k+1}$  точка полностью выходит за границы области поиска, т. е.  $X > X_{\max}$  или  $X < X_{\min}$ , то шаг считается неудачным, и значения всех координат точки полагаются равными граничным. Если на какой-либо итерации дела-

ется подряд  $m$  неудачных шагов ( $m = 10 \dots 20$ ), то шаг поиска уменьшают:  $\gamma_{k+1} = \alpha \gamma_k$ , где  $0 \leq \alpha \leq 1$ .

Если в процессе поиска на какой-либо итерации окажется, что для всех  $m$  пробных шагов функция не убывает, то шаг поиска снова уменьшается. Поиск оптимальной точки продолжается, пока  $\gamma_k < \varepsilon = \gamma_{\min}$ .

Схема случайного поиска с пересчетом приведена на рисунке 5.2.

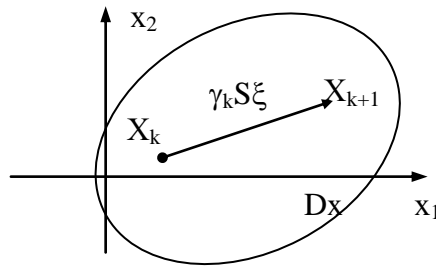


Рисунок 5.2 – Схема случайного поиска с пересчетом

#### 5.4.2 Метод случайного поиска по наилучшей пробе (поиск по статистическому градиенту)

В отличие от предыдущего метода, метод случайного поиска по наилучшей пробе не перемещается в первую же найденную «удачную» точку, а находит несколько точек и выбирает из них наилучшую.

##### *Алгоритм метода*

1 Из точки  $x_k$  выполняется  $m$  случайных пробных шагов, т.е. находят-ся векторы

$$\Delta \bar{x}_j = \gamma_k S \xi_j,$$

где  $j = 1 \dots m$ ,

$\gamma_k$  – шаг.

Затем вычисляются значения функции для всех пробных точек, т. е.  $f_j(x_k + \Delta x_j)$ ;

2 Из всех проб выбирается наилучшая  $\Delta x_n$ , которая приводит к наибольшему уменьшению целевой функции:



$$f(x_k + \Delta x_n) = \min_j f_j(x_k + \Delta x_j).$$

Если  $f(x_k + \Delta x_n) < f(x_k)$ , то переход на этап 3, иначе – переход на этап 5;

3 Производится смещение в новую точку  $x_{k+1} = x_k + \Delta x_n$ ;

4 Полагаем, что  $k = k + 1$ , переход на этап 1;

5 Уменьшение шага  $\gamma_{k+1} = \alpha \gamma_k$ ;

6 Проверка условия окончания поиска. Если  $\gamma_{k+1} > \gamma_{\min}$ , то переход на этап 1, иначе – переход на этап 7.

7 Оптимальная точка  $x^* = x_{k+1}$ ,  $f^* = f(x^*)$ .

Схема случайного поиска по наилучшей пробе приведена на рисунке 5.3.

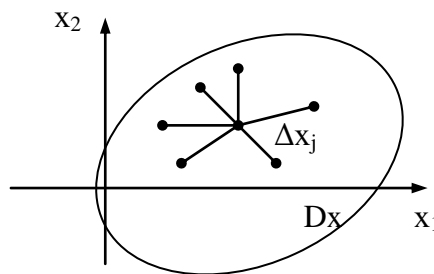


Рисунок 5.3 – Схема случайного поиска по наилучшей пробе

С увеличением числа проб  $m$  направление наилучшей пробы приближается к антиградиентному.

## 6 МЕТОДЫ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ

Практически все методы оптимизации приспособлены для отыскания локального минимума функции. Для надежной оптимизации моделей, которые могут иметь несколько локальных минимумов, следует воспользоваться несколькими методами решения задачи, чтобы найти глобальный минимум.

Наиболее широко используемый метод состоит в проведении ряда оптимизационных расчетов при различных начальных условиях. Иногда этот метод называют *методом с несколькими начальными точками*. В этом методе начальные точки выбираются из определенной решетки (множества

точек) или генерируются случайным образом. В первом случае допустимая область разбивается на непересекающиеся подобласти, и оптимизация выполняется в каждой такой подобласти по отдельности. Во втором случае начальные точки выбираются случайным образом с равномерным распределением поля рассеяния. В обоих случаях в качестве глобального оптимума из всех найденных локальных минимумов принимается локальный минимум с самым минимальным значением целевой функции.

В первом случае разбиение на непересекающиеся подобласти производится произвольно, поэтому нет уверенности в том, что в каждой подобласти имеется только один локальный минимум. Во втором случае нет уверенности, что начальные точки распределены так, что каждая из них находится в окрестности только одного из локальных минимумов. С ростом числа различных стартовых точек повышается вероятность того, что не пропущен ни один локальный минимум. Это, однако, достигается ценой существенного увеличения объемов оптимизационных расчетов.

В практических исследованиях глобальный минимум находится методом случайного выбора стартовых точек, причем сначала следует предположить, какого количества начальных точек достаточно, чтобы наверняка получить глобальное решение. Несмотря на трудность повторных расчетов, этот метод следует использовать в тех случаях, когда предполагается существование нескольких локальных минимумов.

В общем случае речь идет о нахождении как можно более «глубокого» минимума, поскольку если нет априорной информации о количестве локальных минимумов, о структуре функции и т.д., то невозможно доказать, что найденный минимум является глобальным.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

- 1 Бахвалов Н.С. Численные методы / Н.С. Бахвалов. – М.: Наука, 1975. – 275 с.
- 2 Бажин И.И. Информационные системы менеджмента / И.И. Бажин. – М.: ТУ-ВШЭ, 2000. – 688 с.
- 3 Березин И.С. Методы вычислений / И.С.Березин, Н.П.Жидков. – М.: Наука, 1966. – 472 с.
- 4 Васильев Ф.П. Численные методы решения экстремальных задач / Ф.П.Васильев. – М.: Наука. Главная редакция физико-математической литературы, 1980. – 518 с.
- 5 Волков Е.А. Численные методы / Е.А.Волков. – М.: Наука, 1982. – 194 с.
- 6 Дьяченко В.Ф. Основные понятия вычислительной математики / В.Ф.Дьяченко. – М.: Наука, 1977. – 362 с.
- 7 Евтушенко Ю.Г. Методы решения экстремальных задач и их применение в системах оптимизации /Ю.Г. Евтушенко. – М.: Наука, 1982. – 274 с.
- 8 Калиткин Н.Н. Численные методы / Н.Н.Калиткин. – М.: Наука, 1978. – 384 с.
- 9 Конспект лекций по курсу «Математические модели в менеджменте и маркетинге» (для студентов специальности 7.050102 очной и заочной формы обучения)/ Сост. Гитис В.Б. – Краматорск: ДГМА, 2004. – 64 с.
- 10 Крылов В.И. Вычислительные методы / В.И.Крылов, В.В.Бобков, П.И.Монастырный. – М.: Наука, 1977. – 477 с.
- 11 Пшеничный Б.Н. Численные методы в экстремальных задачах / Б.Н.Пшеничный, Ю.М.Данилин. – М.: Наука, 1975. – 420 с.
- 12 Самарский А.А. Введение в численные методы / А.А.Самарский. – М.: Наука, 1982. – 233 с.

13 Турчак Л.И. Основы численных методов: учеб. пособие / Л.И.Турчак. – М.: Наука. Гл. ред. физ.-мат. лит., 1987. – 320 с.

14 Численные методы и вопросы организации вычислений / Под. ред. В.П.Ильина, В.Н. Кублановской. – Л.: Наука, 1984. – 184 с.