

**Федеральное государственное бюджетное
образовательное учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ СТРОИТЕЛЬНЫЙ
УНИВЕРСИТЕТ»**

Кафедра Информатики и прикладной математики

Методические указания к лекционным занятиям

Методы оптимизации

Москва 2024

ОГЛАВЛЕНИЕ

Раздел 1. Вариационное исчисление. Прямые методы вариационного исчисления. Оптимальный расчёт строительных конструкций.

Теоретические предпосылки вариационного исчисления. Необходимые и достаточные условия экстремума функционала простейшего вида. Задачи вариационного исчисления для функционалов различного типа с различными граничными условиями. Вариационные задачи на условный экстремум.

Оптимальный расчет изгибаемой балки и стержня переменного сечения. Прямые методы вариационного исчисления

Раздел 2. Линейное программирование

Постановка задачи линейного программирования. Многогранник решений.

Геометрическая интерпретация. Симплекс-алгоритм решения задач линейного программирования. Взаимно двойственные задачи в ЛП. Экономическая интерпретация. Теоремы двойственности и равновесия. Методы поиска опорных и оптимальных планов в транспортных задачах.

Раздел 3. Методы решения нелинейных задач математического программирования

Точные методы решения нелинейных задач математического программирования

Численные методы поиска экстремума в одномерных, нелинейных задачах математического программирования. Численные методы поиска экстремума в нелинейных задачах математического программирования

Раздел 4. Методы оптимизации в машинном обучении

Методы одномерной минимизации. Градиентные методы и метод Ньютона.

Оптимизация в пространстве большой размерности: общий метод сопряжённых градиентов и неточный (безгессианный) метод Ньютона.

Раздел 1. Вариационное исчисление. Прямые методы вариационного исчисления. Оптимальный расчёт строительных конструкций

Значение методов оптимизации для инженеров. Классификация и примеры задач математического программирования (МП).

Предлагаемое учебное пособие построено на материалах, используемых при чтении одноименного курса лекций и компьютерного практикума, которые проводятся в Московском Государственном Строительном Университете (МГСУ), и учитывают специфику математической подготовки студентов различных специальностей МГСУ. В предлагаемой методичке на основе тщательно отобранного материала создан базовый вариант семестрового курса "Методы оптимизации", включающий необходимые теоретические сведения для точных и численных решений задач оптимизации.

Сущность оптимизационной модели

Для принятия обоснованного решения необходимо иметь и обработать большое количество информации, что, как правило, связано с большими материальными потерями. В настоящее время недостаточно знать путь, ведущий к достижению цели. Необходимо из всех возможных вариантов выбрать наиболее экономичный, который наилучшим образом соответствует поставленной задаче.

Решение оптимизационной задачи заключается в выборе наилучшего (оптимального) варианта из множества возможных по какому-либо признаку – критерию оптимальности.

Признаки оптимизационной модели:

1) Наличие *признака оптимальности* (специального показателя выгоды или критерия оптимальности), который называется *целевой функцией*. Типичные критерии оптимальности: максимум дохода, прибыли, валовой продукт, производительность, эффективность. В таких случаях выгодно, чтобы показатель оптимальности был для выбранного варианта

решения максимальным. Другая группа критериев – это минимум издержек, себестоимости, капиталовложений, трудоемкости, т.е. в этих случаях критерий должен быть минимальным.

2) Наличие *системы ограничений*, т.е. условий, которые описывают множество возможных вариантов (решений), из которых выбирается оптимальный. Множество возможных решений всегда ограничено (ресурсами сырья, наличием рабочей силы, количеством и качеством оборудования и т.п.). Поэтому каждое из рассматриваемых решений должно быть *допустимым*, т.е. удовлетворять имеющимся ограничениям.

Все оптимизационные задачи делятся на два больших класса: 1) *задачи математического программирования (статические)* и 2) *задачи оптимального управления (динамические)*.

Различие между ними в том, что в задаче математического программирования необходимо найти оптимальное число (в общем случае вектор), а в задаче оптимального управления – оптимальную функцию. С формально-математической точки зрения это различие существенное, но в прикладном плане оно зачастую весьма условное. Во-первых, потому, что к задачам математического программирования сводится большинство реальных задач планирования и управления, во-вторых, многие задачи оптимального управления могут быть сведены к задачам математического программирования при условии дискретизации временной характеристики.

Математическая постановка оптимизационной задачи в общем случае состоит в том, чтобы найти вектор $\bar{x} = (x_1, x_2, x_3, \dots, x_n)$ множества значений переменных, при котором достигается наибольшее или наименьшее значение непрерывной скалярной функции $f(\bar{x})$ при условии, что $\bar{x} \in G$:

$$\text{extr}\{f(\bar{x}) | \bar{x} \in G\} \quad (1)$$

Здесь: $f(\bar{x})$ – целевая функция, G – допустимая область, которая представляет собой некоторое подмножество n -мерного евклидова пространства R^n .

Условия, описывающие множество G , называются системой ограничений, которую можно записать в виде системы уравнений $G(x_1, x_2, \dots, x_n) = 0$.

В качестве переменных $x_1, x_2, \dots, x_n$ могут рассматриваться объемы производства различных видов продукции, объемы закупок товаров, производственных площади, количества компонентов в смесях и т.п., т.е. различные технико-экономические факторы и показатели, влияющие на целевую функцию.

Если ограничения в задаче (1) представить в виде системы неравенств $G(x_1, x_2, \dots, x_n) \leq 0$ и добавить ограничения в виде требования неотрицательности переменных $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$, то получим *задачу математического программирования*.

Если критерий оптимальности известен и вариантов немного, то оптимальное решение может быть найдено путем перебора и сравнения всех вариантов.

Однако в большинстве случаев число возможных вариантов велико, полный их перебор выполнить невозможно, поэтому приходится формулировать задачу на языке математики, применять специальные методы оптимизации и реализовывать в специальных пакетах программ.

Сочетание различных элементов модели приводит к различным классам задач оптимизации:

Таблица 1.1

Классы задач оптимизации

Исходные данные	Переменные	Зависимости	Задача
Детерминированные	Непрерывные	Линейные	Линейного программирования
	Целочисленные	Линейные	Целочисленного программирования
	Непрерывные, целочисленные	Нелинейные	Нелинейного программирования
Случайные	Непрерывные	Линейные	Стохастического программирования

Классификация задач математического программирования

Все экономические задачи, решаемые с применением математического программирования, отличаются альтернативностью решения и определенными ограничивающими условиями. Решить такую задачу – значит выбрать из всех допустимо возможных (альтернативных) вариантов лучший, оптимальный. Важность и ценность использования в экономике метода математического программирования состоят в том, что оптимальный вариант выбирается из достаточно значительного количества альтернативных вариантов.

Задачи математического программирования классифицируются в зависимости от вида *целевой функции* и свойств допустимой *области ограничений* G .

К ним относятся задачи: нелинейного программирования, линейного программирования, целочисленного программирования, дробно-линейного программирования, параметрического программирования, сепарабельного программирования, квадратичного программирования, динамического программирования, стохастического программирования, геометрического программирования и другие.

Существуют различные признаки классификации этих задач.

Наиболее важный классификационный признак – ***выпуклость***. По этому признаку все задачи математического программирования разделяются на *выпуклые* и *невыпуклые*.

Задача математического программирования называется *выпуклой*, если она состоит в максимизации (минимизации) выпуклой вверх (вниз) целевой функции на выпуклом множестве, в противном случае задача называется *невыпуклой*.

Если задача оптимизации (1) $\text{extr}\{f(\bar{x})|\bar{x} \in G\}$ является *невыпуклой*, то для ее решения нужно определить все точки локальных максимумов, а затем, сравнив значения целевой функции в них, определить точку глобального максимума. Если точек локальных максимумов много, то решение задачи усложняется.

Однако если задача (1) является *выпуклой*, то ее решение существенно упрощается. В выпуклой задаче локальный экстремум одновременно является и

глобальным. Выпуклая задача может иметь только один строгий экстремум и все сводится к нахождению этого единственного экстремума.

Для решения выпуклых задач разработаны многочисленные численные методы, приспособленные для решения на ЭВМ, связанные с понятием градиента целевой функции и основной идеей о том, что функция наиболее быстро убывает, если двигаться в направлении, противоположном градиенту. К ним относятся метод градиентного спуска, метод сопряженных градиентов и т.д. Но есть и методы, основанные на других идеях: метод штрафных функций, многочисленные варианты метода случайного поиска и т.д.

Другой классификационный признак – *способ задания области ограничений* задачи (1), т.е. множества G . Возможны варианты:

- 1) ограничений нет вообще,
- 2) ограничения заданы системой уравнений,
- 3) ограничения заданы системой неравенств.

Для 1-го случая, если ограничения на переменные не накладываются (т.е. $G = R^n$), а целевая функция является непрерывной дифференцируемой функцией, то задача (1) называется *классической задачей на экстремум (безусловный экстремум)*. В основе ее решения лежит теория дифференциального исчисления, в частности, теория экстремумов и опирающиеся на нее методы.

Для 2-го случая, если $f(\bar{x})$ является непрерывной дифференцируемой функцией, а множество G задано системой уравнений $\varphi_i(\bar{x}) = b_i$, $i = \overline{1, m}$, где $\varphi_i(\bar{x})$ – непрерывно дифференцируемые скалярные функции, b_i – действительные числа, то задача (1) называется *классической задачей на условный (относительный) экстремум*. Обычно эти задачи сводятся к задачам безусловной оптимизации с помощью метода множителей Лагранжа.

В 3-м случае задачи с ограничениями-неравенствами разделяют на два класса – нелинейные и линейные.

Задача вида $f(\bar{x}) \rightarrow \text{extr}$,

$$\begin{cases} \varphi_i(\bar{x}) \leq b_i, & i = \overline{1, m}, \\ x_j \geq 0 & j = \overline{1, n}, \end{cases}$$

где $f(\bar{x})$ и $\varphi_i(\bar{x})$ – непрерывно дифференцируемые нелинейные скалярные функции, b_i – действительные числа, называется *задачей нелинейного программирования*.

Универсального и эффективного метода решения задач нелинейного программирования (даже выпуклых) не существует. Их класс довольно обширен. В нем выделяют отдельные категории задач, имеющих ту или иную специфику, позволяющую строить более эффективные, чем в общем случае, алгоритмы. В качестве примеров задачи нелинейного программирования можно привести квадратичные, сепарабельные, дробно-линейные задачи.

В задаче *квадратичного программирования* целевая функция представляет собой полином второй степени, а система ограничений линейна. В матричных обозначениях данная задача имеет вид

$$\bar{x}^T D \bar{x} + \bar{c}^T \bar{x} \rightarrow \text{extr},$$

$$A \bar{x} \leq \bar{b},$$

$$\bar{x} \geq \bar{0},$$

где D – симметричная матрица размерности $n \times n$, $\bar{c}$ – вектор коэффициентов целевой функции, A – матрица коэффициентов в системе линейных неравенств, $\bar{b}$ – вектор действительных чисел.

Способы решения таких задач во многом определяются видом матрицы D : если D – положительно определённая матрица, то целевая функция будет выпуклой и любой её локальный минимум будет глобальным. Если D – отрицательно определённая матрица, то может быть несколько локальных минимумов, но глобальный минимум, если он существует, достигается обязательно на вершине допустимой области. В общем случае, когда собственные числа матрицы D имеют разные знаки, задача сильно усложняется, так как глобальный минимум может достигаться где угодно – и внутри области, и на её границе.

В задаче *сепарабельного программирования* целевая функция имеет вид $f(\bar{x}) = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n)$.

В задаче дробно-линейного программирования целевая функция представляет собой дробно-линейную функцию, а система ограничений линейна.

Если целевая функция и ограничения (уравнения и неравенства) – линейные, то получим задачу линейного программирования (ЗЛП).

Линейное программирование рассматривают как самостоятельный раздел математического программирования ввиду его особой роли в экономической теории и практике. Для решения ЗЛП разрабатываются специфические алгоритмы, основанные на комбинаторике, графах и т.д. Существует много эффективных методов решения ЗЛП:

- геометрический метод;
- простой симплекс-метод и его модификации:
- симплексные таблицы;
- метод искусственного базиса;
- симплекс-метод с обратной матрицей;
- двойственный симплекс-метод;
- метод последовательного сокращения невязок и другие.

В настоящее время существует много модификаций симплекс-метода, позволяющих существенно сократить время счета, сделать алгоритм нечувствительным к вырожденности опорных планов, повысить размерность решаемых задач, решать так называемые блочные задачи и т.д. Несмотря на обилие этих модификаций, продолжают появляться все новые и новые его варианты.

Кроме того, существуют способы и приемы, позволяющие расширить область применения методов решения ЗЛП. Например, некоторые нелинейные задачи могут быть преобразованы и решены методами, используемыми для ЗЛП.

Важную роль в классе ЗЛП играют задачи целочисленного программирования, в которых на переменные накладывается условие целочисленности. Это связано с физической неделимостью многих объектов расчета. Простое округление до целых чисел здесь не помогает – план может

получиться не оптимальным. Поэтому приходится использовать специальные алгоритмы решения, к наиболее известным из которых относятся алгоритмы Гомори, основанные на так называемой идее отсечения.

Интерес к целочисленному программированию обусловлен еще и тем, что многие нелинейные невыпуклые задачи могут быть сведены к ЗЛП с дополнительным требованием целочисленности. Теория целочисленного программирования используется также для разработки методов решения комбинаторных задач, например, для составления расписаний.

Частным случаем задачи целочисленного программирования относятся *задачи булевского программирования*, где переменные x_j принимают всего два значения – 0 и 1. Наиболее известные из этих задач – это задача о назначениях (какого работника на какую работу поставить), задача выбора маршрута (задача коммивояжера, задача почтальона), задача о максимальном паросочетании и т.д.

Методы и модели линейного программирования используются при решении следующих задач:

- планирование товарооборота;
- построение торговой сети;
- распределение сотрудников по операциям;
- распределение ресурсов;
- организация рациональных перевозок (транспортная задача);
- организация рациональных закупок продуктов (задача о диете);
- оптимальное производство продукции;
- задача о раскрое материала и др.

Многие задачи исследования операций, такие как распределение ресурсов, сетевого планирования, календарного планирования описываются математическими моделями *дискретного программирования*:

Найти $\max f(x_1, \dots, x_n)$ при условиях

$$g_1(x_1, \dots, x_n) \leq 0$$

$$\dots, \\ g_m(x_1, \dots, x_n) \leq 0$$

$$\bar{x} \in D,$$

где D – конечное или счетное множество. Тогда условие $\bar{x} \in D$ – условие дискретности. Чаще всего условие дискретности разделяют по отдельным переменным следующим образом: $x_j \in D_j, j = \overline{1, n}$.

Таким образом, *дискретное программирование* – это раздел математического программирования, в котором на экстремальные задачи налагается условие дискретности переменных при конечной области допустимых значений. Если вводим ограничения x_j – целые числа, то приходим к задачам целочисленного программирования, которые являются частным случаем дискретного программирования. В задачах дискретного программирования область допустимых решений является невыпуклой и несвязной, поэтому отыскание решения в таких задачах сопряжено со значительными трудностями. В частности, невозможно применение стандартных приемов, используемых при замене дискретной задачи ее непрерывным аналогом, состоящих в дальнейшем округлении найденного решения до ближайшего целочисленного.

Задачи геометрического программирования – задачи наиболее плотного расположения некоторых объектов в заданной двумерной или трехмерной области. Такие задачи встречаются в задачах раскроя материала для производства каких-то изделий и т.п. Имеющиеся здесь алгоритмы в основном ориентированы на сокращение перебора вариантов с поиском локальных минимумов.

В задаче *стохастического линейного программирования* коэффициенты c_j целевой функции, коэффициенты a_{ij} в матрице коэффициентов, коэффициенты ограничений b_i – являются случайными величинами. В этом случае сама целевая функция становится случайной величиной, и ограничения типа неравенств могут выполняться лишь с некоторой вероятностью. Приходится менять постановку самих задач с учетом этих эффектов и разрабатывать совершенно новые методы их решения.

Динамическое программирование – это вычислительный метод для решения экстремальных задач определенной структуры, который возник и сформировался в 1950-1953г.г. благодаря работам Р. Беллмана над

динамическими задачами управления запасами. Динамическое программирование представляет собой направленный последовательный перебор вариантов, который обязательно приводит к глобальному максимуму.

Основные свойства задач, к которым возможно применить принцип динамического программирования:

- Задача должна допускать интерпретацию как n -шаговый процесс принятия решений.
- Задача должна быть определена для любого числа шагов и иметь структуру, не зависящую от их числа.
- При рассмотрении k -шаговой задачи должно быть задано некоторое множество параметров, описывающих состояние системы, от которых зависят оптимальные значения переменных. Причем это множество не должно изменяться при увеличении числа шагов.
- Выбор решения (управления) на k -м шаге не должен оказывать влияния на предыдущие решения, кроме необходимого пересчета переменных.

Задача о выборе траектории, задача последовательного принятия решения, задача об использовании рабочей силы, задача управления запасами – классические задачи динамического программирования.

Основные принципы реализации методов оптимизации

Несмотря на разнообразие, все методы решения задач математического программирования имеют некоторые общие черты. Во-первых, практически все они являются численными, то есть представляют собой алгоритмы, ориентированные на компьютерную реализацию. Во-вторых, любой алгоритм является реализацией одного из трех принципов:

- последовательное приближение;
- последовательный анализ вариантов;
- случайный поиск.

1) Большинство методов оптимизации основаны на *принципе последовательного приближения*: в некоторой точке пространства переменных определяется допустимое направление возрастания (или убывания) целевой

функции и делается шаг в этом направлении. Затем анализируется результат, т.е. проверяется, не является ли новая точка искомым решением. Если нет, то вся процедура повторяется вновь. По этому принципу построены, например, градиентные методы и симплексный метод. Как правило, методы этой группы очень эффективны. Недостаток этих методов состоит в том, что они могут найти только локальный экстремум. Если задача невыпуклая (многоэкстремальная), то нет гарантии, что найденное таким способом решение является действительно оптимальным.

2) Принцип *последовательного анализа вариантов* используется в таких методах, как динамическое программирование, метод ветвей и границ. Принцип заключается в построении правил отбраковки подмножеств допустимых вариантов, среди которых не может содержаться оптимального решения. Недостатком использования принципа является то, что данная процедура существенно учитывает специфику задачи, поэтому здесь трудно разработать стандартные алгоритмы для решения широкого круга прикладных задач. Однако эти методы могут быть использованы для решения невыпуклых и комбинаторных задач.

3) Суть *случайного поиска*: формируется некоторый случайный вариант решения (для этого используется компьютерные программы, генерирующие псевдослучайные числа) и вычисляется соответствующее значение целевой функции. Новый вариант сравнивается с лучшим из ранее достигнутых. Если сравнение выполнено в пользу нового варианта, то он запоминается вместо того, который был раньше, и процедура повторяется. Эффективность метода определяется скоростью, с которой компьютер генерирует и оценивает варианты. Сама процедура генерирования вариантов не является случайной, а носит лишь элемент случайности. Гарантии нахождения оптимума нет. Поэтому эти методы применяют лишь тогда, когда нет более надежных и эффективных методов.

Основные понятия вариационного исчисления

Курс вариационного исчисления является продолжением дифференциального и интегрального исчислений и в нем изучаются разнообразные экстремальные задачи.

Известно высказывание великого Эйлера: «В мире нет ничего, в чем не был бы виден смысл какого-либо максимума или минимума».

Например, в аналитической механике все дифференциальные и интегральные принципы основаны на отыскании экстремума тех или иных функций. Так ***интегральный принцип Гамильтона*** формулируется следующим образом:

Из всех движений механической системы, допускаемых идеальными удерживающими голономными связями за один и тот же промежуток времени в потенциальном поле сил, действительным является то, в котором действие по Гамильтону принимает стационарные значения (действие принимает минимальное значение)

$$\delta \int_{t_1}^{t_2} L(q, \dot{q}, t) dt = 0, \quad (1.1)$$

где $L = T - \Pi$;

T – кинетическая энергия; Π – потенциальная энергия;

$q, \dot{q}$ – обобщенные координаты и скорости.

Из этого принципа могут быть получены (и наоборот) уравнения Лагранжа 2-го рода для потенциальных сил

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = 0, \quad j = 1 \dots S. \quad (1.2)$$

Этот вариационный принцип отражает тот факт, что всякое движение в природе происходит по линии наименьшего сопротивления.

Еще раз подчеркнем, что многие важнейшие уравнения механики (динамики, теории упругости и т. д.) получены именно как решение экстремальных задач.

Необходимо также отметить, что классическое вариационное исчисление является базой для решения задач оптимального управления и фундаментальной основой современных методов оптимального управления, таких как принцип максимума Понтрягина, динамическое программирование и т. д.

Примеры решения экстремальных задач известны даже из древней истории. Известна так называемая задача Дидоны. В IX в. до н. э. финикийская царица Дидона со спутниками бежала из г. Тира и высадилась на африканском берегу Средиземного моря. Хитрая Дидона упростила местных жителей отдать ей участок земли, который можно охватить шкурой быка. Простодушный правитель не понял глубины проблемы, а Дидона разрешила шкуру на тончайшие полоски и получившейся веревкой охватила достаточно большую площадь, на которой был заложен Карфаген.

Если длина веревки L , то требуется найти такую форму кривой, которая охватывает наибольшую площадь.

В формальной постановке будем считать, что кривая $y(x)$ гладкая, т. е. непрерывно дифференцируема, и определена на отрезке $[-a, a]$ (рис. 1.1)

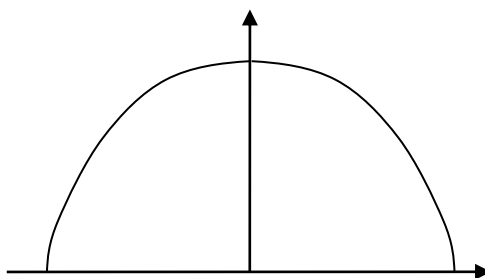


Рис. 1.1. Кривая $y(x)$

Длина кривой, как это известно, из интегрального исчисления, определяется по формуле

$$L = \int_{-a}^a ds$$

$$ds^2 = dx^2 + dy^2.$$

$$ds = \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx = \sqrt{1 + (y')^2} dx.$$

$$L = \int_{-a}^a \sqrt{1 + [y'(x)]^2} dx \quad (1.3)$$

Площадь S определяется интегралом

$$S = \int_{-a}^a y(x) dx \quad (1.4)$$

Экстремальная задача, здесь следующая: найти такую гладкую функцию $y(x)$, которая удовлетворяет условиям $y(-a) = 0$, $y(a) = 0$ при заданном L и обеспечивает интегралу $L = \int_{-a}^a \sqrt{1 + [y'(x)]^2} dx$ максимальное значение.

Подобные задачи ставили и решали еще Аристотель и Архимед, однако считается, что вариационное исчисление как математическая дисциплина оформилась в 1696 г., когда И. Бернулли предложил задачу, решение которой независимо нашли Я. Бернулли, Лопиталь и И. Ньютон.

Задача о брахистохроне

В вертикальной плоскости через две заданные точки O и B , не лежащие на одной вертикали (рис. 1.2), провести кривую (т. е. найти её уравнение), двигаясь по которой материальная точка под действием силы тяжести пройдет от точки O до B за наименьшее время.

В силу теоремы о сохранении энергии, если скорость точки в точке O равна 0, можно записать

$$\frac{mV^2}{2} - mgy = 0 \quad (1.5)$$

Откуда $V = \sqrt{2qy}$.

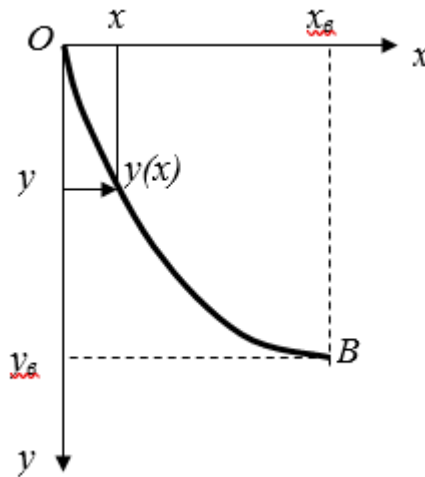


Рис. 1.2. Задача о брахистохроне

Так как

$$V = \frac{ds}{dt} = \frac{\sqrt{1 + (y')^2} dx}{dt}, \quad (1.6)$$

где ds – дифференциал дуги, то

$$\sqrt{1 + (y')^2} dx = dt \cdot \sqrt{2qy}, \quad (1.7)$$

$$dt = \frac{\sqrt{1 + (y')^2}}{\sqrt{2qy}} dx, \quad t = \int_0^a \frac{\sqrt{1 + (y')^2}}{\sqrt{2qy}} dx.$$

Таким образом, нужно найти гладкую функцию $y(x)$, для которой t стремится к $\min$ при краевых условиях $y(0) = 0$, $y(a) = y_a$.

Задача Бернулли о брахистохроне является частным случаем задачи о преломлении света, что послужило основой оптико-механической аналогии.

Задача о минимальной поверхности вращения

Найти $y(x)$, которая обеспечивает наименьшую площадь боковой поверхности тела вращения (рис. 1.3).

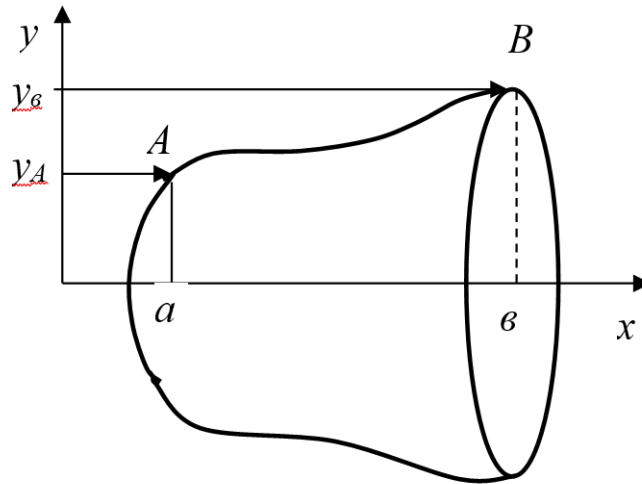


Рис. 1.3. Задача о минимальной поверхности

Как известно, площадь S боковой поверхности тела вращения определяется формулой

$$S = 2\pi \int_a^b y \cdot ds = 2\pi \int_a^b y \cdot \sqrt{1 + (y')^2} dx \rightarrow \min.$$

Задача о геодезических линиях

На поверхности, заданной в прямоугольной системе координат уравнением $u(x, y, z) = 0$, найти кривую, соединяющую точки A и B с наименьшей длиной. На плоскости это прямые линии, на сфере – дуги большого круга (рис. 1.7).

Если поверхность $u(x, y, z)$ гладкая, а искомая кривая определяется уравнениями $y = y(x)$, $z = z(x)$, то длина кривой L равна

$$L = \int_A^B \sqrt{dx^2 + dy^2 + dz^2} = \int_a^b \sqrt{1 + (y')^2 + (z')^2} dx,$$

т. е. пришли снова к вариационной задаче.

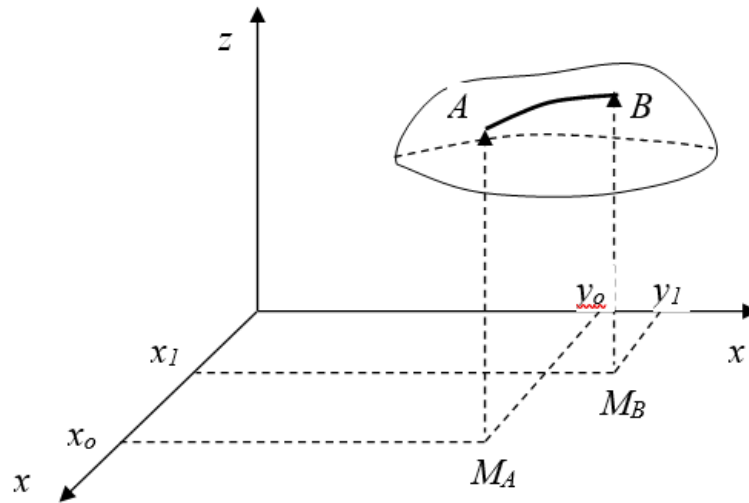


Рис. 1.4. Задача о геодезических

Задачи строительной механики

К вариационной задаче сводится проблема определения упругой линии струны, мембраны, балки оболочки и т. д. Например, форма балки, защемленной по краям, находится из условия равновесия, т. е. min

потенциальной энергии $\Pi = \int_0^l \frac{K \cdot y^2}{2} + P \cdot y \, dx$

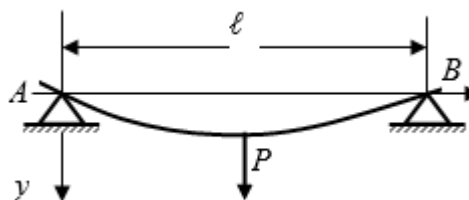


Рис. 1.5. Форма балки

Можно отметить, что форма балки и соответственно решение вариационной задачи зависит от граничных условий, например если балка закреплена (рис. 1.6), то форма изгиба балки будет отлична от показанной на рис. 1.5.

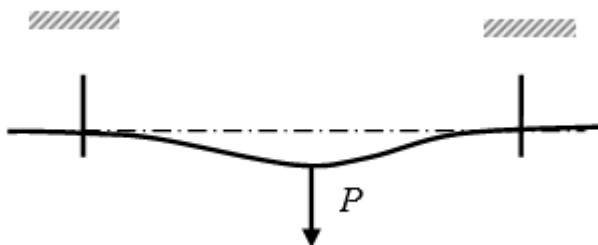


Рис. 1.6. Консоль

Большой класс задач, приводящих к вариационным проблемам, связан с оптимальным управлением. Например, задача поворота вала двигателя на заданный угол без остановки (с остановкой) за минимальное время с ограничением на управление (без этого ограничения задача становится бессмысленной). Часто ставится также вариационная задача поворота вала двигателя на заданный угол за заданное время T при минимальном расходе энергии.

Очень много задач связано с управлением летательных аппаратов, например ракет. Проблема вывода ракеты в заданную точку фазового пространства (координаты и скорости) за минимальное время при ограничении на величину реактивной силы $P \leq P_m$. Или, например, задача перевода летательного аппарата на максимальную дальность или высоту, т. е. при заданных уравнениях объекта, фазовых ограничениях, ограничениях на управление (запас топлива ограничен и известно ограничение на максимальную силу тяги) и заданных краевых условиях. Таким образом, необходимо найти управление, обеспечивающее максимальную дальность или высоту.

К вариационным задачам приводят также проблемы обработки информации, выделение полезного сигнала на фоне помех и т. д.

Примеров постановок вариационных задач можно приводить очень много, но из изложенного выше следует, что изучение курса «Основы вариационного исчисления» должно обеспечить:

- понимание постановки вариационных задач;

- умение ставить вариационную задачу;
- умение находить решение вариационных задач в рамках классического вариационного исчисления.

Основные определения

Одним из базовых понятий вариационного исчисления является понятие функционала.

Переменная величина $J[y(x)]$ называется функционалом, зависящим от функции $y(x)$, если каждой функции $y(x)$ для некоторого класса функций соответствует число J .

Аналогично определяются функционалы, зависящие от нескольких функций. Можно считать, что функционал представляет собой обобщение понятия функции, в котором роль независимой переменной играет другая функция. Еще раз подчеркнём, что значением функционала является число.

Довольно часто придется иметь дело с функционалами вида

$$J = \int_{x_1}^{x_2} f[y(x), y'(x), x] dx.$$

Можно привести достаточно много примеров использования функционалов при решении различных прикладных задач:

- путь, пройденный объектом за время T

$$S = \int_0^T v(t) dt$$

- сила сопротивления воды движению судна является функционалом, зависящим от формы корпуса;
- стоимость прокладки дороги между двумя пунктами будет функционалом от трассы и т. д.

При формулировке понятия функционала использовано выражение «некоторый класс функций», вследствие чего необходимо расшифровать это выражение и рассмотреть функциональные пространства.

При изучении функций n -переменных удобно пользоваться геометрическим языком, рассматривая набор n чисел $(y_1, y_2, \dots, y_n)$ как точку n -мерного пространства. При $n = 3$ имеем обычное трехмерное пространство. При $n = 3$ – многомерное эвклидово пространство.

При изучении функционалов каждую функцию $y(x)$, принадлежащую какому-либо классу, будем рассматривать как точку некоторого пространства.

Пространства, элементами которых являются функции, называются функциональными пространствами. Если для изучения функций можно рассматривать одно n -мерное эвклидово пространство, то не существует «универсального» функционального пространства – они выбираются в зависимости от характера вариационной задачи.

Например, для $J = \int_a^b F(y, y', x) dx$ функционал нужно рассматривать как совокупность всех функций, имеющих непрерывную первую производную, а для $J = \int_a^b F(y, y', y'', x) dx$ необходимо взять класс дважды дифференцируемых функций.

Элементами линейного нормированного пространства могут быть различные объекты: системы чисел, векторы, матрицы, функции и т. д.

Представляют особый интерес следующие пространства.

Пространство C , состоящее из всех непрерывных функций с нормой $\|y\|_c = \max |y(x)|$,

Пространство C_1 функций с нормой

$$\|y\|_{C_1} = \max \{y^{(l)}\}.$$

Гильбертово пространство L_2 функций, суммируемых с квадратом на отрезке $[a, b]$ и с нормой

$$\|y\|_{L_2} = \sqrt{\int_a^b y^2(x) dx}.$$

Если функционал $J(y)$ задан на линейном пространстве L и представляет собой линейную форму, т. е.

$$J[\alpha_1 y_1 + \alpha_2 y_2] = \alpha_1 J(y_1) + \alpha_2 J(y_2),$$

то функционал называется линейным.

При изучении функционалов вводят ряд понятий, аналогичных понятиям для функций: непрерывность, дифференцируемость, экстремум и др.

Функционал $J[y(x)]$, определенный на нормированном пространстве M , называют непрерывным в точке y_0 в M , если для всякого числа ε существует такая δ окрестность точки y_0 , что для любой точки y из этой окрестности выполнено неравенство

$$|J[y] - J[y_0]| < \varepsilon.$$

Разность $y(x) - y_0(x) = \delta y(x)$ называется вариацией функции $y_0(x)$.

Необходимо подчеркнуть отличие понятия вариации от приращения функции в точке.

Приращение функции в точке x_0 – число, равное разности двух значений функций, а вариация – это функция, равная разности двух функций, являющихся аргументами функционала.

Отметим, что для дифференцируемых функций следует различать производную вариации $\delta y' = (\delta y)'$ и вариацию производной

Вариация функционала

Еще раз подчеркнем, что функционалы, заданные с помощью интегралов, называются интегральными функционалами, а подынтегральная функция — интегрантом.

$$\text{Пусть } J[y] = \int_a^b f(x, y, y') dx.$$

Пусть $y(x)$ трижды дифференцируемая функция. Приращение функционала, соответствующее приращению аргумента $\square y(x)$ (полное приращение функционала), можно записать в виде

$$\begin{aligned} \Delta J &= J[y + \delta y] - J[y] = \int_a^b f(x, y + \delta y, y' + \delta y') dx - \int_a^b f(x, y, y') dx = \\ &= \int_a^b [f(x, y + \delta y, y' + \delta y') - f(x, y, y')] dx. \end{aligned}$$

Если первая вариация, т. е. функционал по y линеен, то этот функционал называют слабым дифференциалом (дифференциалом Гато) в точке « y ».

Полное приращение функционала $J[y]$ представим еще и следующим образом:

$$\Delta J[h] = J[y + h] - J[y] = \delta J[h] + \alpha \|h\|,$$

где $\alpha \rightarrow 0$, когда $\|h\| > 0$.

По сути, второе слагаемое представляет собой вторую и следующие вариации, т. е. нелинейные слагаемые высших порядков.

Условия экстремума функционала

Теорема: для того чтобы функционал $J[y]$ при $y = y_0$ достигал экстремума, необходимо, чтобы его дифференциал (если он существует) обращался в нуль при $y = y_0$, т. е. первая вариация тождественно равна нулю.

ВАРИАЦИОННАЯ ЗАДАЧА С ЗАКРЕПЛЕННЫМИ ГРАНИЧНЫМИ ТОЧКАМИ. УРАВНЕНИЕ ЭЙЛЕРА (ПРОСТЕЙШАЯ ВАРИАЦИОННАЯ ЗАДАЧА)

Исследуем на экстремум функционал $J[y(x)] = \int_{x_1}^{x_2} f[y(x), y'(x), x] dx$, где $f(x, y, y')$ непрерывная и трижды дифференцируемая функция своих аргументов.

Искомая функция, для которой этот функционал принимает экстремальное значение, удовлетворяет краевым условиям

$$y(x_1) = y_1; \quad y(x_2) = y_2$$

Эта задача о нахождении экстремума функционала при заданных условиях называется вариационной задачей с закрепленными граничными точками.

Подчеркнем, что целью исследования является также и нахождение функции $y(x)$, доставляющей экстремум функционалу.

Непрерывно дифференцируемые функции $y(x)$, определенные на $[x_1, x_2]$ и удовлетворяющие условиям, называются допустимыми функциями.

Будем искать такую кривую $y = y(x)$ на участке $x_1 < x < x_2$, которая реализует экстремум функционала $J = \int_{x_1}^{x_2} f(x, y, y') dx$

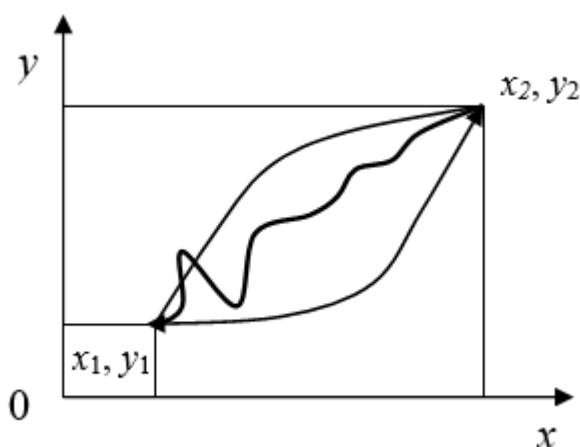


Рис. 1.6. Вариация функции

Примем, что решение найдено и соответствует функции $y(x) = y_0(x)$.

В этом случае уравнение Эйлера примет вид

$$f_y - \frac{d}{dx} f_{y'} = 0$$

Прежде чем переходить к обсуждению уравнения Эйлера рассмотрим в качестве примера классическую задачу о брахистохроне.

Найти линию, соединяющую две точки A и B , лежащие на одной прямой в вертикальной плоскости и обладающие тем свойством, что свободно отпущенная материальная точка под действием только сил тяжести скатится из точки A в точку B за кратчайшее время (течением и сопротивлением воздуха пренебрегаем).

Пусть точка A имеет координаты x_0, y_0 , $B - x_1, y_1$.

Обозначим через V – скорость движения точки вдоль кривой, ds – элемент дуги.

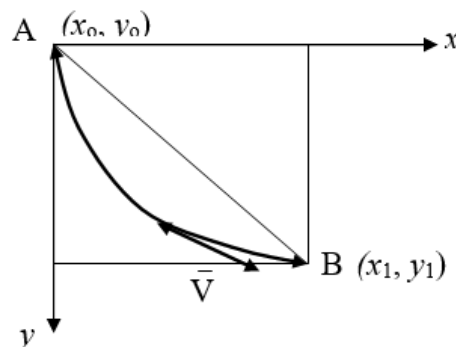


Рис. 1.7. Движение точки

В этом случае $V = \frac{ds}{dt}$, откуда $t = \int_A^B \frac{ds}{V}$.

Если точку A с координатами x_0, y_0 , принять за начало отсчета, то

$$t = \int_0^{x_1} \frac{ds}{V}.$$

Элемент дуги можно представить в виде

$$ds = \sqrt{dx^2 + dy^2} = \sqrt{1 + y'^2} dx.$$

На основании теоремы об изменении кинетической энергии для материальной точки имеем

$$\frac{1}{2}mV^2 - \frac{1}{2}mV_0^2 = mg \cdot y,$$

и так как начальная скорость $V_0 = 0$, то

$$\frac{1}{2}mV^2 = mg \cdot y,$$

откуда для V имеем

$$V = \sqrt{2gy}.$$

Функционал, экстремум которого необходимо отыскать в рамках поставленной задачи, будет

$$T = \frac{1}{\sqrt{2g}} \int_0^{x_1} \frac{\sqrt{1 + y'^2}}{\sqrt{2gy}} dx.$$

Отметим, что полученный интеграл представляет собой действие по Лагранжу в форме Якоби.

Кривая $y = y(x)$ на участке $x = 0$, $x = x_1$, реализующая экстремум функционала $J[y(x)]$ должна удовлетворять уравнению Эйлера

$$f_y - \frac{d}{dx} f_{y'} = 0,$$

Часто уравнения Эйлера, в простейших случаях, полностью решают поставленную задачу поиска экстремума.

Преобразуем уравнение Эйлера, используя правило дифференцирования сложной функции.

Продифференцируем второе слагаемое уравнения Эйлера

$$\frac{d}{dx} f_{y'} = \frac{\partial}{\partial x} \left(f_{y'} \right) \frac{dx}{dx} + \frac{\partial f_{y'}}{\partial y} \frac{dy}{dx} + \frac{\partial f_{y'}}{\partial y'} \frac{dy'}{dx} = f_{xy'} + f_{y'y'} y' + f_{y'y'} y''.$$

Таким образом, уравнение Эйлера есть нелинейное дифференциальное уравнение 2-го порядка.

$$f_{xy'} + f_{y'y'} y' + f_{y'y'} y'' - f_y = 0.$$

Пример. Найти экстремали функционала

$$J[y] = \int_0^2 \left[x(y')^3 - 3y(y')^2 \right] dx,$$

при граничных условиях $y(0) = 4$, $y(2) = 6$;

$$f_y = -3(y')^2, \quad f_{y'} = 3x(y')^2 - 6y \cdot y';$$

$$\frac{d}{dx} f_{y'} = 3(y')^2 - 6(y')^2 + 6xy' \cdot y'' - 6y \cdot y''.$$

Уравнение Эйлера

$$3(y')^2 - 6(y')^2 + 6xy' \cdot y'' + 3(y')^2 = 0;$$

$$y''(xy' - y)^2 = 0 \Rightarrow y'' = 0, \quad C_1 x + C_2 = y; \quad xy' - y = 0; \quad y = Cx.$$

Решение второе входит в семейство первых. C_1 и C_2 находим из граничных условий.

$$C_2 = 4;$$

$$C_1 2 + C_2 = 6.$$

Таким образом, экстремаль $y = x + 4$.

Функционалы, зависящие от производных высших порядков

Исследуем функционал

$$J[y(x)] = \int_{x_0}^{x_1} f \left[x, y(x), y'(x), y^{(n)}(x) \right] dx,$$

где $y(x)$ дифференцируема $n + 2$ раза и граничные условия имеют вид

$$y(x_0) = y_0, \quad \{ y'(x_0) = y'_0, \quad y^{(n)}(x_0) = y_0^{(n)}, \\ y(x_1) = y_1, \quad \{ y'(x_1) = y'_1, \quad y^{(n)}(x_1) = y_1^{(n)}$$

Рассмотрим, как и ранее, однопараметрическое семейство функций

$$y(x, \alpha) = y(x) + \alpha \delta y,$$

при этом $x = x_0$ и $x = x_1$ вариации $\delta y = \delta y \dots \delta y^{(n-1)} = 0$, а при $\alpha = 0$ $y(x, \alpha) = y(x)$, где $y(x)$ – кривая, на которой достигается экстремум.

Если рассматривать значения функционала $J[y(x)]$ только на кривых семейства $y = y(x, \alpha)$, то функционал превратится в функцию параметра α , достигающую экстремума при $\alpha = 0$, следовательно, первая вариация функционала должна быть равна нулю $\delta J = \frac{d}{d\alpha} J[y(x, \alpha)] \Big|_{\alpha=0} = 0$.

Проведем дифференцирование под знаком интеграла

$$\delta J = \left[\frac{d}{d\alpha} \int_{x_0}^{x_1} f(x, y(x, \alpha), y'(x, \alpha), \dots, y^{(n)}(x, \alpha)) dx \right]_{\alpha=0} = \\ = \int_{x_0}^{x_1} \left[f_y \delta y + f_{y'} \delta y' + f_{y''} \delta y'' + \dots + f_{y^{(n)}} \delta y^{(n)} \right] dx.$$

Интегрируем второе слагаемое по частям один раз

$$\int_{x_0}^{x_1} f_{y'} \delta y' dx = f_{y'} \delta y \Big|_{x_0}^{x_1} - \int_{x_0}^{x_1} \frac{d}{dx} f_{y'} \delta y dx,$$

третье слагаемое – два раза

$$\int_{x_0}^{x_1} f_{y''} \delta y'' dx = \left[f_{y''} \delta y' \right]_{x_0}^{x_1} - \left[\frac{d}{dx} f_{y''} \delta y \right]_{x_0}^{x_1} + \int_{x_0}^{x_1} \frac{d^2}{dx^2} f_{y''} \delta y dx,$$

и последнее слагаемое n – раз

$$\int_{x_0}^{x_1} f_{y^{(n)}} \delta y^{(n)} dx = \left[f_{y^{(n)}} \delta y^{(n-1)} \right]_{x_0}^{x_1} - \left[\frac{d}{dx} f_{y^{(n)}} \delta y^{(n-2)} \right]_{x_0}^{x_1} + \dots + (-1)^n \int_{x_0}^{x_1} \frac{d^n}{dx^n} f_{y^{(n)}} \delta y dx.$$

Учитывая граничные условия, при которых для $x = x_0$ и $x = x_1$ вариации $\delta y = \dots = y^{(n)} = 0$ для первой вариации функционала, получим

$$\delta J = \int_{x_0}^{x_1} \left(f_y - \frac{d}{dx} f_{y'} + \frac{d^2}{dx^2} f_{y''} + \dots + (-1)^n \frac{d^n}{dx^n} f_{y^{(n)}} \right) \delta y dx = 0.$$

Так δy произвольная функция, равная нулю только на концах интервала, то в силу леммы Лагранжа первый множитель тождественно равен нулю

$$f_y - \frac{d}{dx} f_{y'} + \frac{d^2}{dx^2} f_{y''} + \dots + (-1)^n \frac{d^n}{dx^n} f_{y^{(n)}} \equiv 0.$$

Таким образом, функция $y = y(x)$, реализующая экстремум функционала

$$J[y(x)] = \int_{x_0}^{x_1} f[x, y' \dots y^{(n)}] dx,$$

должна быть решением дифференциального уравнения Эйлера-Пуассона

$$f_y - \frac{d}{dx} f_{y'} + \frac{d^2}{dx^2} f_{y''} + \dots + (-1)^n \frac{d^n}{dx^n} f_{y^{(n)}} = 0,$$

при $2n$ граничных условий

$$\begin{aligned} y(x_0) = y_0, \quad \{ y'(x_0) = y'_0, \quad y^{(n)}(x_0) = y_0^{(n)}, \\ y(x_1) = y_1, \quad \{ y'(x_1) = y'_1, \quad y^{(n)}(x_1) = y_1^{(n)} \end{aligned}$$

Пример. Однородная тяжелая балка закреплена в точках A и B .

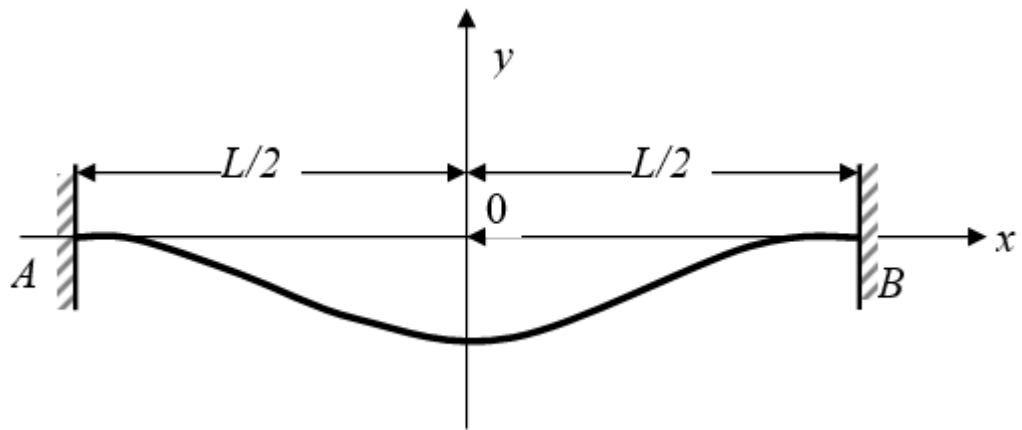


Рис. 1.8. Изогнутая ось балки

Требуется найти форму изогнутой оси балки (рис. 1.8).

Для составления вариационной задачи используем известный принцип: в состоянии устойчивого равновесия потенциальная энергия балки достигает минимума.

Обозначим через μ масса единицы длины балки; ds – элемент изогнутой оси и пусть $y(x)$ – уравнение оси.

Потенциальная энергия упругих сил

$$\Pi_y = \frac{1}{2} \mu \int_0^l \left(\frac{d\alpha}{ds} \right)^2 ds,$$

где μ коэффициент, зависящий от модуля упругости материала;

α угол между касательной к оси балки и осью x ;

ℓ – длина балки равна 2ℓ .

Потенциальная энергия от сил тяжести

$$\Pi_T = \rho \int_0^l y ds.$$

Полная потенциальная энергия

$$\Pi = \Pi_y + \Pi_T = \int_{-l}^l \left[\frac{1}{2} \mu \left(\frac{d\alpha}{ds} \right)^2 + \rho y \right] ds.$$

Учитывая, что $ds = \sqrt{1 + y'^2} dx$ и $\frac{d\alpha}{ds}$ - кривизна оси балки для потенциальной энергии получим

$$\Pi = \int_{-\ell}^{+\ell} \left[\frac{1}{2} \mu \frac{(y'')^2}{(1 + y'^2)^{5/2}} + \rho y \sqrt{1 + y'^2} \right] dx.$$

В положении равновесия $y(x)$ является экстремалью функционала Π . Точное решение для экстремали достаточно громоздко, однако для малых прогибов вторыми степенями y можно пренебречь и для функционала получим

$$\Pi = \int_{-l}^{+l} \left(\frac{1}{2} \mu y'^2 \rho y \right) dx,$$

уравнение Эйлера-Пуассона примет вид

$$\rho + \mu \frac{d^2}{dx^2} y'' = 0,$$

$$\text{следовательно } y^{(4)} = -\frac{\rho}{\mu} \quad \text{и} \quad y = -\frac{\rho}{24\mu} x^4 + C_3 x + C_2 x^2 + C_1 x + C_0$$

Постоянные интегрирования определяются из граничных условий

$$\begin{aligned} y(-l) &= y(l) = 0, \\ y'(-l) &= y'(l) = 0, \end{aligned}$$

из симметрии изгиба следует, что $C_3 = C_1 = 0$, т. к. это коэффициенты при нечетных степенях.

$$-\frac{\rho}{24\mu} l^4 + C_2 l^2 + C_0 = 0,$$

$$-\frac{\rho}{6\mu} l^3 + 2C_2 = 0$$

Решая систему, окончательно получим

$$y = \frac{\rho}{24\mu}(-x^4 + 2l^2x^2 - l^4).$$

Наибольший прогиб будет в середине балки при $x = 0$

$$J_{\max} = -\frac{\rho}{24\mu}l^4.$$

Раздел 2. Линейное программирование

Основные понятия. Примеры моделей, приводящих к задачам линейного программирования

Линейное программирование является одной из основных частей того раздела современной математики, который получил название **математического программирования**. В общей постановке задачи этого раздела выглядят следующим образом.

Имеются переменные $x_1, x_2, \dots, x_n$ и функция этих переменных $f(x_1, x_2, \dots, x_n)$, которая носит название **целевой функции**. Ставится задача: найти экстремум (максимум или минимум) целевой функции $f(x)$ при условии, что переменные $x_1, x_2, \dots, x_n$ принадлежат некоторой области G :

$$\begin{cases} f(x) \rightarrow \text{extr} \\ x \in G \end{cases}.$$

В зависимости от вида функции $f(x)$ и области G различают разделы математического программирования: квадратичное программирование, выпуклое программирование, целочисленное программирование и т.д. Подробнее об этом будет сказано ниже.

Линейное программирование характеризуется тем, что:

а) функция $f(x)$ является **линейной** функцией;

б) область G определяется системой **линейных** равенств или неравенств.

Чтобы понять, откуда берутся задачи линейного программирования, рассмотрим некоторые, уже ставшие классическими, примеры подобных задач.

Общая производственная задача

Рассмотрим деятельность некоторого производственного предприятия. Пусть наше предприятие может производить n видов продукции. Для их производства затрачивается m видов материальных ресурсов (сырье, денежные ресурсы, людские ресурсы и т.д.).

Технологией производства товара j назовем набор чисел $a_{ij}, i=1..m, j=1..n$, показывающий, какое количество i -го ресурса необходимо для производства единицы продукции вида j .

Получается **технологическая матрица** $A=[a_{ij}]$, которая полностью описывает технологические потребности производства.

Пусть у нас имеется b_i запасов i -го ресурса, и мы планируем произвести x_j единиц j -ой продукции. Так как мы не можем выйти за пределы имеющихся у нас ресурсов, то наш план производства $x = (x_1, x_2, \dots, x_n)$ должен удовлетворять ограничениям

[illegible]

Ясно также, что $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$. Обозначим через c_j цену единицы j -ой продукции. Так как мы стремимся получить максимальную прибыль от нашего производства, получается такая задача

$$\begin{aligned} \max \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j \leq b_i, i = 1..m \\ x_j \geq 0, j = 1..n. \end{aligned}$$

Задача о диете

Задача о диете возникает при составлении наиболее экономного (т.е. наиболее дешевого) рациона питания животных, удовлетворяющего определенным требованиям.

Предположим, что в нашем распоряжении имеется n видов кормов (сено, зерно, комбикорм, соль и т.д.). Рациональная диета должна доставлять животному определенные компоненты или питательные вещества (белки, жиры, углеводы, витамины, микроэлементы и т.д.). Пусть таких компонент будет m .

Обозначим через a_{ij} количество i -го компонента, содержащегося в единице веса j -го продукта. Матрица $A = [a_{ij}]$, называется **матрицей питательности**.

Рацион кормления должен указать, какое количество j -го продукта скармливается животному за определенный срок (например, за месяц). Он означает, что за этот срок животное должно получить x_j единиц каждого вида корма $j = 1..n$.

Что же требуется от рациона? Во-первых, должны быть выполнены определенные требования, которые заключаются в том, что за указанный срок животное должно получить не менее определенного количества каждого компонента (не менее определенного количества белков, жиров, витаминов и т.д.). Обозначим через b_i то минимальное количество i -го компонента, которое должно получить животное.

Пусть стоимость единицы веса j -го продукта равна c_j . Тогда наши суммарные затраты будут следующими: $c_1 x_1 + c_2 x_2 + \dots + c_n x_n$.

При этом должны быть выполнены ограничения:

[illegible]

То есть мы должны обеспечить минимальные потребности животных в каждом из компонентов. По смыслу задачи ясно, что переменные $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$.

Так как мы минимизируем наши затраты, получаем следующую задачу:

$$\begin{aligned} & \min \sum_{j=1}^n c_j x_j \\ & \sum_{j=1}^n a_{ij} x_j \geq b_i, i = 1..m \\ & x_j \geq 0, j = 1..n. \end{aligned}$$

Задача о посевной площади

Пусть у нас имеется m полей. Площадь каждого – $S_1, S_2, \dots, S_m$. На этих полях можно посеять n видов сельскохозяйственных культур. Обозначим через a_{ij} урожайность j культуры на i поле. Матрицу $A = [a_{ij}]$ будем называть матрицей **урожайности**. Пусть у нас есть минимальный план сбора по каждой культуре b_j . Мы производим сельскохозяйственную продукцию на продажу. Пусть c_j цена за единицу j культуры. Нам требуется, чтобы все поля были полностью засеяны, и при этом получалась бы максимальная прибыль. Обозначим через x_{ij} план посева j культуры на i поле. Ясно, что $x_{ij} \geq 0$. Получаем следующую задачу:

$$\begin{aligned} \max \sum_{j=1}^n c_j (\sum_{i=1}^m a_{ij} x_j) \\ \sum_{j=1}^n x_{ij} = S_i, i = 1..m \\ \sum_{i=1}^m a_{ij} x_{ij} \geq b_j, j = 1..n \\ x_{ij} \geq 0, i = 1..m, j = 1..n. \end{aligned}$$

Мы не будем рассматривать примеры других задач линейного программирования. Отметим лишь, что они встречаются очень часто при оптимизации самых разнообразных производственных и экономических задач.

Различные формы задачи линейного программирования

Прежде чем рассматривать формы, договоримся об обозначениях. Для более краткой записи мы будем использовать векторную или матричную запись. Под **векторами** мы будем понимать **векторы-столбцы**, например,

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \text{ или } b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}.$$

Иногда будем рассматривать векторы-строки:

$c = (c_1, c_2, \dots, c_n)$. Матрица A (технологий, питательности, урожайности и т.д.) будет:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Заметим, что $c_1 x_1 + c_2 x_2 + \dots + c_n x_n$ есть не что иное, как **скалярное произведение** векторов c и x . Поэтому в **векторной форме** задача ЛП примет вид

$$\min(c, x)$$

$$A_1x_1 + A_2x_2 + \dots + A_nx_n \geq b$$

$$x \geq 0.$$

Здесь $A_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \dots \\ a_{mj} \end{pmatrix}$ вектор-столбец коэффициентов при переменной x_j .

А в матричной она будет выглядеть так:

$$\min cx$$

$$Ax \geq b$$

$$x \geq 0.$$

Задачу $\min(c, x)$ $A_1x_1 + A_2x_2 + \dots + A_nx_n \geq b$, $\min cx$ $Ax \geq b$ мы будем называть $x \geq 0.$ $x \geq 0.$

задачей линейного программирования (ЛП) в стандартной форме

Заметим, что в дальнейшем мы будем рассматривать задачи только на минимум, поскольку переход от задачи на максимум к задаче на минимум достигается умножением коэффициентов целевой функции на 1.

Кроме задачи в стандартной форме мы будем рассматривать каноническую

форму задачи ЛП: $\min cx$
 $Ax = b$
 $x \geq 0.$

Переход от стандартной формы задачи ЛП к канонической

Рассмотрим задачу (1.3). Введем m неотрицательных переменных:

$$x_{n+1} \geq 0, \dots, x_{n+m} \geq 0$$

$$x_{n+i} = \sum_{j=1}^n a_{ij}x_j - b_i, i=1..m.$$

Теперь наша задача ЛП будет выглядеть так:

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

$$\sum_{j=1}^n a_{ij} x_j - x_{n+i} = b_i, i = 1..m$$

$$x_j \geq 0, j = 1..n + m.$$

$$\min c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

$$a_{11} x_1 + a_{12} x_2 + \dots + a_{1n} x_n \geq b_1$$

.....

$$a_{r1} x_1 + a_{r2} x_2 + \dots + a_{rn} x_n \geq b_r$$

$$a_{r+11} x_1 + a_{r+12} x_2 + \dots + a_{r+1n} x_n = b_{r+1}$$

.....

$$a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n = b_m$$

$$x_1 \geq 0, \dots, x_k \geq 0$$

$$k = 0..n, r = 0..m.$$

Очевидно, что при $k=n, r=0$ получается задача ЛП в канонической форме, а при $k=n, r=m$ - задача ЛП в стандартной форме. То есть каноническая и стандартная есть частные случаи задачи ЛП в общей форме. Однако часто требуется переходить от задачи в общей форме к задачам в канонической и стандартной формах.

Геометрическая интерпретация задач линейного программирования

Для понимания всего дальнейшего полезно знать и представлять себе геометрическую интерпретацию задач линейного программирования, которую можно дать для случаев $n=2$ и $n=3$. Мы будем рассматривать стандартную форму задачи ЛП.

Наиболее наглядна эта интерпретация для случая $n=2$, т.е. для случая двух переменных x_1 и x_2 . Пусть нам задана задача линейного программирования в стандартной форме:

$$\begin{aligned}
& \min c_1 x_1 + c_2 x_2 \\
& a_{11} x_1 + a_{12} x_2 \geq b_1 \\
& a_{21} x_1 + a_{22} x_2 \geq b_2 \\
& \dots\dots\dots \\
& a_{m1} x_1 + a_{m2} x_2 \geq b_m \\
& x_1 \geq 0, x_2 \geq 0.
\end{aligned}$$

Рассмотрим на плоскости декартову систему координат и каждой паре чисел (x_1, x_2) поставим в соответствие точку на этой плоскости.

Обратим, прежде всего, внимание на ограничения $x_1 \geq 0$ и $x_2 \geq 0$. Они из всей плоскости вырезают лишь её первую четверть. Рассмотрим теперь, какие области соответствуют неравенствам вида $a_{i1} x_1 + a_{i2} x_2 \geq b_i$. Сначала рассмотрим область, соответствующую равенству $a_{i1} x_1 + a_{i2} x_2 = b_i$. Как известно, это прямая линия (см. рис. 2.1).

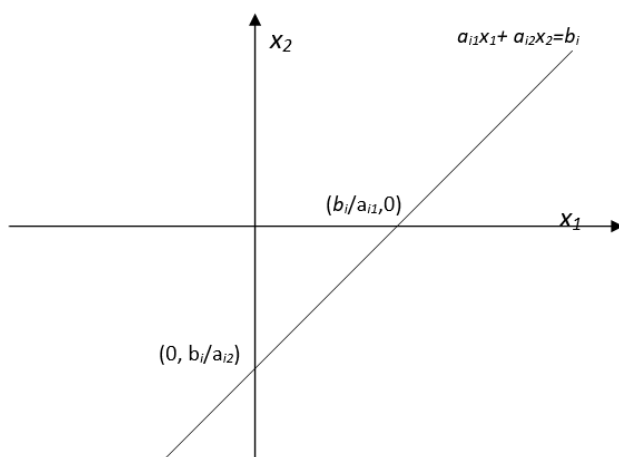


Рис. 2.1. Прямая, задаваемая уравнением $a_{i1} x_1 + a_{i2} x_2 = b_i$.

Неравенство вида $a_{i1} x_1 + a_{i2} x_2 \geq b_i$ определяет полуплоскость, все точки которой лежат либо на этой прямой, либо “выше” ее.

Вернёмся теперь к задаче линейного программирования. Там имеется m неравенств. Каждое такое неравенство определяет полуплоскость.

Нас интересуют те точки, которые удовлетворяют всем этим m неравенствам, т.е. точки, принадлежащие **всем этим полуплоскостям**

одновременно. Следовательно, область, определяемая неравенствами задачи (1.15), геометрически изображается общей частью (пересечением) всех полуплоскостей, определяемых отдельными ограничениями (к ним, естественно, надо добавить ограничения $x_1 \geq 0$ и $x_2 \geq 0$).

Как уже говорилось выше, эта область называется областью допустимых решений задачи линейного программирования.

Не приводя строгих доказательств, укажем те случаи, которые могут получиться.

1. Возможен случай, когда неравенства **противоречат друг другу**, и допустимая область вообще **пуста**.
2. Область имеет вид ограниченного **выпуклого многоугольника** (см. рис.2.2).
3. Получается неограниченный выпуклый многоугольник, имеющий вид, подобный изображенному на рис. 2.3. Мы будем называть такую область **выпуклым многогранным множеством**.

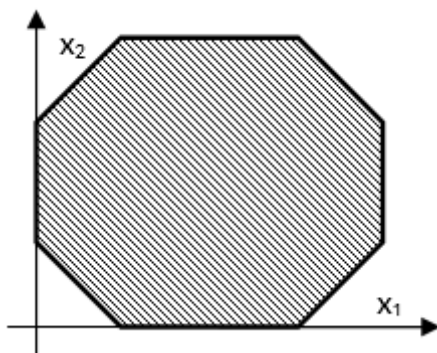


Рис.2.2. Многоугольник

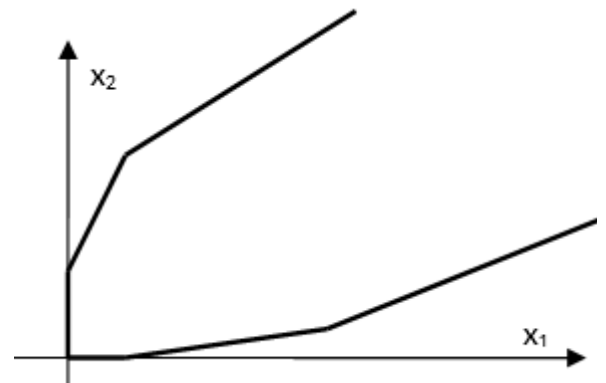


Рис.2.3. Выпуклое многогранное
множество

Вернёмся к исходной задаче линейного программирования. В ней, кроме системы неравенств, есть еще **целевая функция** $c_1x_1 + c_2x_2 = d$. При

фиксированном значении d это прямая. Если рассматривать d как параметр, то получается семейство параллельных прямых. При этом векторы $n^1 = (c_1, c_2)$ и $n^2 = (-c_1, -c_2)$ являются векторами нормали. Оба они перпендикулярны прямым семейства. В направлении одного из них целевая функция возрастает, в направлении другого – убывает. Мы будем считать, что в направлении вектора n^1 целевая функция убывает.

А теперь сведём всё вместе. Итак, надо решить задачу

$$\begin{aligned} \min & c_1 x_1 + c_2 x_2 \\ & a_{11} x_1 + a_{12} x_2 \geq b_1 \\ & a_{21} x_1 + a_{22} x_2 \geq b_2 \\ & \dots\dots\dots \\ & a_{m1} x_1 + a_{m2} x_2 \geq b_m \\ & x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

Ограничения задачи вырезают на плоскости некоторый многоугольник. Пусть при некотором d , прямая $c_1 x_1 + c_2 x_2 = d$ пересекает допустимую область. Это пересечение даёт значения переменных x_1 и x_2 , которые являются допустимыми решениями.

Уменьшая d , мы начнем двигать прямую, и её пересечение с допустимой областью будет изменяться (см. рис. 2.4). В конце концов, эта прямая выйдет на границу допустимой области как правило, это будет одна из **вершин многоугольника**. Дальнейшее уменьшение d приведёт к тому, что пересечение прямой $c_1 x_1 + c_2 x_2 = d$ с допустимой областью будет пусто.

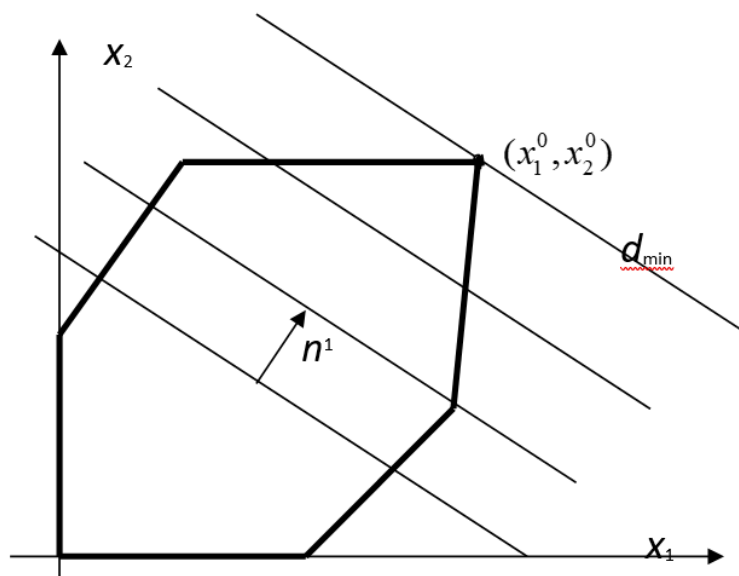


Рис.2.4. Единственное решение задачи ЛП.

Поэтому то положение прямой, при котором она вышла на граничную точку допустимой области, и даст решение задачи, а соответствующее значение d и будет оптимальным значением целевой функции.

Оптимальное решение, как правило, соответствует какой-то вершине многоугольника, изображающего допустимую область. И лишь в том случае, когда прямая $c_1x_1 + c_2x_2 = d$ параллельна одной из сторон многоугольника, может случиться так, что решение не будет единственным. Но и в этом случае вершины, соответствующие границам этой стороны, дают оптимальные решения нашей задачи линейного программирования. Таким образом, вершины допустимой области играют в решении задач линейного программирования особую роль.

Выпуклые множества и выпуклые функции

Рассмотрим n -мерное евклидово пространство R^n , и пусть

$x = (x_1, x_2, \dots, x_n)$ точка в этом пространстве.

Определение. Рассмотрим две точки $x^1 = (x_1^1, x_2^1, \dots, x_n^1)$ и $x^2 = (x_1^2, x_2^2, \dots, x_n^2)$, принадлежащие R^n . Множество точек $x = (x_1, x_2, \dots, x_n)$, которые могут быть представлены в виде

$$x = \alpha x^1 + (1 - \alpha)x^2, 0 \leq \alpha \leq 1,$$

называется **выпуклой комбинацией** точек x^1 и x^2 , или **отрезком**, соединяющим точки x^1 и x^2 . Сами точки x^1 и x^2 называются **концами отрезка**.

Определение. Пусть в R^n заданы k точек $x^1, x^2, \dots, x^k$. Точка $x = \sum_{i=1}^k \alpha_i x^i, \alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1$,

называется **выпуклой линейной комбинацией** точек $x^1, x^2, \dots, x^k$.

Пусть $X \subset R^n$ есть некоторая область в пространстве R^n .

Определение. Множество $X \subset R^n$ называется **выпуклым**, если из того, что $x^1 \in X$ и $x^2 \in X$, следует, что $x = \alpha x^1 + (1 - \alpha)x^2, 0 \leq \alpha \leq 1 \Rightarrow x \in X$. Другими словами, X выпуклое множество, если оно, вместе с любыми двумя своими точками, содержит в себе отрезок, соединяющий эти точки.

По определению будем считать **пустое множество выпуклым**.

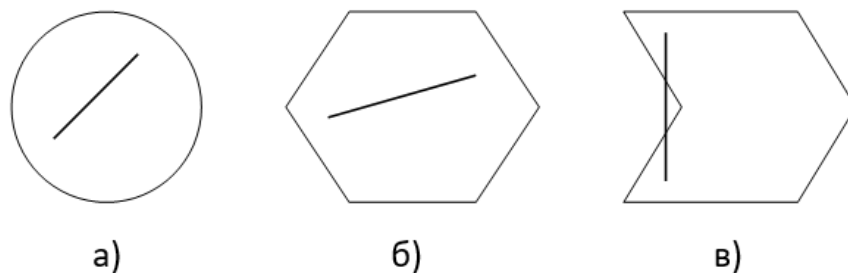


Рис. 2.5. Примеры выпуклых и невыпуклых множеств

На этих рисунках "а" и "б" - выпуклые множества, а "в" не является выпуклым множеством, так как в нём есть такая пара точек, что соединяющий их отрезок не весь принадлежит этому множеству. Справедливы следующие теоремы:

Теорема 1. Пересечение выпуклых множеств является выпуклым множеством.

Теорема 2. Пусть X -выпуклое множество. Тогда любая выпуклая комбинация точек, принадлежащих этому множеству, также принадлежит этому множеству.

Определение. Суммой двух множеств $X_1 \subset R^n, X_2 \subset R^n$ назовем множество $X = \{x \mid x = x^1 + x^2, x^1 \in X_1, x^2 \in X_2\}$, состоящее из всех попарных сумм элементов множеств $X_1 \subset R^n, X_2 \subset R^n$.

Определение. Крайней точкой выпуклого множества называется любая его точка, которая не является внутренней точкой никакого отрезка, целиком принадлежащего этому множеству, то есть точка $x=(x_1, x_2, \dots, x_n)$ выпуклого множества X называется крайней, если ее нельзя представить в виде

$$x \neq \frac{x^1 + x^2}{2}, \forall x^1, x^2 \in X, x^1 \neq x^2.$$

Определение. Гиперплоскостью в пространстве R^n мы будем называть множество точек $x=(x_1, x_2, \dots, x_n)$, удовлетворяющих уравнению

$$p_1 x_1 + p_2 x_2 + \dots + p_n x_n = d, ((p, x) = d),$$

где $p=(p_1, p_2, \dots, p_n) \neq 0$ называется вектором нормали.

Теорема 3 (о разделяющей гиперплоскости). Пусть $X \subset R^n$ выпуклое замкнутое множество. Точка $a \notin X, a = (a_1, a_2, \dots, a_n)$. Тогда существует гиперплоскость с нормалью $p \neq 0$, такая что $(p, x) \leq d \forall x \in X$, а $(p, a) > d$.

Определение. Гиперплоскость $(p, x) = d$ называется опорной к множеству X в точке $b \in X$, если $\forall x \in X$ выполняются соотношения $(p, x) \leq d$ и $(p, b) = d$.

Определение. Множество $K \subseteq R^n$ называется выпуклым конусом, если $\forall x', x'' \in K$ оно содержит и все точки $x = \alpha x' + \beta x'', \alpha \geq 0, \beta \geq 0$.

Из этого определения непосредственно следует, что выпуклый конус является выпуклым множеством. Кроме того, вместе с любым вектором x он

содержит и луч $\alpha x, \alpha \geq 0$. Очевидно, что всякий выпуклый конус содержит начало координат. Нетрудно доказать следующую теорему.

Теорема 4. Всякая гиперплоскость, опорная к выпуклому конусу K , проходит через начало координат.

Определение. Конечное пересечение гиперплоскостей и полупространств мы будем называть *выпуклым многогранным множеством*.

Определение. *Крайние точки выпуклого многогранного множества мы будем называть вершинами.*

Определение. Множество X , представимое в виде

$$X = \{x \mid x = \sum_{i=1}^k \alpha_i x^i + \sum_{j=1}^l \beta_j y^j\}; x^i, y^j \in X; \alpha_i \geq 0, \beta_j \geq 0; \sum_{i=1}^k \alpha_i = 1,$$

называется *выпуклым многогранным множеством*.

Теорема. Приведенные выше определения *выпуклого многогранного множества эквивалентны*.

Мы не будем здесь приводить доказательство данной теоремы. Заметим, что всякое выпуклое многогранное множество X можно представить в виде $X=M+K$, где M – многогранник, а K – выпуклый конус.

Определение. Пусть X – выпуклое множество. Функция $f(x)$ определена на множестве X . Функция $f(x)$ называется *выпуклой функцией* на множестве X , если $\forall x', x'' \in X$

$$f(\alpha x' + (1-\alpha)x'') \leq \alpha f(x') + (1-\alpha)f(x''), 0 \leq \alpha \leq 1.$$

При этом, если неравенство строгое, то функция $f(x)$ называется *строго выпуклой функцией* на множестве X , если неравенство имеет противоположный знак, то *вогнутой функцией* на множестве X .

Базисные, допустимые и оптимальные решения.

Рассмотрим задачу ЛП в канонической форме:

$$\min(cx) \quad (a)$$

$$Ax=b \quad (b)$$

$$x \geq 0 \quad (c)$$

Определение. Вектор x , удовлетворяющий системе линейных уравнений (b), называется *решением* задачи ЛП.

Определение. Решение задачи ЛП, удовлетворяющее соотношению (c), называется *допустимым решением* задачи ЛП.

Определение. Допустимое решение задачи ЛП, доставляющее минимум целевой функции (a), называется *оптимальным решением*.

Определение. Решение задачи ЛП называется *базисным*, если векторы столбцы матрицы ограничений, соответствующие всем ненулевым координатам этого решения, образуют линейно независимую систему векторов.

Замечание. В дальнейшем (если не оговаривается противное) будем предполагать, что ранг матрицы ограничений равен m – числу уравнений в системе (b). Кроме того, будем считать, что $n > m$, так как если $n = m$, то система будет крамеровской и, значит, имеет единственное решение, которое может быть либо допустимым, либо нет.

Определение. Допустимое базисное решение называется *вырожденным*, если число ненулевых координат в нем строго меньше, чем $m = rk(A)$.

Определение. Задача ЛП называется *вырожденной*, если она имеет хотя бы одно вырожденное допустимое базисное решение.

Определение. m линейно независимых векторов столбцов матрицы ограничений A называются *базисом*, если они соответствуют всем ненулевым координатам некоторого допустимого базисного решения.

Пример

$$\min(x_1 + x_2 + x_3 + x_4)$$

$$x_1 + 0x_2 + x_3 + 0x_4 = 1$$

$$2x_1 + x_2 + 0x_3 - x_4 = 2$$

$$x_i \geq 0, i = 1..4.$$

$x^1 = (0, 2, 1, 0)$ - базисное допустимое невырожденное решение,

$x^2 = (0.5, 1, 0.5, 0)$ - небазисное допустимое решение,

$x^3 = (1, 0, 0, 0)$ - базисное допустимое вырожденное решение,

$x^4 = (0, 1, 1, -1)$ - небазисное недопустимое решение,

$x^5 = (0, 0, 1, -2)$ - базисное недопустимое решение,

$x^6 = (0, 0, 0, 0)$ - не является решением данной задачи ЛП.

Теорема 1. Множество всех допустимых решений задачи ЛП есть выпуклое замкнутое множество.

Поскольку доказательство теоремы очевидно, мы здесь его приводить не будем.

Теорема 2. Если задача ЛП имеет допустимое решение, то она имеет и базисное допустимое решение.

Теорема 3. Пусть $x = (x_1, x_2, \dots, x_n)$ - допустимое решение задачи ЛП. x тогда и только тогда будет базисным допустимым решением задачи ЛП, когда точка $x = (x_1, x_2, \dots, x_n)$ является вершиной области допустимых решений этой задачи.

Из теоремы вытекает конечность числа вершин области допустимых решений любой задачи ЛП. В самом деле, пусть задача ЛП дана в канонической форме. Тогда, в силу теоремы, каждой вершине соответствует линейно независимая подсистема системы векторов $A_1, A_2, \dots, A_n$. При этом различным вершинам соответствуют различные подсистемы. Последнее очевидно, если

учесть, что совокупности номеров положительных координат различных вершин различны, так как иначе имели бы не единственное представление вектора b в виде линейной комбинации одних и тех же линейно независимых векторов, что невозможно. Так как число различных подсистем конечной системы векторов $A_1, A_2, \dots, A_n$ конечно, то конечно и число вершин области допустимых решений.

Теорема 4. Если задача ЛП имеет оптимальное решение, то она имеет и базисное оптимальное решение (без доказательства).

Фактически любое оптимальное решение может быть представлено в виде выпуклой линейной комбинации базисных оптимальных решений задачи ЛП.

В заключение данного параграфа рассмотрим следующую теорему:

Теорема 5. Пусть дана система m -мерных векторов $A_1, A_2, \dots, A_n$ ранга m ($m < n$) и $A_1, A_2, \dots, A_l, \dots, A_m$ – ее базис. Заменим в этом базисе какой-нибудь из векторов, например, A_l , на вектор A_k , $k > m$. Тогда полученная система векторов $A_1, A_2, \dots, A_k, \dots, A_m$ также является базисом этой системы тогда и только тогда, когда в разложении вводимого вектора A_k по базису $A_1, A_2, \dots, A_l, \dots, A_m$

$$A_k = x_{1k}A_1 + \dots + x_{lk}A_l + \dots + x_{mk}A_m$$

коэффициент при заменяемом векторе A_l отличен от нуля, то есть $x_{lk} \neq 0$.

Симплекс-метод

Построение допустимого базисного решения

Рассмотрим задачу ЛП в канонической форме:

$$\min(cx), \quad Ax=b, \quad x \geq 0 \tag{2.1}$$

С помощью элементарных преобразований ее можно привести к задаче следующего вида:

является решением задачи ЛП (2.2). Когда это решение будет допустимым? Рассмотрим два случая:

1. $x_{ik} \leq 0 \quad \forall i$. В этом случае при любом выборе $\theta \geq 0$, решение x^1 будет допустимым. Будет ли оно базисным? Если $\theta > 0$, то векторов столбцов, которые соответствуют положительным координатам, будет $m+1$, а любые $m+1$ столбца в m -мерном пространстве линейно зависимы. Значит, в этом случае решение базисным не будет.

2. $\exists x_{ik} > 0$. В этом случае выберем $\theta \geq 0$ так, чтобы решение x^1 было допустимым. Для этого требуется, чтобы

$$x_{i0} - \theta x_{ik} \geq 0, \text{ откуда получаем: } \theta \leq \frac{x_{i0}}{x_{ik}} \text{ для } x_{ik} > 0. \text{ Если мы выберем } \theta \geq 0:$$

$$\theta = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}} = \frac{x_{l0}}{x_{lk}},$$

То все координаты вектора x^1 будут неотрицательными, а значит, это будет допустимое решение задачи. Будет ли оно базисным?

Если мы рассмотрим l -ую координату решения x^1 , то, очевидно, она равна 0, а значит, l -ый вектор выходит из базиса, а k -ый вектор входит в базис. На основании теоремы о замене вектора в базисе, поскольку коэффициент $x_{lk} > 0$, делаем вывод, что решение x^1 будет базисным.

Критерий оптимальности

Снова рассмотрим частный случай задачи ЛП. Напомним, что вектор $x^0 = (x_{10}, x_{20}, \dots, x_{m0}, 0, \dots, 0)$ является допустимым базисным решением (д.б.р.) этой задачи. Вычислим значение целевой функции для этого вектора. Обозначим его через $z(x^0)$:

$$z(x^0) = c_1 x_{10} + c_2 x_{20} + \dots + c_m x_{m0}.$$

Обозначим $z_k = c_1x_{1k} + c_2x_{2k} + \dots + c_mx_{mk}$ и $\Delta_k = z_k - c_k = (c_{\bar{o}}, A_k) - c_k$,

где $(x_{1k}, x_{2k}, \dots, x_{mk})^T$ - коэффициенты разложения вектора A_k по текущему базису, а $c_{\bar{o}} = (c_1, c_2, \dots, c_m)$ - вектор, составленный из коэффициентов целевой функции, соответствующих базисным векторам. Величины Δ_k будем называть оценками.

Симплекс-таблица. Преобразование симплекс-таблицы.

Для удобства проведения расчетов данные задачи заносят в специальные таблицы, которые называют симплекс-таблицами. Пусть x^0 - начальное д.б.р. задачи. Рассмотрим структуру симплекс-таблицы.

Таблица 2.1

Симплекс-таблица

			c_1	...	c_l	...	c_m	...	c_k	...	c_n
Б	С_Б	A₀	A₁	...	A_l	...	A_m	...	A_k	...	A_n
A_1	c_1	x_{10}	1	...	0	...	0	...	x_{1k}	...	x_{1n}
...	...	...	...	...	...	...	...	...	...	...	...
A_l	c_l	x_{l0}	0	...	1	...	0	...	x_{lk}	...	x_{ln}
...	...	...	...	...	...	...	...	...	...	...	...
A_m	c_m	x_{m0}	0	...	0	...	1	...	x_{mk}	...	x_{mn}
		$z(x^0)$	Δ_1	...	Δ_l	...	Δ_m	...	Δ_k	...	Δ_n

Итак, если $\Delta_j \leq 0$ для любого j , то x^0 - на основании теоремы 7 - оптимальное решение, и преобразования симплекс-таблицы не требуется. Поэтому рассмотрим случай, когда $\exists \Delta_j > 0$. Найдем $\max \Delta_j = \Delta_k$. Если при этом

$\forall i=1..k, x_{ik} \leq 0$, то на основании теоремы 8 целевая функция задачи неограничена. В этом случае также не требуется пересчета симплекс-таблицы.

Если же $\exists x_{ik} > 0$, то вычислим $\theta = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}} = \frac{x_{l0}}{x_{lk}}$. В этом случае на основании теоремы 6 возможно улучшение начального д.б.р., если ввести в базис вектор

A_k , а вывести вектор A_l . Рассмотрим, как изменятся коэффициенты разложения всех векторов в симплекс-таблице при таком преобразовании.

$$x_{1j}A_1 + x_{2j}A_2 + \dots x_{lj}A_l + \dots + x_{mj}A_m = A_j$$

$$x_{1k}A_1 + x_{2k}A_2 + \dots x_{lk}A_l + \dots + x_{mk}A_m = A_k.$$

- разложение произвольного вектора по текущему базису. Так как $x_{lk} > 0$, то из можно выразить вектор A_l

$$A_l = -\frac{x_{1k}}{x_{lk}}A_1 - \frac{x_{2k}}{x_{lk}}A_2 - \dots + \frac{1}{x_{lk}}A_k + \dots - \frac{x_{mk}}{x_{lk}}A_m.$$

Подставим полученное соотношение в (2.8) и приведем подобные члены:

$$A_j = (x_{1j} - \frac{x_{lj}x_{1k}}{x_{lk}})A_1 + (x_{2j} - \frac{x_{lj}x_{2k}}{x_{lk}})A_2 + \dots + \frac{x_{lj}}{x_{lk}}A_k + \dots + (x_{mj} - \frac{x_{lj}x_{mk}}{x_{lk}})A_m$$

Фактически мы получили разложение произвольного вектора по новому базису, в котором вектор A_l заменен на вектор A_k . Если обозначить эти коэффициенты через x'_{ij} , то получаем следующие формулы пересчета:

$$x'_{ij} = \begin{cases} x_{ij} - \frac{x_{lj}x_{ik}}{x_{lk}}, i \neq l \\ \frac{x_{lj}}{x_{lk}}, i = l \end{cases}$$

Определение. Строку l мы будем называть ведущей строкой. Столбец k – ведущим столбцом, а элемент x_{lk} – ведущим элементом.

Формулы выше означают, что для перехода к новому базису, когда мы вектор A_l заменяем на вектор A_k , нужно ведущую строку l поделить на ведущий элемент x_{lk} , а затем из каждой строки симплекс-таблицы вычесть преобразованную ведущую строку, умноженную на соответствующий элемент данной строки.

Рассмотрим, как будут выглядеть коэффициенты столбца A_k в новом базисе.

Они получаются, если в формулы (2.11) подставить $j=k$:

$$x'_{ik} = \begin{cases} 1, i = l \\ 0, i \neq l \end{cases}$$

Отсюда следует, что для пересчета симплекс-таблицы достаточно с помощью элементарных преобразований над строками добиться того, чтобы вновь вводимый в базис столбец имел такую структуру – на месте ведущего элемента – 1, а остальные элементы столбца – 0.

Алгоритм симплекс-метода

Рассмотрим задачу ЛП. Будем считать ее невырожденной. В этих предположениях приведем алгоритм симплекс-метода.

Алгоритм

1. Начать с д.б.р. $x^0 = (x_{10}, x_{20}, \dots, x_{m0}, 0, \dots, 0)$.
2. Вычисляем оценки по формуле $\Delta_j = (c_{\bar{o}}, A_j) - c_j$.
3. Если $\Delta_j \leq 0, \forall j \in 1..n$, то x^0 - оптимальное решение, конец, иначе
4. Находим $\max \Delta_j = \Delta_k$ (это правило носит рекомендательный характер, поскольку можно выбрать любую положительную оценку).
5. Если $x_{ik} \leq 0, \forall i \in 1..m$, то целевая функция неограничена, конец, иначе
6. Находим $\theta = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}} = \frac{x_{l0}}{x_{lk}}$.
7. Элемент x_{lk} становится ведущим, и делается преобразование симплекс-таблицы по формулам.

8. Переход на 2.

В случае, если задача ЛП не вырождена, алгоритм, очевидно, конечен. В самом деле, в этом случае $\theta > 0$, а значит, действует замечание к теореме 6, и новое д.б.р. будет таким, что значение целевой функции будет меньше, чем на предыдущем д.б.р. Ну а поскольку допустимых базисных решений конечное число, а на каждом шаге алгоритма мы обязательно переходим к новому д.б.р., то за конечное число итераций алгоритм закончит свою работу либо в п. 3, либо в п.5.

Симплекс-метод в общем случае. Зацикливание

Рассмотрим задачу ЛП. Предположим, что она вырожденная. Не умаляя общности, можно считать, что уже начальное д.б.р. x^0 вырожденное и что у него первые $r < m$ координат положительны, а остальные равны 0. То есть,

$$x^0 = (x_{10}, x_{20}, \dots, x_{r0}, 0, \dots, 0), x_{i0} > 0, i = 1..r.$$

Запустим алгоритм симплекс-метода, стартовав с д.б.р. x^0 . Возможны следующие ситуации:

1. $\Delta_j \leq 0, \forall j \in 1..n$. Тогда x^0 – оптимальное решение.

2. $\exists \Delta_k > 0, x_{ik} \leq 0$. Тогда целевая функция в задаче 2.2 будет неограничена.

3. $\exists \Delta_k > 0, \exists x_{ik} > 0. \theta = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}} = \frac{x_{l0}}{x_{lk}} > 0$. В этом случае выполняется обычная

итерация симплекс-метода. Целевая функция уменьшается. Мы переходим к новому д.б.р., и процесс продолжается дальше.

4. $\exists \Delta_k > 0, \exists x_{ik} > 0. \theta = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}} = \frac{x_{l_1 0}}{x_{lk}} = \frac{x_{l_2 0}}{x_{lk}} = \dots = \frac{x_{l_s 0}}{x_{lk}} = 0$. В этом случае новое

д.б.р. $x^1 = (x_{10} - \theta x_{1k}, x_{20} - \theta x_{2k}, \dots, x_{r0} - \theta x_{rk}, 0, \dots, \theta, \dots, 0)$ будет полностью совпадать с д.б.р. x^0 . Такая ситуация называется **холостым ходом** симплекс-метода. То есть, само решение не меняется, меняется только базис.

Пусть мы сделали несколько холостых ходов. Тогда возможны такие ситуации:

1. $\Delta_j \leq 0, \forall j \in 1..n$. Тогда получено оптимальное решение.

2. $\exists \Delta_k > 0, x_{ik} \leq 0$. Тогда целевая функция в задаче ЛП будет неограничена.

3. $\exists \Delta_k > 0, \exists x_{ik} > 0. \theta = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}} = \frac{x_{i0}}{x_{lk}} > 0$. В этом случае целевая функция

уменьшается. Мы переходим к новому д.б.р., и процесс продолжается дальше.

4. Встретился базис, который уже был раньше. Такое явление называется “зацикливанием”. То есть мы будем находиться на одном и том же д.б.р., бесконечно перебирая базисы по одному и тому же циклу.

Следует отметить, что явление “зацикливания” встречается крайне редко. Имеется лишь небольшое количество задач ЛП, на которых и демонстрируется это явление. Тем не менее, было разработано несколько процедур для борьбы с ним. Мы рассмотрим одну такую процедуру.

Заметим, что проблемы возникают при определении вектора, который должен быть выведен из базиса. В случае вырожденности таких векторов может быть несколько. Это возможно тогда, когда при вычислении θ

$$\theta = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}} = \frac{x_{i_1 0}}{x_{lk}} = \frac{x_{i_2 0}}{x_{lk}} = \dots = \frac{x_{i_s 0}}{x_{lk}} = 0, \quad x_{i_1 0} = x_{i_2 0} = \dots = x_{i_s 0} = 0.$$

И если выбирать один из этих векторов произвольно, то и возможно зацикливание. Поэтому рассмотрим процедуру борьбы с “зацикливанием”, которая производит упорядочение таких векторов.

Строим множество:

$$I_0 = \left\{ i \mid \theta_0 = \min_{x_{ik} > 0} \frac{x_{i0}}{x_{ik}} \right\}. \text{ Если } I_0 = \{l_0\}, \text{ то этот вектор выводится из базиса и}$$

процесс выбора вектора на этом завершается, иначе строится множество:

$$I_1 = \left\{ i \in I_0 \mid \theta_1 = \min_{x_{ik} > 0} \frac{x_{i1}}{x_{ik}} \right\}. \text{ Если } I_1 = \{l_1\}, \text{ то этот вектор выводится из базиса}$$

и процесс выбора вектора на этом завершается, иначе продолжаем до тех пор, пока очередное множество индексов

$$I_u = \left\{ i \in I_{u-1} \mid \theta_u = \min_{x_{ik} > 0} \frac{x_{iu}}{x_{ik}} \right\} \text{ состоит из одного вектора } I_u = \{l_u\} \text{ и именно этот}$$

вектор выводится из базиса.

Двойственность в линейном программировании

Вспомним общую производственную задачу.

Рассмотрим деятельность некоторого производственного предприятия. Пусть наше предприятие может производить n видов продукции. Для их производства затрачивается m видов материальных ресурсов (сырье, денежные ресурсы, людские ресурсы и т.д.).

Технологией производства товара j назовем набор чисел $a_{ij}, i = 1..m, j = 1..n$, показывающий, какое количество i -го ресурса необходимо для производства единицы продукции j .

Получается **технологическая матрица** $A = [a_{ij}]$, которая полностью описывает технологические потребности производства.

Пусть у нас имеется b_i запасов i ресурса, и мы планируем произвести x_j единиц j -ой продукции. Так как мы не можем выйти за пределы имеющихся у нас ресурсов, то наш план производства $x = (x_1, x_2, \dots, x_n)$ должен удовлетворять ограничениям

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

Ясно также, что $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$. Так как мы стремимся иметь максимальную прибыль от нашего производства, получается такая задача

$$x_j \geq 0, j = 1..n$$

Наряду с этой задачей рассмотрим другую задачу, которая тесно связана с данной. Пусть $y = (y_1, y_2, \dots, y_m)$ - оценки ресурсов (“теневые” цены). Тогда суммарная оценка всех ресурсов задается так:

$y_1b_1 + y_2b_2 + \dots + y_mb_m$. Мы будем ее минимизировать. При этом должны быть выполнены такие ограничения:

$$a_{1n}y_1 + a_{2n}y_2 + \dots + a_{mn}y_m \geq c_n.$$

Итак, получается такая задача:

$$y_i \geq 0, i = 1..m.$$

Такие задачи и называются парой двойственных задач. Свойство двойственности, очевидно, взаимно.

Различные формы прямой и двойственной задач. Принципы двойственности

Рассмотрим, как будет выглядеть пара двойственных задач в зависимости от формы задачи.

$$\begin{array}{ll} \max cx & \min yb \\ Ax \leq b & (3.1) \quad yA \geq c \\ x \geq 0 & y \geq 0 \end{array}$$

$$\begin{array}{ll} \min cx & \max yb \\ Ax = b & (3.3) \quad yA \leq c \\ x \geq 0 & \end{array}$$

$$\begin{array}{ll} \max cx & \min yb \\ Ax = b & (3.5) \quad yA \geq c \\ x \geq 0 & \end{array}$$

$$\begin{array}{ll} \min cx & \max yb \\ Ax \geq b & (3.7) \quad yA \leq c \\ x \geq 0 & y \geq 0 \end{array}$$

Как мы видим, действуют общие принципы формирования двойственной задачи. Приведем эти принципы:

1. Если исходная задача – задача на минимум, то двойственная задача – на максимум. И наоборот.
2. Столбец свободных членов прямой задачи становится коэффициентами целевой функции двойственной задачи.
3. Коэффициенты целевой функции прямой задачи становятся столбцом свободных членов ограничений двойственной задачи.
4. Матрица ограничений транспонируется.
5. Знак неравенства в задачах в стандартной форме меняется на противоположный.

6. Если в исходной задаче есть ограничения-равенства, то в двойственной задаче на соответствующие переменные не накладываются условия неотрицательности. И наоборот.

Первая теорема двойственности

Теорема 2. Первая теорема двойственности. Для любой пары двойственных задач выполняется одна и только одна из следующих альтернатив:

1. Обе задачи имеют оптимальные решения, причем их оптимумы (значения целевых функций) совпадают.
2. Одна из задач имеет допустимые решения, а вторая не имеет. Тогда первая задача имеет неограниченную целевую функцию.
3. Ни одна из задач не имеет допустимых решений.

Условия дополняющей нежесткости

Вначале рассмотрим условия дополняющей нежесткости в слабой форме для пары двойственных задач ЛП

Теорема 3 (условия дополняющей нежесткости в слабой форме). Пусть $\bar{x}$ и $\bar{y}$ - допустимые решения пары двойственных задач. Для того, чтобы эти допустимые решения были оптимальными, необходимо и достаточно выполнение следующих условий:

$$\begin{cases} \bar{y}(b - A\bar{x}) = 0 \\ (\bar{y}A - c)\bar{x} = 0. \end{cases}$$

Замечание. Условия дополняющей нежесткости других пар двойственных задач будут отличаться

Запишем условия (3.5) в координатной форме.

$$\begin{cases} \sum_{i=1}^m \bar{y}_i (b_i - \sum_{j=1}^n a_{ij} \bar{x}_j) = 0 \\ \sum_{j=1}^n (\sum_{i=1}^m a_{ij} \bar{y}_i - c_j) \bar{x}_j = 0 \end{cases}$$

Так как каждое слагаемое в этих суммах неотрицательно, то выполнение условий (3.6) возможно лишь в случае:

$$\begin{cases} \bar{y}_i (b_i - \sum_{j=1}^n a_{ij} \bar{x}_j) = 0, i = 1..m \\ (\sum_{i=1}^m a_{ij} \bar{y}_i - c_j) \bar{x}_j = 0, j = 1..n \end{cases}$$

Отсюда следует, что если $\bar{x}_j > 0$, то $\sum_{i=1}^m a_{ij} \bar{y}_i = c_j$ и, наоборот,

если $\sum_{i=1}^m a_{ij} \bar{y}_i > c_j$, то $\bar{x}_j = 0$. Аналогично, если $\bar{y}_i > 0$, то $\sum_{j=1}^n a_{ij} \bar{x}_j = b_i$, и, наоборот,

если $\sum_{j=1}^n a_{ij} \bar{x}_j > b_i$, то $\bar{y}_i = 0$. Отсюда и название - “условия дополняющей нежесткости”.

Транспортная задача

Рассмотрим транспортную задачу, т. е. задачу, в которой речь идет о рациональной перевозке некоторого однородного продукта от производителей к потребителям.

Пусть имеется m пунктов производства однородного продукта (добыча руды в карьерах, производство автобусов, кондитерских изделий, компьютеров и т.д.) и n пунктов потребления этого продукта. Мощности пунктов производства составляют a_i ($i = \overline{1, m}$) единиц однородного продукта, а потребности каждого j -го пункта потребления равны b_j ($j = \overline{1, n}$) единиц. Известны затраты c_{ij} на перевозку единицы продукта от i -го поставщика j -му потребителю. Составить такой план перевозок, при котором суммарные затраты на все перевозки были бы

наименьшими. Пусть спрос и предложение совпадают, т.е. $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$. Такую транспортную задачу называют *сбалансированной (закрытой)*. При этом предполагается, что вся продукция от поставщиков будет вывезена и спрос каждого из потребителей будет удовлетворен.

Составим математическую модель задачи. Обозначим через x_{ij} – количество продукта, перевозимого из i -го пункта производства в j -й пункт потребления. Тогда матрица: $X = \|x_{ij}\|$ ($i = \overline{1, m}; j = \overline{1, n}$) – план перевозок.

Матрицу $C = \|c_{ij}\|$ ($i = \overline{1, m}; j = \overline{1, n}$) называют *матрицей затрат* (тарифов).

Внесем исходные данные и перевозки x_{ij} в транспортную таблицу:

Таблица 2.2

Транспортная задача

b_j a_i	b_1	b_2	...	b_n
a_1	c_{11} x_{11}	c_{12} x_{12}	...	c_{1n} x_{1n}
a_2	c_{21} x_{21}	c_{22} x_{22}	...	c_{2n} x_{2n}
...	...	...	...	...
a_m	c_{m1} x_{m1}	c_{m2} x_{m2}	...	c_{mn} x_{mn}

Предположим, что транспортные затраты прямо пропорциональны количеству перевозимого продукта. Тогда суммарные затраты выразятся функцией цели:

$$Z = c_{11}x_{11} + c_{12}x_{12} + \dots + c_{mn}x_{mn},$$

которую необходимо минимизировать при ограничениях:

$$x_{i1} + x_{i2} + \dots + x_{in} = a_i$$

(весь продукт из каждого i -го ($i = \overline{1, m}$) пункта должен быть вывезен полностью),

$$x_{1j} + x_{2j} + \dots + x_{mj} = b_j$$

(спрос каждого j -го ($j = \overline{1, n}$) потребителя должен быть полностью удовлетворен).

Из условия задачи следует, что все

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

Итак, математическая модель *сбалансированной* транспортной задачи имеет вид:

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

при ограничениях:

$$\begin{cases} \sum_{j=1}^n x_{ij} = a_i \quad (i = \overline{1, m}), \\ \sum_{j=1}^m x_{ij} = b_j \quad (j = \overline{1, n}), \end{cases}$$

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

Раздел 3. Методы решения нелинейных задач математического программирования

Точные методы решения нелинейных задач математического программирования.

Методы оптимизации функции одной переменной

Функции одной переменной

Задача оптимизации, в которой целевая функция задана функцией одной переменной, относится к наиболее простому типу оптимизации задач. Тем не менее анализ задач такого типа занимает важное место в исследованиях. Это связано с тем, что одномерные методы оптимизации часто используются для анализа подзадач, которые возникают при реализации итеративных процедур, ориентированных на решение многомерных задач оптимизации.

Свойства функций одной переменной

Определение: Функция $f(x)$ представляет собой правило, которое позволяет каждому значению x поставить в соответствие единственное значение $y = f(x)$.

В этом случае x называется независимой переменной, y – зависимой переменной (функцией).

Обычно значения независимой переменной x определены в некоторой области $S : x \in S$.

Если S является подмножеством множества действительных чисел, то f называется скалярной функцией, определенной на множестве S .

Например

а) $f(x) = x^3 + 2x^2 - x + 3$ для всех $x \in R$, где R – множество действительных чисел.

б) $f(x) = x^3 + 2x^2 - x + 3$ для всех $x \in S$, $S = \{x | -5 \leq x \leq 5\}, S \subset R$.

В теории оптимизации f - называется целевой функцией, S – областью допустимых значений X (допустимой областью).

Экстремумы функции.

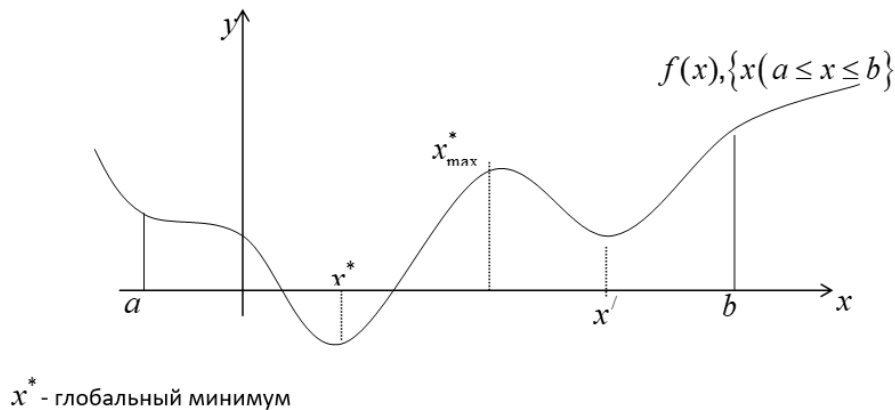


Рис. 3.1. Экстремумы функции

Функция $f(x)$ имеет локальный минимум в точке x^* , если существует окрестность точки x^* такая, что для всех точек этой окрестности $f(x) \geq f(x_0^*)$.

Функция $f(x)$ имеет глобальный минимум в точке x^* если для всех x из области допустимых значений ($x \in S$) справедливо неравенство $f(x) \geq f(x^*)$.

Аналогичные определения глобального и локального максимума можно получить путем замены знака неравенства на противоположный.

Топологические свойства функций

Для выбора метода оптимизации важное значение имеет форма функций, определяющая топологические свойства функций в рассматриваемом интервале. Очевидно, что в зависимости от формы целевой функции и структуры допустимой области следует использовать различные методы для поиска точек оптимума.

Первая классификация функций в соответствии с их формой предполагает их разделение на *непрерывные*, *разрывные* и *дискретные*.

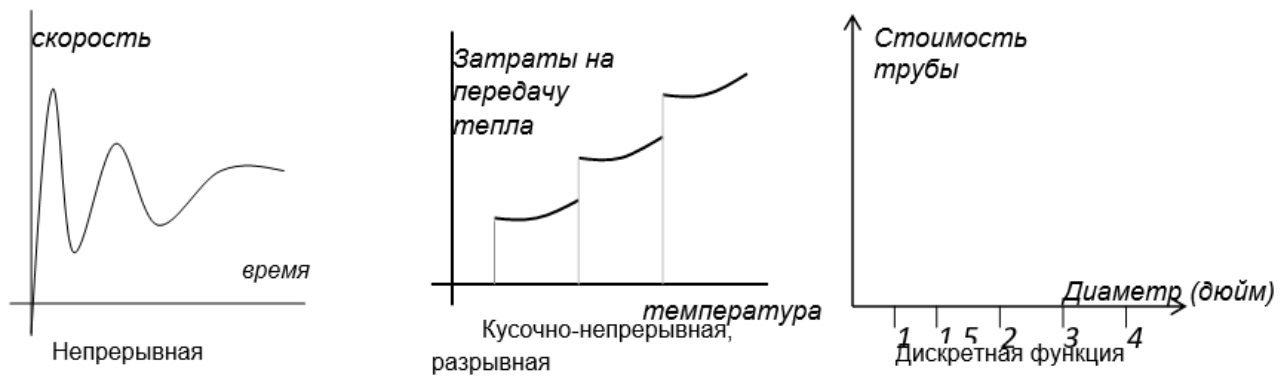


Рис. 3.2. Классификация функций по форме

Монотонные функции

Функция $f(x)$ является монотонной, если для двух произвольных точек x_1 и x_2 , таких, что $x_1 \leq x_2$, выполняется одно из следующих неравенств:

$f(x_1) \leq f(x_2)$ (монотонно возрастающая функция).

$f(x_1) \geq f(x_2)$ (монотонно убывающая функция).

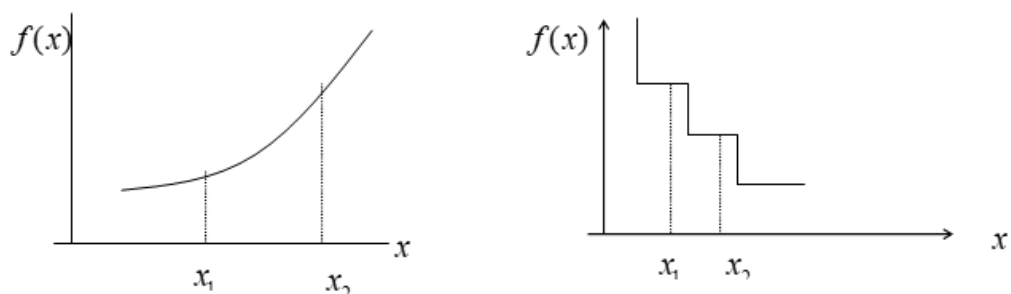


Рис. 3.3. Классификация функций по монотонности

Унимодальные функции

Функция $f(x)$ является унимодальной на отрезке $a \leq x \leq b$ в том и только том случае, если она монотонна по обе стороны от единственной на рассматриваемом интервале оптимальной точки x^* .

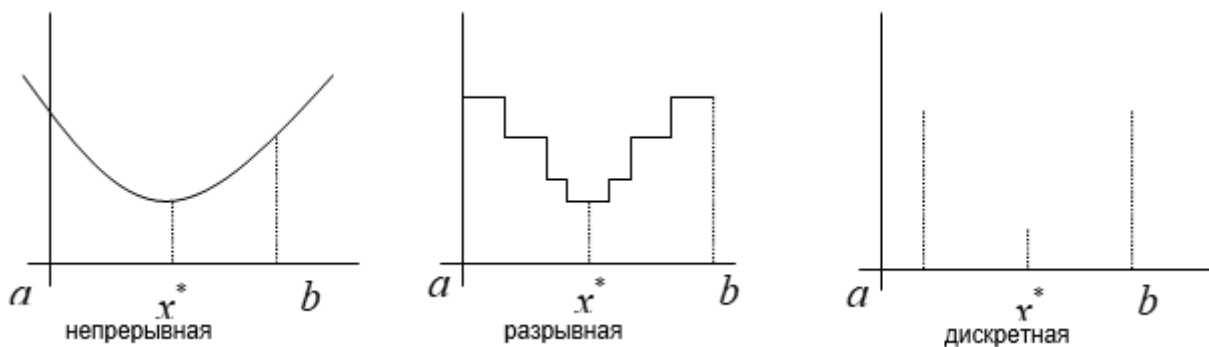


Рис. 3.2. Унимодальные функции

Экстремумы функции

Функция $f(x)$ имеет локальный минимум в точке x_0 , если существует некоторая положительная величина δ , такая, что если $(x - x_0) < \delta$, то $f(x) \geq f(x_0)$, то есть если существует окрестность точки x_0 , такая, что для всех точек этой окрестности $f(x) \geq f(x_0)$.

Функция $f(x)$ имеет глобальный минимум в точке x^* , если для всех x справедливо неравенство $f(x^*) \leq f(x)$.

Замечания.

1. Аналогичные определения глобального максимума и локального максимума можно получить путем замены знака неравенства на противоположный.

2. Если функция обладает свойством унимодальности, то локальный минимум автоматически является глобальным минимумом.

3. Если функция не является унимодальной, то возможно наличие нескольких локальных оптимумов; при этом глобальный минимум можно определить путем нахождения всех локальных оптимумов и выбора наименьшего из них.

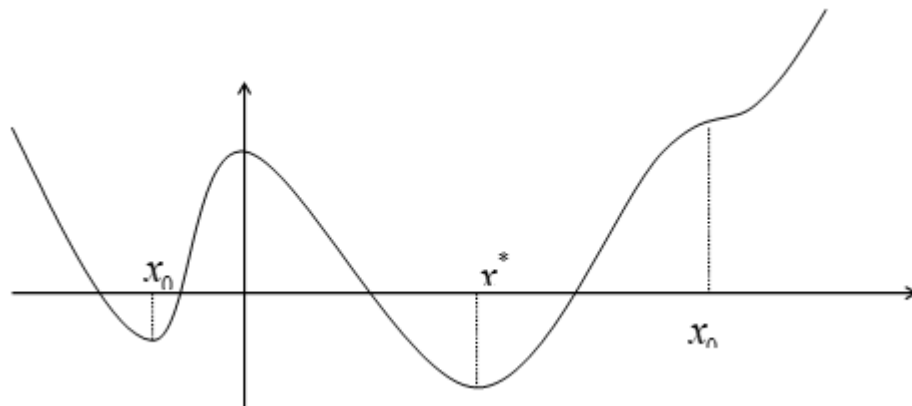


Рис. 3.5. График функции и экстремумы

Классический метод оптимизации.

Классический подход к задаче нахождения значений x^* и x_0 состоит в поиске уравнений, которым эти значения должны удовлетворять.

Можно сформулировать следующее правило:

Если функция $f(x)$ и ее производные непрерывны, то точка x_0^* является точкой экстремума тогда и только тогда, когда n – четное, где n – порядок первой не обращающейся в нуль производной в точке x_0^* .

Если $f''(x_0) < 0$, то в точке x_0 достигается максимум, если $f''(x_0) > 0$, то – минимум.

Пример.

Исследовать функцию $f(x) = x^3 - 2x^2 + x + 1$ на экстремумы.

$f'(x) = 0$; $3x^2 - 4x + 1 = 0$; $(3x - 1)(x - 1) = 0$, т.е. $x_1 = \frac{1}{3}$; $x_2 = 1$ – точки стационарные (особые).

$$f''(x) = 6x - 4;$$

$$f''(1/3) = -2 \text{ в т. } x = \frac{1}{3} \text{ достигается максимум;}$$

$$f''(1) = 2 \text{ в т. } x = 1 \text{ достигается минимум.}$$

Это правило можно обосновать, если рассмотреть разложение функции $f(x)$ в окрестности точки x^* экстремума в ряд Тейлора.

$$f(x^* + h) - f(x^*) = hf'(x^*) + \frac{h^2}{2!} f''(x^*) + \dots$$

При достаточно малом шаге h первое слагаемое доминирует над остальными. Если в точке x^* достигается минимум, то левая часть будет неотрицательной для любого достаточно малого h .

Следовательно, первая производная должна быть равна нулю и это является необходимым условием существования минимума. (Если $f'(x^*) > 0$ то при $h < 0 \rightarrow h \cdot f'(x^*) > 0$).

Так как h^2 всегда больше нуля, то при $f''(x^*) > 0 \rightarrow f'(x^*) < f(x^* + h)$ достигается минимум, а при $f''(x^*) < 0 \rightarrow f'(x^*) > f(x^* + h)$ достигается максимум функции $f(x)$ в точке x^* .

Стационарной точкой называется точка x^* , в которой $\left. \frac{df}{dx} \right|_{x=x^*} = 0$.

Если стационарная точка не соответствует локальному оптимуму, то она является точкой перегиба или *седловой* точкой.

Численные методы поиска экстремума в одномерных нелинейных задачах математического программирования

Методы оптимизации функции одной переменной

Задача оптимизации, в которой целевая функция задана функцией одной переменной, относится к наиболее простому типу оптимизации задач. Тем не менее анализ задач такого типа занимает важное место в исследованиях. Это связано с тем, что одномерные методы оптимизации часто используются для анализа подзадач, которые возникают при реализации итеративных процедур, ориентированных на решение многомерных задач оптимизации.

Свойства функций одной переменной

Определение: Функция $f(x)$ представляет собой правило, которое позволяет каждому значению x поставить в соответствие единственное значение $y = f(x)$.

В этом случае x называется независимой переменной, y – зависимой переменной (функцией).

Обычно значения независимой переменной x определены в некоторой области $S : x \in S$.

Если S является подмножеством множества действительных чисел, то f называется скалярной функцией, определенной на множестве S .

Например

а) $f(x) = x^3 + 2x^2 - x + 3$ для всех $x \in R$, где R – множество действительных чисел.

б) $f(x) = x^3 + 2x^2 - x + 3$ для всех $x \in S$, $S = \{x | -5 \leq x \leq 5\}, S \subset R$.

В теории оптимизации f – называется целевой функцией, S – областью допустимых значений x (допустимой областью).

Экстремумы функции.

Функция $f(x)$ имеет локальный минимум в точке x^* , если существует окрестность точки x^* такая, что для всех точек этой окрестности $f(x) \geq f(x_0^*)$.

Функция $f(x)$ имеет глобальный минимум в точке x^* если для всех x из области допустимых значений ($x \in S$) справедливо неравенство $f(x) \geq f(x^*)$.

Аналогичные определения глобального и локального максимума можно получить путем замены знака неравенства на противоположный.

Классический метод оптимизации

Классический подход к задаче нахождения значений x^* и x_0 состоит в поиске уравнений, которым эти значения должны удовлетворять.

Можно сформулировать следующее правило:

Если функция $f(x)$ и ее производные непрерывны, то точка x_0^* является точкой экстремума тогда и только тогда, когда n – четное, где n – порядок первой не обращающейся в нуль производной в точке x_0^* .

Если $f''(x_0) < 0$, то в точке x_0 достигается максимум, если $f''(x_0) > 0$, то – минимум.

Пример 1.

Исследовать функцию $f(x) = x^3 - 2x^2 + x + 1$ на экстремумы.

$f'(x) = 0$; $3x^2 - 4x + 1 = 0$; $(3x - 1)(x - 1) = 0$, т.е. $x_1 = \frac{1}{3}$; $x_2 = 1$ – точки стационарные (особые).

$$f''(x) = 6x - 4;$$

$$f''(1/3) = -2 \quad \text{– в т. } x = \frac{1}{3} \text{ достигается максимум;}$$

$$f''(1) = 2 \quad \text{– в т. } x = 1 \text{ достигается минимум.}$$

Это правило можно обосновать, если рассмотреть разложение функции $f(x)$ в окрестности точки x^* экстремума в ряд Тейлора.

$$f(x^* + h) - f(x^*) = hf'(x^*) + \frac{h^2}{2!} f''(x^*) + \dots$$

При достаточно малом шаге h первое слагаемое доминирует над остальными. Если в точке x^* достигается минимум, то левая часть будет неотрицательной для любого достаточно малого h .

Следовательно, первая производная должна быть равна нулю и это является необходимым условием существования минимума. (Если $f'(x^*) > 0$ то при $h < 0 \rightarrow h \cdot f'(x^*) > 0$)).

Так как h^2 всегда больше нуля, то при $f''(x^*) > 0 \rightarrow f'(x^*) < f(x^* + h)$ достигается минимум, а при $f''(x^*) < 0 \rightarrow f'(x^*) > f(x^* + h)$ достигается максимум функции $f(x)$ в точке x^* .

Вид экстремума (максимум, минимум) зависит от знака четвертой производной. Ряд можно продолжить, если нулевыми окажутся также третья и четвертая производные функции $f(x)$:

$$f'''(x^*) = 120(x-1)^3, \quad x=1 \quad f'''(x) = 0$$

$$f^{IV}(x) = 360(x-1)^2, \quad f^V(x) = 720(x-1), \quad f^{VI}(x) = 720 > 0$$

Первая не обращающаяся в нуль производная в т. $x=1$ – шестая, $f^{VI} > 0$ – следовательно функция имеет минимум в точке $x=1$.

Определения.

Стационарной точкой называется точка x^* , в

которой $\left. \frac{df}{dx} \right|_{x=x^*} = 0$.

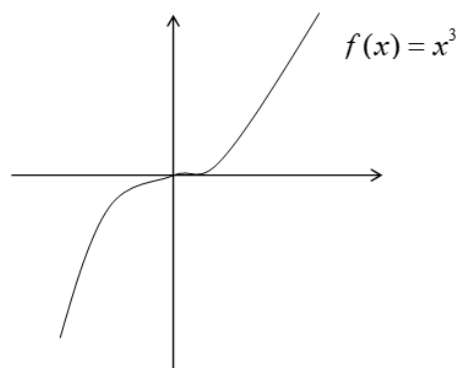


Рис. 3.6. Точка перегиба

Если стационарная точка не соответствует локальному оптимуму, то она является точкой перегиба или *седловой* точкой.

Метод Ньютона

Для функции одной переменной классический подход при поиске значений x , удовлетворяющих необходимому условию существования экстремума, состоит в решении уравнения. $f'(x) = 0$.

Решить такое уравнение не всегда удастся аналитическими методами. Поэтому целесообразно использовать численные методы решения таких уравнений.

Вспомним один из таких методов – метод Ньютона. Приблизительный эскиз кривой $y = f'(x)$ позволяет получить приближенное решение.

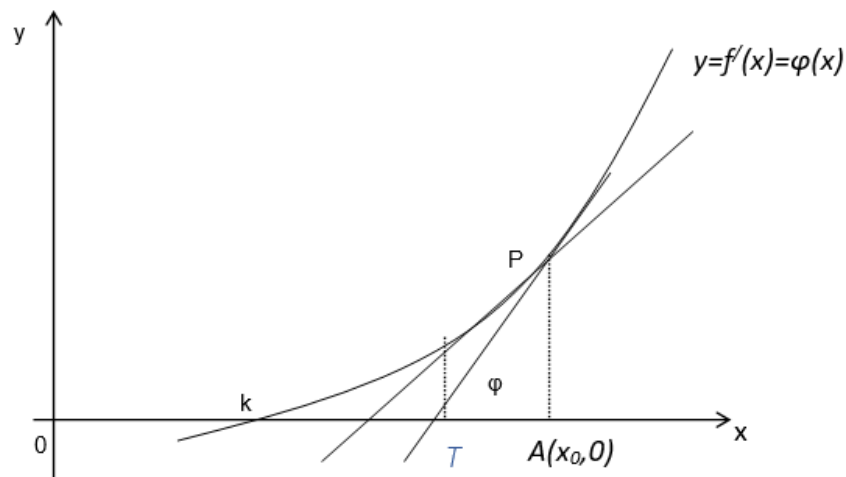


Рис. 3.7. Метод Ньютона

Далее метод Ньютона позволяет улучшить относительно грубую аппроксимацию корня. Пусть PT – касательная к кривой $\varphi(x)$ в точке P первого приближения. T – точка, в которой касательная пересекает ось X . Тогда в общем случае $x = OT$ является лучшей аппроксимацией корня, лежащего в точке k .

Очевидно

$$OT = OA - TA = x_0 - TA$$

$$\frac{PA}{TA} = \operatorname{tg} \psi = \varphi'(x_0),$$

следовательно, $TA = \frac{PA}{\varphi'(x_0)} = \frac{\varphi(x_0)}{\varphi'(x_0)}$ и $x_1 = x_0 = -\frac{\varphi(x_0)}{\varphi'(x_0)}$

Аналогично в общем случае $x_{2+1} = x_2 = -\frac{\varphi(x_2)}{\varphi'(x_2)}$

Метод секущих

Логическая схема поиска предыдущего метода основана лишь на исследовании знака производной. При реализации метода секущих кроме знака производной используется также и ее значение.

Пусть в процессе поиска стационарной точки функция $f(x)$ в интервале (a,b) обнаружены две точки L и R , в которых знаки производных различны.

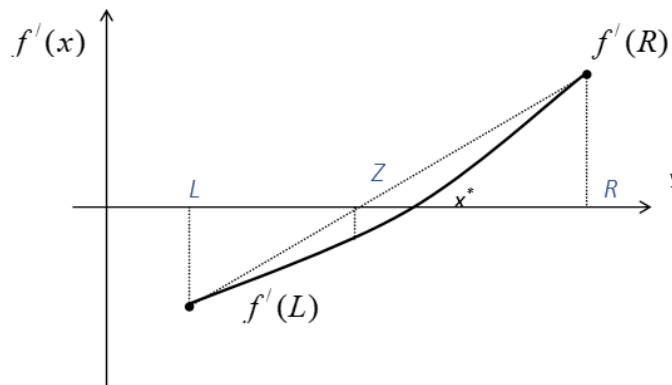


Рис. 3.8. Метод секущих

Метод секущих позволяет аппроксимировать функцию $f'(x)$ на отрезке (L,R) прямой линией, соединяющей точки $f'(L)$ и $f'(R)$. Следующее приближение к точке x^* определяется по формуле $Z = R - \frac{f'(R) \cdot (R - L)}{(f'(R) - f'(L))}$

Если $|f'(Z)| \leq \varepsilon$, поиск следует закончить. В противном случае необходимо выбрать одну из точек L или R таким образом, чтобы знаки производной в этой точке и в точке Z были различны.

Независимо от того, требуется ли найти максимальное или минимальное значение целевой функции можно пользоваться одними и теми же методами

решения экстремальных задач, т.к. задачу минимизации можно превратить в задачу на поиск максимума, если поменять знак целевой функции на противоположный.

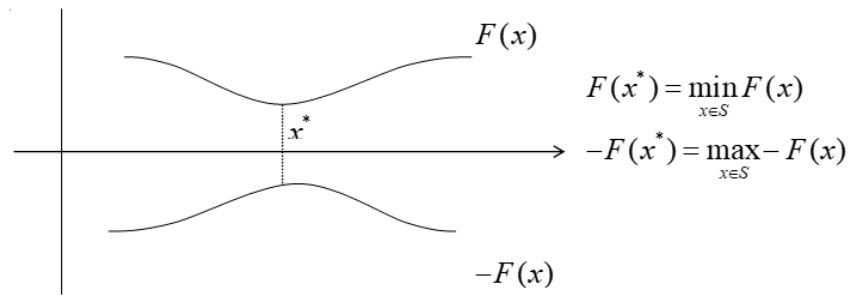


Рис. 3.9. Функции $F(x)$ и $-F(x)$

Аналогично изменением знака целевой функции на противоположный задача на максимум превращается в задачу на минимум.

Функции n переменных

Рассмотрим функцию n действительных переменных $f(x_1, x_2, \dots, x_n) = f(\mathbf{x})$. Точку в n -мерном евклидовом пространстве обозначим вектором столбцом $\mathbf{x}$.

Градиент функции $f(\mathbf{x})$ обозначим $\text{grad}(f(\mathbf{x}))$.

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}; \quad \text{grad}(f(\mathbf{x})) = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix} = \nabla f(\mathbf{x}) = g(\mathbf{x})$$

Матрица Гессе (гессиан) функции $f(\mathbf{x})$ обозначается, как $\mathbf{G}(\mathbf{x})$ и является симметрической матрицей $n \times n$ элементов вида:

$$G_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1^2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{vmatrix}, \quad i = 1 \dots n, \quad j = 1 \dots n.$$

Предположим, что в точке $\mathbf{x}^*$ функция $f(\mathbf{x})$ достигает минимума, тогда справедливо неравенство

$$f(\mathbf{x}^*) \leq f(\mathbf{x}^* + \Delta \mathbf{x}).$$

Полагая функцию дифференцируемой, выполним разложение её в ряд Тейлора в окрестности точки $\mathbf{x}^*$:

$$f(\mathbf{x}^* + \Delta \mathbf{x}) - f(\mathbf{x}^*) = \Delta \mathbf{x}^T \nabla f(\mathbf{x}^*) + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{G}(\mathbf{x}^*) \Delta \mathbf{x} + \dots$$

Сопоставляя два последних выражения, приходим к неравенству:

$$\Delta \mathbf{x}^T \nabla f(\mathbf{x}^*) + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{G}(\mathbf{x}^*) \Delta \mathbf{x} \geq 0;$$

Если $\text{grad} f(\mathbf{x}^*) \neq 0$, то в силу малости $\Delta \mathbf{x}$, первый член разложения будет доминирующим. Но тогда соответствующим подбором $\Delta \mathbf{x}$ можно добиться того, что левая часть неравенства будет отрицательной.

Следовательно, необходимым условием минимума является равенство нулю всех частных производных в точке минимума, т.е.:

$$\left. \frac{\partial f(\mathbf{x})}{\partial x_i} \right|_{\bar{\mathbf{x}} = \bar{\mathbf{x}}^*} = 0, \quad i = 1, 2, \dots, n \quad \text{или} \quad \nabla f(\mathbf{x}^*) = 0.$$

Точка, $\mathbf{x}^*$ в которой $\text{grad} f(\mathbf{x}) = 0$ называется *стационарной*.

В этой точке значение левой части неравенства определяется членом:

$$\frac{1}{2} \Delta \mathbf{x}^T \mathbf{G}(\mathbf{x}^*) \Delta \mathbf{x} = Q(\mathbf{x}).$$

В точке минимума $Q(\mathbf{x})$ должна быть больше нуля или равна нулю (таким образом можно найти стационарную точку $\mathbf{x}^*$ определить в ней матрицу Гессе и, подбирая малые $\Delta \mathbf{x}$ посмотреть, как ведет себя функция $Q(\mathbf{x})$ в окрестности стационарной точки).

Определения:

Функция n переменных $Q(x_1, x_2, \dots, x_n)$ называется квадратичной формой, если $Q(x_1, x_2, \dots, x_n) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j = \mathbf{x}^T \mathbf{A} \mathbf{x}$, где $\mathbf{A} = (a_{ij})$ и $\mathbf{x}^T = (x_1, x_2, \dots, x_n)$.

1) Матрица $\mathbf{A}$ является положительно определенной тогда и только тогда, когда значение квадратичной формы $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ для всех $\mathbf{x} \neq 0$.

2) Матрица $\mathbf{A}$ является положительно полуопределенной, когда значение квадратичной формы $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$ и существует вектор $\mathbf{x} \neq 0$ такой что $\mathbf{x}^T \mathbf{A} \mathbf{x} = 0$;

3) Матрица $\mathbf{A}$ является отрицательно определенной тогда и только тогда, когда $-\mathbf{A}$ есть положительно определенная матрица, т.е. $-\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$ или $\mathbf{x}^T \mathbf{A} \mathbf{x} < 0$;

4) Матрица $\mathbf{A}$ является отрицательно полуопределенной тогда и только тогда, когда $\mathbf{A}$ есть положительно полуопределенная матрица.

5) Матрица $\mathbf{A}$ является неопределенной, если квадратичная форма может принимать как положительные, так и отрицательные значения.

Следовательно:

Необходимыми и достаточными условиями минимума являются $\nabla f(\mathbf{x}^*) = 0$; $\mathbf{G}(\mathbf{x})$ — положительно определена.

По аналогии:

Необходимыми и достаточными условиями максимума являются $\nabla f(\mathbf{x}^*) = 0$; $\mathbf{G}(\mathbf{x})$ — отрицательно определенная матрица.

- Значения параметра, λ для которого существуют нетривиальные (ненулевые) решения системы уравнений, называются собственными числами матрицы $A = [a_{ij}]$.

- Соответствующие собственным числам векторные решения называются собственными векторами матрицы A .

Система уравнений в векторных обозначениях имеет вид:

$$Ax = \lambda x,$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}; \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

Перенеся правую часть уравнения влево, получим $(A - \lambda I)x = 0$, где I - единичная матрица.

Это уравнение обладает нетривиальным решением только тогда, когда определитель равен нулю:

$$|A - \lambda I| = \begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{vmatrix} = 0;$$

Разложение этого определителя приводит к характеристическому уравнению:

$$|A - \lambda I| = (-\lambda)^n + b_{n-1}(-\lambda)^{n-1} + \dots + b_1(-\lambda) + b_0 = 0;$$

Корни этого уравнения $\lambda_1, \lambda_2, \dots, \lambda_n$ и есть собственные числа матрицы A . Соответствующие каждому из этих значений λ_k векторные решения исходного уравнения составят n собственных векторов, $(V_1, V_2, \dots, V_n)$ для которых, очевидно:

$$\left. \begin{array}{l} \mathbf{A}\mathbf{V}_1 = \lambda\mathbf{V}_1 \\ \dots\dots\dots \\ \mathbf{A}\mathbf{V}_n = \lambda\mathbf{V}_n \end{array} \right\}$$

Свойства λ_i и V_i :

а) Все собственные значения действительной симметрической матрицы
есть действительные числа.

б) Собственные векторы действительной симметрической матрицы, соответствующие различным собственным значениям попарно ортогональны.

в) Существует, по крайней мере, одно ортогональное множество векторов, удовлетворяющее условию нормализации т.е. $(V_i, V_i) = 1$.

Диагонализация матриц

Пусть $\lambda_1, \lambda_2, \dots, \lambda_n$ - различные собственные значения симметрической матрицы A , и $V_1, V_2, \dots, V_n$ - соответствующая им система нормализованных собственных векторов.

Рассмотрим матрицу $T = (V_1, V_2 \dots V_n)$ столбцами, которой служат собственные векторы.

Распишем выражение $AT=(AV_1, AV_2, \dots AV_n)$, но для собственных векторов справедливо

$$AV_i = \lambda V_i \text{ следовательно } AT = (\lambda_1 V_1, \lambda_2 V_2, \dots, \lambda_n V_n).$$

Домножим выражение слева на матрицу T^T :

$$\mathbf{T}^T \mathbf{A} \mathbf{T} = \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \\ \vdots \\ \mathbf{V}_n^T \end{pmatrix} (\lambda_1 \mathbf{V}_1, \lambda_2 \mathbf{V}_2, \dots, \lambda_n \mathbf{V}_n) = \begin{pmatrix} \lambda_1 \mathbf{V}_1^T \mathbf{V}_1 & \lambda_2 \mathbf{V}_1^T \mathbf{V}_2 & \dots & \dots & \dots \\ \lambda_1 \mathbf{V}_2^T \mathbf{V}_1 & \lambda_2 \mathbf{V}_2^T \mathbf{V}_2 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \lambda_n \mathbf{V}_n^T \mathbf{V}_1 & \dots & \dots & \dots & \lambda_n \mathbf{V}_n^T \mathbf{V}_n \end{pmatrix}$$

$$\text{Учитывая, что } (V_i, V_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \text{ имеем } T^T A T = \begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \\ & & & \lambda_n \end{pmatrix} = A$$

Произведем в квадратичной форме, $Q(x) = x^T A x$ замену переменных полагая $x = Ty$, тогда $Q(x) = x^T A x = y^T T^T A T y = y^T A y$

$$\text{или развертывая выражение имеем } Q(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j = \sum_{i=1}^n \lambda_i y_i^2.$$

Таким образом, мы получаем возможность судить о свойствах матрицы по ее собственным числам

1. Квадратичная форма $x^T A x$ является положительно определенной тогда, и только тогда, когда все собственные числа матрицы A положительны.

2. Квадратичная форма $x^T A x$ является положительно полуопределенной, тогда и только тогда, когда все собственные значения A неотрицательны и хотя бы одно из них равно нулю.

3. Квадратичная форма $x^T A x$ является неопределенной, когда матрица A обладает как положительными, так и отрицательными собственными значениями.

Пример 1:

Исследовать экстремальные точки функции

$$f(x) = x_1^2 + x_2^2 + x_3^2 - 4x_1 - 8x_2 - 12x_3 + 100$$

$$\text{grad}(f) = \begin{vmatrix} 2x_1 - 4 \\ 2x_2 - 8 \\ 2x_3 - 12 \end{vmatrix} = 0 \text{ при } x_1 = 2, x_2 = 4, x_3 = 6$$

$$\text{Гессиан } G(f) = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{vmatrix}$$

Характеристическое уравнение матрицы

$$\det \begin{vmatrix} 2-\mu & 0 & 0 \\ 0 & 2-\mu & 0 \\ 0 & 0 & 2-\mu \end{vmatrix} = 0; (2-\mu)(2-\mu)(2-\mu) = 0$$

Все собственные значения положительные и равны 2. Матрица положительно определена.

Пример 2:

Найти максимумы и минимумы функции $z = 3x^3 - x + y^3 - 3y^2 - 1$.

Необходимые условия

$$\text{grad}(z) = \begin{vmatrix} 9x^2 - 1 \\ 3y^2 - 6y \end{vmatrix} = 0; \text{ или } \frac{\partial z}{\partial x} = 9x^2 - 1 = 0$$

$$\frac{\partial z}{\partial y} = 3y^2 - 6y = 0$$

удовлетворяются при $x = \frac{1}{3}, y = 0; x = -\frac{1}{3}, y = 0; x = \frac{1}{3}, y = 2; x = -\frac{1}{3}, y = 2$.

Достаточное условие

$$\mathbf{G}(z) = \begin{vmatrix} 18x & 0 \\ 0 & 6y - 6 \end{vmatrix} = 0;$$

Характеристическое уравнение $(18x - \mu)(6y - 6 - \mu) = 0$

Рассматривая все точки, видим:

а) $(6 - \mu)(-6 - \mu) = 0; \mu_1 = 6; \mu_2 = -6$ — неопределенная матрица, седловая точка;

б) $(-6 - \mu)(-6 - \mu) = 0; \mu_1 = -6; \mu_2 = -6$

матрица отрицательно определена - точка максимума.

в) $(6 - \mu)(12 - 6 - \mu) = 0; \mu_1 = 6; \mu_2 = 6$ — точка минимума.

г) $(-6 - \mu)(6 - \mu) = 0$; $\mu_1 = -6$; $\mu_2 = 6$ — седловая точка.

Метод последовательных исключений

Для симметрической матрицы O найти такую матрицу P и диагональную матрицу D , чтобы выполнялось равенство $O = P^T D P$. Рассмотрим процедуру на примере.

$$\text{Положим } O = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

а) Выберем ведущий элемент $d_1 = a$ и разделим на него первую строку

б) Умножим полученную первую строку на b и вычтем из второй

$$O_2 = \begin{pmatrix} 1 & b/a \\ 0 & c - \frac{b^2}{a} \end{pmatrix}$$

в) Выберем в качестве ведущего элемента ненулевой элемент второй строки $d_2 = \left(c - \frac{b^2}{a}\right)$ и разделим на d_2 вторую строку матрицы O_2 :

$O_3 = \begin{pmatrix} 1 & b/a \\ 0 & 1 \end{pmatrix}$. Получением треугольной матрицы O_3 (на диагонали которой 1)

процесс исключения заканчивается.

$$\text{Положим } P = O_3 = \begin{pmatrix} 1 & b/a \\ 0 & 1 \end{pmatrix}, \quad D = \begin{pmatrix} d_1 & 0 \\ 0 & d_2 \end{pmatrix} = \begin{pmatrix} a & 0 \\ 0 & c - \frac{b^2}{a} \end{pmatrix}.$$

В том что действительно выполняется равенство $O = P^T D P$ можно убедиться непосредственно.

Произведем замену переменных в квадратичной форме $Q = x^T O x$; $x = P y$
тогда

$$Q(\mathbf{x}) = \mathbf{x}^T \mathbf{O} \mathbf{x} = \mathbf{y}^T \mathbf{P}^T \mathbf{O} \mathbf{P} \mathbf{y} = \mathbf{y}^T \mathbf{D} \mathbf{y}$$

Таким образом, о свойствах квадратичной формы $Q(\mathbf{x})$ и, соответственно, симметрической матрицы $\mathbf{O}$ мы можем судить по свойствам квадратичной формы $\mathbf{y}^T \mathbf{D} \mathbf{y}$ и, соответственно, диагональной матрицы $\mathbf{D}$.

Главный минор

Ведущий главный минор порядка k матрицы порядка $n \times n$ строится путем исключения из матрицы последних $n - k$ строк и соответствующих столбцов.

Положительно определенная матрица:

а) Все диагональные элементы должны быть > 0

б) Все ведущие главные миноры > 0

Положительно полуопределенная матрица:

а) Все диагональные элементы должны быть неотрицательны

б) Все ведущие главные миноры неотрицательны.

Пример:

$$Q(\mathbf{x}) = 4x_1 + 5x_1x_2 + 3x_2^2$$

$$Q(\mathbf{x}) = (x_1, x_2) \begin{pmatrix} 4 & 2 \\ 3 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Главные миноры равны $4 > 0$ и $\begin{vmatrix} 4 & 2 \\ 3 & 3 \end{vmatrix} = 6 > 0$ таким образом матрица

положительно определенная.

Примеры:

$\mathbf{Q} = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$ - положительно определенная;

$\mathbf{Q} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ - положительно полуопределенная;

$$\mathbf{O} = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} - \text{отрицательно полуопределенная};$$

$$\mathbf{O} = \begin{pmatrix} -2 & 1 \\ 1 & -3 \end{pmatrix} - \text{отрицательно определенная};$$

$$\mathbf{O} = \begin{pmatrix} 1 & -1 \\ 1 & -2 \end{pmatrix} - \text{неопределенная};$$

$$\mathbf{O} = \begin{pmatrix} 12 & -2 \\ -2 & 10 \end{pmatrix} - \text{положительно определенная}.$$

Оптимизация при наличии ограничений.

Ограничения в виде равенств

Рассмотрим общую задачу минимизации, содержащую K ограничений в виде равенств:

$$\text{минимизировать } f(x_1, x_2, \dots, x_N)$$

$$\text{при ограничениях } h_i(x_1, x_2, \dots, x_N) = 0$$

где $i = 1, 2, \dots, K$.

Если систему из K уравнений можно разрешить относительно K неизвестных, то эти переменные можно из целевой функции вообще исключить.

Наличие ограничений в виде равенств позволяет в этом случае уменьшить размерность исходной задачи с N до $N-K$ и получить при этом задачу безусловной минимизации.

Пример.

$$\text{Минимизировать } f(\mathbf{x}) = x_1 x_2 x_3$$

$$\text{при ограничении } h_1(\mathbf{x}) = x_1 + x_2 + x_3 - 1 = 0.$$

Выразим переменную x_3 из равенства $x_3 = 1 - x_1 - x_2$ и исключим ее из целевой функции $f(\mathbf{x}) = x_1 x_2 (1 - x_1 - x_2)$. Получим, таким образом, задачу

безусловной минимизации функции двух переменных $\min f(x_1, x_2) = x_1 x_2 (1 - x_1 - x_2)$. Для оптимизации такой функции можно использовать один из методов, рассмотренных выше.

Однако возможны ситуации, когда аналитическое выражение одной переменной через другие получить невозможно. Например, если ограничение задать в виде $h_1(x) = x_1^2 x_3 + x_2 x_3^2 + x_2^{-1} x_1 = 0$. Кроме того, при наличии большого числа ограничений процесс исключения переменных становится трудоемкой задачей.

Множители Лагранжа

Метод множителей Лагранжа позволяет преобразовать задачу с ограничениями в виде равенств в задачу безусловной минимизации.

Рассмотрим задачу минимизации функции двух переменных с учетом одного ограничения:

минимизировать $f(x, y)$

при ограничении $h(x, y) = 0$.

В соответствии с методом множителей Лагранжа исходная задача преобразуется в следующую: $L(x, y, \lambda) = f(x, y) + \lambda h(x, y)$. Функция $L(x, y, \lambda)$ называется функцией Лагранжа, а λ - неизвестный коэффициент, который носит название множителя Лагранжа. Необходимые условия минимума функции Лагранжа могут быть записаны в следующем виде:

$$\left. \begin{aligned} \frac{\partial L}{\partial x} &= \frac{\partial f}{\partial x} + \lambda \frac{\partial h}{\partial x} = 0 \\ \frac{\partial L}{\partial y} &= \frac{\partial f}{\partial y} + \lambda \frac{\partial h}{\partial y} = 0 \\ \frac{\partial L}{\partial \lambda} &= h(x, y) = 0 \end{aligned} \right\}$$

Решениями этой системы трех уравнений являются значения x^*, y^*, λ^* в точке минимума функции Лагранжа. Однако, как нетрудно заметить, в точке

минимума выполняется ограничение $h(x^*, y^*) = 0$, а при этом минимум функции $L(x^*, y^*, \lambda^*)$ совпадает с минимумом функции $f(x^*, y^*)$.

Действительно, при $h(x, y) = 0$

$$\min L(x, y, \lambda^*) = \min (f(x, y) + \lambda \cdot h(x, y)) = \min f(x, y)$$

Пример.

Минимизировать $f(x, y) = x^2 + y^2$

при ограничении $h(x, y) = 2x + y - 2 = 0$.

Используя функцию Лагранжа, перейдем к задаче безусловной минимизации.

Минимизировать $L(x, y, \lambda) = f(x, y) + \lambda \cdot h(x, y) = x^2 + y^2 + \lambda(2x + y - 2)$. Приравняем компоненты градиента L к нулю:

$$\left. \begin{aligned} \frac{\partial L}{\partial x} &= 2x + 2\lambda = 0 \\ \frac{\partial L}{\partial y} &= 2y + \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= 2x + y - 2 = 0 \end{aligned} \right\} \begin{aligned} x &= -\lambda \\ y &= -\frac{\lambda}{2} \\ 2\lambda + \frac{\lambda}{2} - 2 &= 0; \rightarrow \lambda = -\frac{4}{5} \end{aligned}$$

Таким образом, минимум функции достигается в точке $(x^* = 4/5; y^* = 2/5); \lambda = -4/5$, следовательно, минимум функции $f(x, y)$ при ограничении $h(x, y) = 0$ достигается в точке $(x^* = 4/5; y^* = 2/5)$.

Метод множителей Лагранжа можно распространить на случай функции n переменных при наличии m ограничений в виде равенств.

Рассмотрим задачу минимизации функции $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$, где на переменную $\mathbf{x}$ наложены ограничения $g_1(\mathbf{x}) = 0, g_2(\mathbf{x}) = 0, \dots, g_m(\mathbf{x}) = 0$.

Функция Лагранжа принимает следующий вид $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{k=1}^m \lambda_k g_k$, где λ_k -

множители Лагранжа, то есть неизвестные параметры, которые необходимо определить.

Приравнивая нулю частные производные функции L по компонентам вектора $\mathbf{x}$, получаем систему n уравнений с $n + m$ неизвестными:

$$\begin{aligned}\frac{\partial L(\mathbf{x}, \lambda)}{\partial x_1} &= 0 \\ \frac{\partial L(\mathbf{x}, \lambda)}{\partial x_2} &= 0 \\ &\dots \dots \dots \\ \frac{\partial L(\mathbf{x}, \lambda)}{\partial x_n} &= 0\end{aligned}$$

Ограничения в виде равенств дополняют систему m недостающими уравнениями.

$$\begin{aligned}\frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda_1} &= g_1(\mathbf{x}) = 0 \\ &\dots \dots \dots \\ \frac{\partial L(\mathbf{x}, \lambda)}{\partial \lambda_m} &= g_m(\mathbf{x}) = 0\end{aligned}$$

Решение расширенной системы из $n + m$ уравнений определяет стационарную точку функции L . Таким образом необходимые условия минимума функции $f(\mathbf{x})$ при наличии ограничений-равенств можно записать в следующем виде:

$$\begin{aligned}\frac{\partial L}{\partial x_i} &= \frac{\partial f}{\partial x_i} + \sum_{k=1}^m \lambda_k \cdot \frac{\partial g_k}{\partial x_i} = 0, \quad i = 1, 2, \dots, n. \\ \frac{\partial L}{\partial \lambda_i} &= g_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, m.\end{aligned}$$

Ограничения в виде неравенств

Рассмотрим задачу нелинейного программирования, поставленную в следующем виде: минимизировать функцию $f(\mathbf{x})$ при наличии m ограничений $h_i(\mathbf{x}) \leq 0$, $i = 1, 2, \dots, m$. Ограничения в виде неравенств называются активными в

точке x_k , если $h_i(x_k) = 0$ и неактивными, если в точке x_k выполняется строгое неравенство $h_i(x_k) > 0$. В настоящее время нет метода, гарантирующего существование решения любой подобной задачи.

Ограничения в виде неравенств могут быть преобразованы в ограничения в виде равенств путем добавления к каждому из них неотрицательной *ослабляющей* переменной u_i^2 : $h_i(x) + u_i^2 = 0$.

Таким образом, задача сводится к минимизации функции $f(x)$ при наличии m ограничений в виде равенств. Используем для решения полученной задачи метод множителей Лагранжа.

Сформируем функцию Лагранжа:

$$L(x, \lambda, u) = f(x) + \sum_{i=1}^m \lambda_i [h_i(x) + u_i^2].$$

Необходимыми условиями существования минимума будут следующие:

$$\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial h_i}{\partial x_j} = 0, \quad j = 1, 2, \dots, n,$$

$$\frac{\partial L}{\partial \lambda_i} = h_i(x) + u_i^2 = 0, \quad i = 1, 2, \dots, m,$$

$$\frac{\partial L}{\partial u_i} = 2\lambda_i u_i = 0, \quad i = 1, 2, \dots, m.$$

Умножив последнее уравнение на $\frac{u_i}{2}$ получим $\lambda_i u_i^2 = 0$, то есть

$$\lambda_i h_i(x) = 0$$

Есть также дополнительное условие, которое должно быть выполнено в точке минимума $\lambda_i \geq 0$. Окончательно можно записать следующую систему необходимых условий минимума функции при наличии ограничений в виде неравенств (**Это условия Куна-Таккера**).

$$\begin{cases} \frac{\partial f}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial h_i}{\partial x_j} = 0, & , j=1,2,\dots,n, \\ h_i \leq 0, & , i=1,2,\dots,m, \\ \lambda_i h_i = 0, \\ \lambda_i \geq 0. \end{cases}$$

Пример. 1.

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

$$h(x, y) = x + y + 4 \leq 0;$$

$$\begin{aligned} \left. \begin{aligned} 2x + \lambda &= 0 \\ 2y + \lambda &= 0 \\ \lambda(x + y + 4) &= 0 \end{aligned} \right\} \begin{aligned} x &= -\lambda/2 \\ y &= -\lambda/2 \\ \lambda^2 - 4\lambda &= 0 \rightarrow \lambda = 0 \quad \vee \quad \lambda = 4 \end{aligned} \\ \left. \begin{aligned} x + y + 4 &\leq 0 \\ \lambda &\geq 0 \end{aligned} \right\} \begin{aligned} \lambda &= 4 \\ x &= -2 \\ y &= -2 \end{aligned} \quad x + y + 4 = 0 \end{aligned}$$

Пример. 2.

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

$$h(x, y) = 2x + y - 2 \leq 0;$$

$$\begin{aligned} \left. \begin{aligned} 2x + 2\lambda &= 0 \\ 2y + \lambda &= 0 \\ \lambda(2x + y - 2) &= 0 \end{aligned} \right\} \begin{aligned} x &= -\lambda \\ y &= -\lambda/2 \\ \lambda(2\lambda + \lambda/2 + 2) &= 0; \quad \lambda(\frac{5}{2}\lambda + 2) = 0 \end{aligned} \\ &\rightarrow \lambda = 0, \quad \lambda = -4/5 \\ \left. \begin{aligned} 2x + y - 2 &\leq 0 \\ \lambda &\geq 0 \end{aligned} \right\} \begin{aligned} \lambda &= 0 \\ x &= y = 0 \end{aligned} \quad 2x + y - 2 < 0 \end{aligned}$$

Общая задача нелинейного программирования

Кун и Таккер обобщили метод множителей Лагранжа на случай общей задачи нелинейного программирования с ограничениями как в виде равенств, так и в виде неравенств:

$$\text{Минимизировать } f(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, \dots, x_N)$$

при ограничениях $g_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, M$.

$$h_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, K.$$

Условия Куна-Таккера, определяющие необходимые условия оптимальности для задачи нелинейного программирования, записываются в следующем виде:

$$\frac{\partial f}{\partial x_n} + \sum_{i=1}^M \lambda_i \frac{\partial g_i}{\partial x_n} + \sum_{j=1}^K v_j \frac{\partial h_j}{\partial x_n} = 0; \quad h = 1, 2, \dots, N.$$

$$g_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, M$$

$$h_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, K$$

$$v_j h_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, K$$

$$v_j \geq 0, \quad j = 1, 2, \dots, K$$

Пример.

Минимизировать $f(\mathbf{x}) = x_1^2 - x_2$ при ограничениях $x_1 + x_2 = 6$; $x_1 - 1 \geq 0$; $x_1^2 + x_2^2 \leq 26$.

Решение. Запишем задачу как общую задачу нелинейного программирования в принятых обозначениях :

$$f(\mathbf{x}) = x_1^2 - x_2, g_1(\mathbf{x}) = x_1 + x_2 - 6 = 0, h_1(\mathbf{x}) = 1 - x_1 \leq 0, h_2(\mathbf{x}) = x_1^2 + x_2^2 - 26 \leq 0$$

Вычислим градиенты функций:

$$\nabla f(\mathbf{x}) = (2x_1, -1), \nabla g(\mathbf{x}) = (1, 1), \nabla h_1(\mathbf{x}) = (-1, 0), \nabla h_2(\mathbf{x}) = (2x_1, 2x_2)$$

Первое условие Куна-Таккера принимает вид

$$\frac{\partial f}{\partial x_n} + \lambda \frac{\partial g}{\partial x_n} + v_1 \frac{\partial h_1}{\partial x_n} + v_2 \frac{\partial h_2}{\partial x_n} = 0, \quad n = 1, 2$$

$$\text{откуда} \quad \left. \begin{aligned} 2x_1 + \lambda + v_1(-1) + v_2(2x_1) &= 0 \\ -1 + \lambda + v_1(0) + v_2(2x_2) &= 0 \end{aligned} \right\}$$

$$\text{или } \left. \begin{aligned} 2x_1 + \lambda + v_1 + 2v_2x_1 &= 0 \\ -1 + \lambda + 2v_1x_2 &= 0 \end{aligned} \right\}$$

Четвертые условия Куна-Таккера $v_j h_j(x) = 0, \quad j=1,2,\dots,K$, известные как условия дополняющей нежесткости, принимают вид $v_1(1-x_1)=0; \quad v_2(x_1^2+x_2^2-26)=0$. Заметим, что на переменные v_1 и v_2 накладывается требование не отрицательности. Окончательно условия Куна-Таккера записываются в следующем виде:

$$\begin{aligned} 2x_1 + \lambda - v_1 + 2v_2x_1 &= 0 & v(x_1^2 + x_2^2 - 26) &= 0 \\ \lambda - 1 + 2v_2x_2 &= 0 & 1 - x_1 &\leq 0 \\ x_1 + x_2 - 6 &= 0 & x_1^2 + x_2^2 - 26 &\leq 0 \\ v_1(1 - x_1) &= 0 & v_1 &\geq 0 \\ & & v_2 &\geq 0 \end{aligned}$$

Методы исключения интервалов

Задачу одномерной оптимизации можно поставить следующим образом:

Минимизировать значение целевой функции $f(x)$ на интервале изменения независимой переменной $a \leq x \leq b$, если известно, что функция обладает свойством унимодальности.

Интервал значений X , в котором заключено оптимальное значение X , будем называть интервалом неопределенности. В начале процесса оптимизации этот интервал имеет длину $L_0 = b - a$.

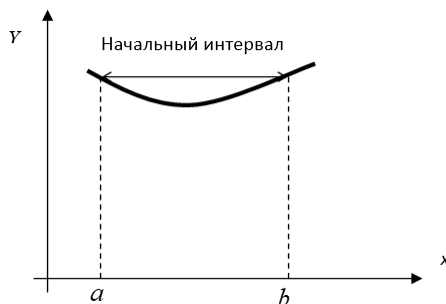


Рис. 3.11. Начальный интервал

Теорема (правило исключения интервалов)

Пусть функция $f(x)$ унимодальна на замкнутом интервале $a \leq x \leq b$, а ее минимум достигается в точке x^* . Рассмотрим точки x_1 и x_2 , расположенные на интервале таким образом, что $a < x_1 < x_2 < b$. Сравнивая значения функции в точках x_1 и x_2 можно сделать выводы:

1. Если $f(x_1) > f(x_2)$, то точка минимума не лежит в интервале (a, x_1) .
2. Если $f(x_1) < f(x_2)$, то точка минимума не лежит в интервале (x_2, b) .

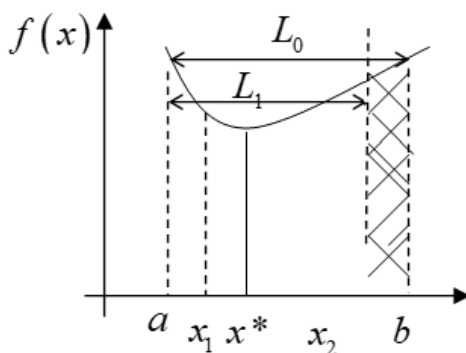


Рис. 3.12. Исключения интервалов

Согласно теореме можно реализовать процедуру поиска точки оптимума путем последовательного сужения интервала неопределенности.

L_0 — исходный интервал неопределенности,

L_i — суженный интервал неопределенности.

Поиск завершается, когда оставшийся интервал уменьшается до достаточно малых размеров $L_i < \varepsilon$, где ε — заданная точность вычисления оптимума.

Существует несколько способов систематического сужения интервала неопределенности.

Метод равномерного поиска

Очевидно, наиболее простым способом сужения интервала неопределенности для одномерной унимодальной функции является деление его на несколько равных частей с последующим вычислением значений целевой функции в узлах полученной сетки. При наличии N точек начальный интервал неопределенности L_0 равен $N + 1$ шагу сетки.

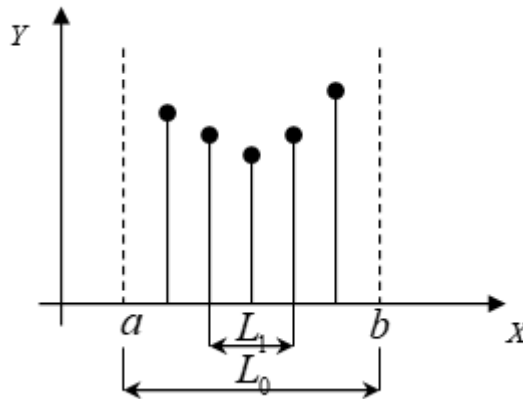


Рис. 3.13. Метод равномерного поиска

В результате применения правил исключения интервалов интервал неопределенности сужается до двух шагов сетки. В качестве оценки эффективности того или иного метода исключения интервалов используют коэффициент относительного уменьшения интервала неопределенности

$$FR(N) = L_N / L_0,$$

где L_0 — исходный интервал неопределенности,

L_N — интервал неопределенности после N вычислений значений функции.

Для метода равномерного поиска

$$FR(N) = \frac{2}{(N+1)}.$$

Чтобы получить значение $FR = 0.01$ потребуется вычислить функцию в 199 точках. Ясно, что эффективность метода очень мала. Ее можно повысить,

избрав, например такой путь: чтобы получить $FR = 0.01$ вычислить сначала функцию в 19 точках и получить $FR = 0.1$, а затем вычислить еще 19 точек на сокращенном интервале получив $FR = 0.01$ всего за 38, а не за 199 вычислений функции.

Метод деления интервала пополам

Рассматриваемый метод позволяет исключать в точности половину интервала на каждой итерации. Метод основан на выборе трех пробных точек, равномерно распределенных в интервале поиска.

Точки x_1, x_m, x_2 делят интервал на четыре равные части. В силу унимодальности функции $f(x)$ возможны три ситуации:

1. $f(x_1) < f(x_m),$

тогда согласно правилу исключения интервалов точка оптимума не лежит в интервале (x_m, b) и, следовательно, этот интервал подлежит исключению.

2. $f(x_2) < f(x_m),$ тогда по правилу исключения интервалов можно исключить из рассмотрения интервал $(a, x_m).$

3. $f(x_1) \geq f(x_m)$ и $f(x_2) \geq f(x_m).$

Дважды применив правило, приходим к выводу, что можно исключить интервалы (a, x_1) и $(x_2, b).$

Другие ситуации невозможны в силу унимодальности функции $f(x).$

Средняя точка последовательно получаемых интервалов всегда совпадает с одной из пробных точек x_1, x_2 или x_m , найденных на предыдущей итерации. Следовательно, на каждой итерации требуется не более двух вычислений значений функции. Таким образом, если проведено N вычислений значений функции, то длина полученного интервала составляет $(1/2)^{(N-1)/2}$ длины исходного интервала, т.е. $FR(N) = 0.5^{(N-1)/2}.$

Метод золотого сечения

В методе золотого сечения функция вычисляется в точках интервала неопределенности расположенных таким образом, чтобы каждое вычисленное значение целевой функции давало полезную информацию.

Сущность метода состоит в следующем. Интервал неопределенности делится на две неравные части таким образом, что отношение большого отрезка к длине всего интервала равно отношению длины меньшего отрезка к длине большого отрезка.

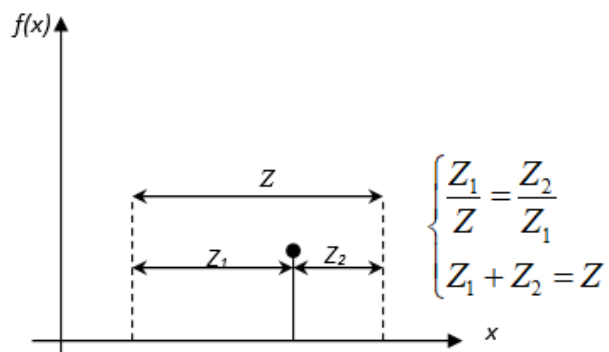


Рис. 3.14. Золотое сечение

Выполним преобразования, разделим второе уравнение на Z

$$\frac{Z_1}{Z} + \frac{Z_2}{Z} - 1 = 0;$$

Учтем из первого уравнения, что $Z = \frac{Z_1^2}{Z_2}$ и после подстановки получим

$$\left(\frac{Z_2}{Z_1} \right)^2 + \frac{Z_2}{Z_1} - 1 = 0$$

Решая это квадратное уравнение, находим для положительного корня

$$\frac{Z_2}{Z_1} = \frac{-1 + \sqrt{5}}{2} \approx 0.618$$

Вторая точка выбирается расположенной в интервале неопределенности симметрично относительно первой.

$$\frac{b - x_2}{x_2 - a} = \frac{x_1 - a}{b - x_1} = 0.618$$

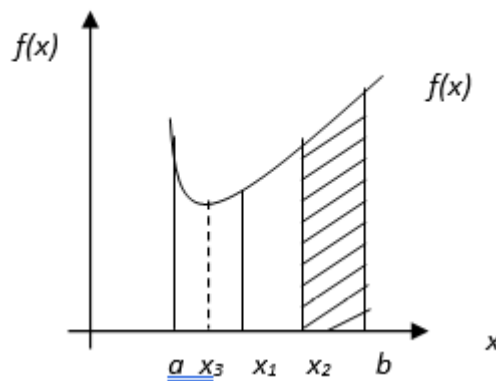


Рис. 3.15. Деление интервала

При делении интервала в этом отношении, два вычисления целевой функции позволяют уменьшить интервал неопределенности в $1/0.618$ раза.

Отметим что: $\frac{x_2 - x_1}{x_1 - a} = 0.618$,

т.е. точка x_1 является точкой золотого сечения для нового интервала неопределенности. Поэтому, чтобы уменьшить интервал неопределенности еще в $1/0.618$ раз потребуется дополнительно вычислить только одно значение целевой функции в точке x_3 , определенной правилом золотого сечения:

$$\frac{x_3 - a}{x_2 - x_3} = 0.618$$

После вычисления N значений целевой функции коэффициент дробления интервала неопределенности составит

$$FR(N) = 0.618^{N-1}.$$

Сравнение методов исключения интервалов

Для сравнения методов равномерного поиска, деления интервала пополам, золотого сечения вспомним коэффициенты относительного уменьшения интервала неопределенности.

Равномерный поиск: $FR(N) = 2/(N + 1)$;

Деление интервала пополам: $FR(N) = (0.5)^{(N-1)/2}$;

Золотое сечение: $FR(N) = 0.618^{N-1}$.

Таблица 3.1

Величины относительного уменьшения интервала

Методы решения	Количество вычислений функции				
	N=2	N=5	N=10	N=15	N=20
Деление интервала пополам	0.5	0.177	0.031	0.006	0.0009
Золотого сечения	0.618	0.146	0.013	0.001	0.0001
Равномерного поиска	0.667	0.333	0.182	0.125	0.095

Из таблицы следует, что метод золотого сечения обеспечивает наибольшее относительное уменьшение интервала при одном и том же количестве вычислений значений функции. С другой стороны, можно сравнить количества вычислений значений функции, необходимые для достижения заданной точности вычислений.

N вычисляется по следующим формулам:

для равномерного поиска $N = (2/E) - 1$

для деления интервала пополам $N = 2\ln(E)/\ln(0.5) + 1$

для золотого сечения

$$N = 1 + 2\ln(E)/\ln(0.618)$$

Таблица 3.2

Требуемое количество вычислений значений функций

Метод поиска	Заданная точность			
	$E = 0.1$	$E = 0.05$	$E = 0.01$	$E = 0.001$
Деление интервала пополам	7	9	14	20
Золотого сечения	6	8	11	16
Равномерный поиск	19	39	199	1999

Полиномиальная аппроксимация

Применение методов исключения интервалов накладывает единственное ограничение на исследуемую функцию: она должна быть *унимодальной*. Следовательно, указанные методы можно применять для анализа и непрерывных, и разрывных, и дискретных функций. Логическая структура поиска с помощью методов исключения интервалов основана на простом сравнении значений функции в двух точках. Методы аппроксимации позволяют учесть относительные изменения в значении функции, поэтому оказываются более эффективными. Однако эффективность достигается ценой дополнительного ограничения на форму исследуемых функций: они должны быть *унимодальны* и *непрерывны*.

Согласно теореме Вейерштрасса, если функция непрерывна в некотором интервале, то её с любой степенью точности можно аппроксимировать полиномом достаточно большого порядка.

Следовательно, если функция унимодальна и найден полином, который достаточно точно её аппроксимирует, то координату точки оптимума функции можно оценить путем вычисления координаты точки оптимума полинома.

Качество оценок можно повышать двумя способами:

- использованием полинома более высокого порядка;
- уменьшением интервала аппроксимации;

Второй способ, вообще говоря, является более предпочтительным.

Квадратичная интерполяция

Если задана последовательность точек x_1, x_2, x_3 и известны соответствующие им значения функции f_1, f_2, f_3 то функция $f(x)$ может быть аппроксимирована квадратичной функцией

$$\phi(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$$

Определим постоянные коэффициенты a_0, a_1, a_2 исходя из условия, что значения функции $\phi(x)$ совпадают со значениями $f(x)$ в контрольных точках.

Для $x = x_1$ имеем $f_1 = f(x_1) = \phi(x_1) = a_0$, т.е. $a_0 = f_1$.

Для $x = x_2$ имеем $f_2 = f(x_2) = \phi(x_2) = f_1 + a_1(x_2 - x_1)$, откуда $a_1 = \frac{f_2 - f_1}{x_2 - x_1}$.

Для $x = x_3$:

$$f_3 = f(x_3) = \phi(x_3) = f_1 + \frac{(f_2 - f_1)(x_3 - x_1)}{(x_2 - x_1)} + a_2(x_3 - x_1)(x_3 - x_2),$$

$$\text{откуда } a_2 = \frac{1}{x_3 - x_1} \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right)$$

Используем полученный квадратичный полином для оценивания координаты точки оптимума.

$$\frac{dy}{dx} = a_1 + a_2(x - x_1) + a_2(x - x_2) = 0$$

Отсюда, точка минимума функции аппроксимируется значением:

$$x^* = \frac{x_1 + x_2}{2} - \frac{a_1}{2a_2}.$$

Метод, разработанный Пауэллом, основан на последовательном применении процедуры оценивания с использованием квадратичной аппроксимации. Схему алгоритма можно описать следующим образом:

Пусть x_1 - начальная точка, Δx - выбранная величина шага по оси X .

1. Вычислить $x_2 = x_1 + \Delta x$
2. Вычислить $f(x_1)$ и $f(x_2)$
3. Если $f(x_1) > f(x_2)$ положить $x_3 = x_1 + 2\Delta x$.

Если $f(x_1) \leq f(x_2)$ положить $x_3 = x_1 - \Delta x$.

4. Вычислить $f(x_3)$ и найти

$$F_{\min} = \min(f_1, f_2, f_3),$$

$$x_{\min} = \text{точка } x_i, \text{ которая соответствует } F_{\min}.$$

5. По трем точкам x_1, x_2, x_3 вычислить x^* , используя формулу квадратичной аппроксимации.

6. Проверка на окончание поиска:

а) является ли разность $(F_{\min} - f(x^*))$ достаточно малой?

б) является ли разность $(x_{\min} - x^*)$ достаточно малой?

Если оба условия выполняются, закончить поиск. В противном случае перейти к п.7.

7. Выбрать наилучшую точку $x_{\min}$ или x^* и две точки по обе стороны от неё, или 3 точки, в которых функция имеет минимальные значения. Перейти к п.5.

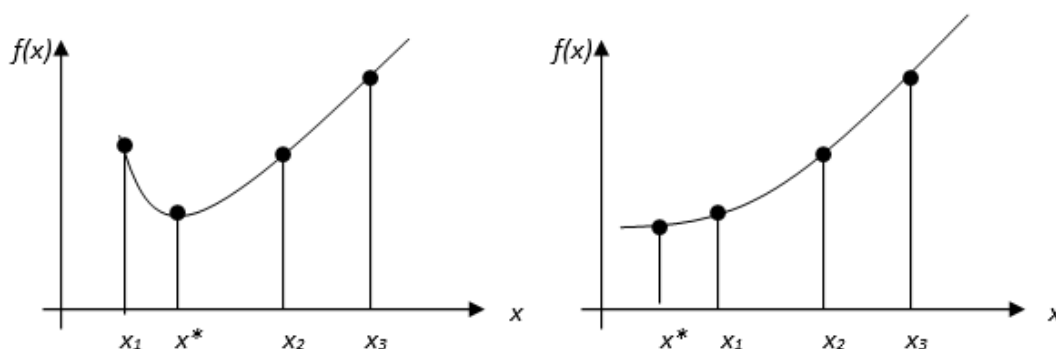


Рис. 3.16. Квадратичная интерполяция

Методы с использованием производных

Использование в методах поиска предположения о непрерывности функции в дополнение к предположению об унимодальности позволило получить более эффективный алгоритм квадратичной интерполяции. Целесообразно предположить, что если в дополнение к условию непрерывности ввести требование *дифференцируемости* функции, то эффективность поисковых процедур можно ещё более повысить.

Метод средней точки

Если функция $f(x)$ унимодальна в заданном интервале поиска, то точкой оптимума является точка, в которой $f'(x) = 0$.

Если имеется, возможность вычислить как значение функции, так и её производной, то для нахождения корня уравнения $f'(x) = 0$ можно использовать следующий эффективный алгоритм исключения интервалов, на каждой итерации которого рассматривается лишь одна пробная точка.

Определим 2 точки L и R таким образом, что $f'(L) < 0$ и $f'(R) > 0$.
 $f'(x) > 0$

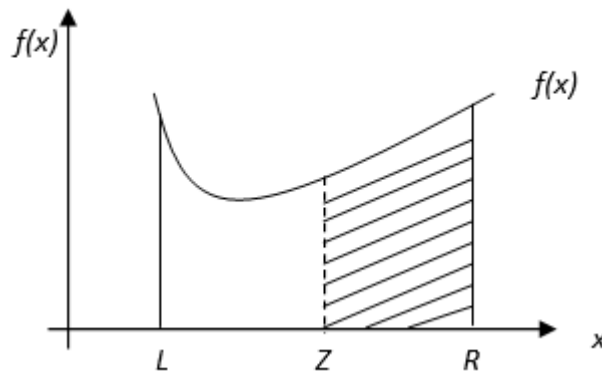


Рис. 3.17. Метод средней точки

Стационарная точка расположена между L и R . Вычислим значение производной в средней точке рассматриваемого интервала $Z = (L + R) / 2$.

Если $f'(Z) > 0$, то интервал (Z, R) можно исключить из интервала поиска.

Если $f'(Z) < 0$, то интервал (L, Z) можно исключить из интервала неопределенности.

Коэффициент относительного уменьшения интервала неопределенности:

$$FR = 0.5^{N-2}$$

Кубическая интерполяция

В рассматриваемом методе исследуемая функция аппроксимируется кубическим полиномом. Для кубической интерполяции используются значение функции и её производной в двух точках. Работа алгоритма начинается в произвольно выбранной точке x_1 . Другая точка x_2 находится из условия, что производные $f'(x_1)$ и $f'(x_2)$ имеют различные знаки.

Аппроксимирующую кубическую функцию запишем в виде:

$$\phi(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + a_3(x - x_1)^2(x - x_2)$$

Параметры аппроксимирующего полинома подбираются таким образом, чтобы значения функции, и её производной в точках x_1 и x_2 совпадали со значениями полинома в этих точках.

Первая производная функции $\phi(x)$ равна:

$$\frac{d\phi}{dx} = a_1 + a_2(x - x_1) + a_2(x - x_2) + a_3(x - x_1)^2 + 2a_3(x - x_2)(x - x_1)$$

Коэффициенты полинома определяются по известным значениям $f(x_1)$, $f(x_2)$, $f'(x_1)$, $f'(x_2)$:

$$f_1 = f(x_1) = a_0$$

$$f_2 = f(x_2) = a_0 + a_1(x_2 - x_1)$$

$$f_1' = f'(x_1) = a_1 + a_2(x_1 - x_2)$$

$$f_2' = f'(x_2) = a_1 + a_2(x_2 - x_1) + a_3(x_2 - x_1)^2$$

После решения полученной системы, приравняем нулю первую производную $\phi'(x) = 0$ и найдем корни квадратного уравнения. Получим решение в следующем виде:

$$x^* = \begin{cases} x_2 & \text{если } \mu < 0 \\ x_2 - \mu(x_2 - x_1) & \text{если } 0 \leq \mu \leq 1 \\ x_1 & \text{если } \mu > 1 \end{cases}$$

где

$$\mu = \frac{f_2' + \omega - z}{f_2' - f_1' + 2\omega}$$

$$z = \left(\frac{3(f_1 - f_2)}{x_2 - x_1} \right) + f_1' + f_2'$$

$$\omega = \begin{cases} \left(z^2 - f_1' f_2' \right)^{\frac{1}{2}} & , \text{если } x_1 < x_2 \\ - \left(z^2 - f_1' f_2' \right)^{\frac{1}{2}} & , \text{если } x_1 > x_2 \end{cases}$$

Формула для ω обеспечивает надлежащий выбор одного из двух корней квадратного уравнения. Формула гарантирует, что получаемая точка расположена в интервале (x_1, x_2) .

Приведем описание алгоритма.

Пусть заданы начальная точка x_0 , положительная величина шага Δ и параметры сходимости ε_1 и ε_2 .

1. Вычислить $f'(x_0)$

Если $f'(x_0) < 0$, вычислить $x_{k+1} = x_k + 2^k \Delta$ для значений $k = 0, 1, \dots$

Если $f'(x_0) > 0$, вычислить $x_{k+1} = x_k - 2^k \Delta$ для значений $k = 0, 1, \dots$

2. Повторить вычисления x_k до точки M , в которой меняется знак производной, т.е. $f'(x_{M-1}) f'(x_M) < 0$. Положить $x_1 = x_{M-1}$; $x_2 = x_M$. Вычислить f_1, f_2, f'_1, f'_2 .

3. Найти точку x^* оптимума кубического полинома.

4. Проверка на окончание поиска:

Если $|f'(x^*)| < \varepsilon_1$ и $|(x^* - x_1)| < \varepsilon_2$ поиск закончить

Иначе а) $x_2 = x_1$ и $x_1 = x^*$, если $f'(x^*) f'(x_1) < 0$

б) $x_1 = x_2$ и $x_2 = x^*$, если $f'(x^*) f'(x_2) < 0$

Перейти к шагу 3.

Сравнение методов

С помощью теоретических выкладок доказано, что методы квадратичной и кубической интерполяции существенно эффективнее методов исключения интервалов.

В случае, когда значения производной вычисляются непосредственно, метод поиска с использованием кубической аппроксимации оказывается наиболее эффективным. Однако, если значения производной вычисляются путём численного дифференцирования, то предпочтение следует отдать квадратичной интерполяции, которая вычислений производной не требует.

Если необходимо получить решение с высокой степенью точности, то лучшим оказываются методы поиска на основе полиномиальной аппроксимации.

С другой стороны, при исследовании мультимодальных или быстро изменяющихся функций методы исключения интервалов сходятся быстрее. Если важно добиться надежной работы алгоритма, то целесообразно выбрать метод золотого сечения.

Рекомендуется использовать поисковые методы полиномиальной аппроксимации совместно с методом золотого сечения.

Метод Пауэлла, метод средней точки и метод золотого сечения использовались для минимизации функции $f(x) = \sin^k x$, $k = 1..79$.

Для всех k функция имеет минимум в точке $x^k = 4.71$ при $f(x^*) = -1$. Однако с увеличением k гладкость функции, которая обладает узкими впадинами в окрестности точки минимума, уменьшается. С ростом k увеличились затраты времени, особенно при реализации метода средней точки, вследствие резкого увеличения градиента функции в окрестности точки минимума. Однако возрастание k не оказывало заметного влияния на продолжительность счета методом золотого сечения.

Точность решения падает с ростом k для метода средней точки и метода Пауэлла. Метод же золотого сечения оказывается нечувствительным к изменениям крутизны функции.

Поиск методом Фибоначчи

Задача: определить минимум функции $f(x_0)$ на интервале (a, b) $a \leq x \leq b$ как можно точнее, используя только n вычислений функции. Начальный интервал неопределенности $L_1 = b - a$. Для сужения интервала требуется знать значение функции как минимум в двух различных точках этого интервала. Как их разместить наилучшим образом внутри интервала?

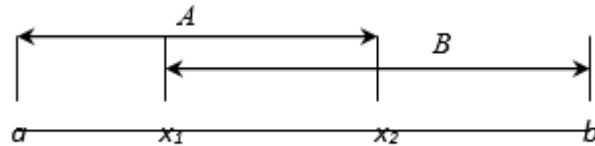


Рис. 3.18. Деление методом Фибоначчи

Поскольку не известно, какая из ситуаций будет иметь место, целесообразно разместить точки x_1, x_2 симметрично относительно интервала. Если можно выполнить еще одно вычисление функции, то нужно поместить следующую точку внутри интервала неопределенности симметрично уже имеющейся там точке.

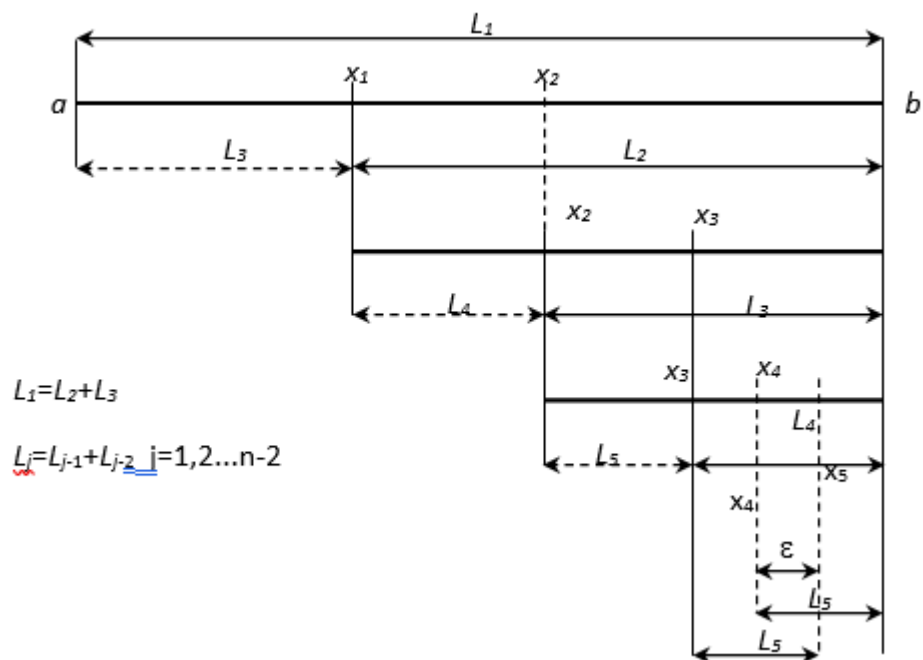


Рис. 3.19. Алгоритм метода Фибоначчи

Для того, чтобы получить наибольшее уменьшение интервала на последнем ($n - m$ шаге) следует разместить точки x_{n-1}, x_n как можно ближе к середине интервала. Тогда мы сможем получить коэффициент уменьшения интервала близкий к $1/2$. Обозначим расстояние между точками x_n и x_{n-1} через ε , тогда интервалы неопределенности L_n и L_{n-1} будут связаны соотношением (см. рис.).

$$L_{n-1} = 2L_n - \varepsilon$$

С учетом общей формулы имеем:

$$L_{n-2} = L_{n-1} + L_n = 3L_n - \varepsilon$$

$$L_{n-3} = L_{n-2} + L_{n-1} = 5L_n - 2\varepsilon$$

$$L_{n-4} = L_{n-3} + L_{n-2} = 8L_n - 3\varepsilon$$

.....

Последовательность чисел Фибоначчи определяется следующим образом

$$F_0 = 1, F_1 = 1, F_k = F_{k-1} + F_{k-2}, \quad k = 2, 3, \dots$$

Очевидно, коэффициенты полученных соотношений для интервалов неопределенности являются числами Фибоначчи и поэтому их можно записать в виде

$$L_{n-j} = F_{j+1}L_n - F_j\varepsilon, \quad j = 2 \dots n-1.$$

Для начального интервала $L_1 = (b-a)$ имеем $j=n-1; \rightarrow L_1 = F_nL_n - F_{n-1}\varepsilon$.

$$\text{Отсюда } L_n = \frac{L_1}{F_n} + \frac{F_{n-1}}{F_n} \varepsilon.$$

Следовательно, после n вычислений функции коэффициент относительного уменьшения интервала неопределённости: $FR = \frac{1}{F_n}$; где F_n - число Фибоначчи.

Мы до сих пор не выяснили: как разместить левую точку x_1 в начальном интервале неопределенности L_1 ? Для ответа на этот вопрос надо определить L_2

$$j = n - 2;$$

$$L_2 = F_{n-1}L_n - \varepsilon F_{n-3} = F_{n-1} \left(\frac{L_1}{F_n} + \varepsilon \frac{F_{n-2}}{F_n} \right) - \varepsilon F_{n-3} =$$

$$\frac{F_{n-1}}{F_n} L_1 + \varepsilon \left(\frac{F_{n-1}F_{n-2} - F_{n-3}F_n}{F_n} \right) = \frac{F_{n-1}}{F_n} L_1 + \frac{(-1)^n \varepsilon}{F_n};$$

Числа Фибоначчи используются только для нахождения первой точки. Значения ε определяются из практических соображений, но должно быть $\varepsilon < \frac{L_1}{F_{n+1}}$;

Действительно, возьмем последний шаг.

Чтобы произошло уменьшение интервала неопределенности необходимо, чтобы $L_n > 2\varepsilon$.

С другой стороны, $L_n = \frac{L_1}{F_n} + \varepsilon \frac{F_{n-2}}{F_n}$. Совместное решение уравнения и неравенства приводят к указанному результату.

Если невозможно обеспечить выполнение неравенства по условиям вычисления функции (2ε -минимальное различимое расстояние между точками) то следует уменьшить n , иначе мы будем понапрасну тратить время на последних итерациях.

Связь метода Фибоначчи с методом золотого сечения

Для золотого сечения также справедливо соотношение (из-за симметрии расположения точек):

$$L_j = L_{j-1} + L_{j-2}, \quad j=1, 2 \dots n-2; \text{но неизвестно } n \text{ и поэтому нельзя записать}$$

$$L_{n-1} = 2L_n - \varepsilon.$$

Вместо этого предположения, используем другое: относительное уменьшение интервала на каждом шаге должно быть одинаковым, т.е. $\frac{L_2}{L_1} = \frac{L_3}{L_2}$ т.е. получили отношение золотого сечения.

Численные методы поиска экстремума в нелинейных задачах математического программирования

Методы прямого поиска в задачах условной оптимизации

В методах прямого поиска для отыскания точек оптимума используется только целевая функция и ограничения. Как и в задачах безусловной оптимизации, необходимость разработки этих методов связана с тем, что в технических приложениях часто приходится сталкиваться с задачами, в которых функции разрывны и недифференцируемы. Поскольку эти методы строятся на интуитивных соображениях, не подкрепленных строгой теорией, то их сходимость к оптимальному решению не гарантируется. В то же время, они позволяют получать достаточно хорошие (но не обязательно оптимальные) решения. Кроме того, в силу своей логической простоты эти методы легко реализуемы.

Метод комплексов

Метод комплексов представляет собой модификацию метода поиска по симплексу применительно к задачам условной оптимизации. Изменение коснулось двух основных положений:

При реализации метода прямого поиска по симплексу симплекс преобразуется путем проектирования «худшей» точки вдоль прямой проходящей через центр тяжести остальных точек симплекса. В задачах с ограничениями новая точка может оказаться недопустимой.

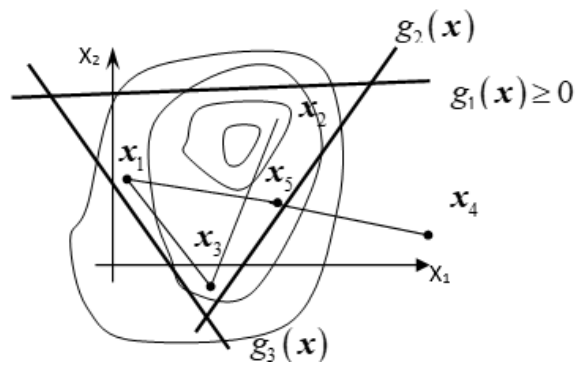


Рис. 3.20. Симплекс

Тогда ее можно сдвинуть к центру тяжести, пока она не станет допустимой.

x_4 -недопустимая точка, полученная в результате преобразований симплекса

x_5 -новая точка, расположенная внутри допустимой области, получена перемещением точки x_4 .

1. При наличии ограничений затруднительно построение точек исходного симплекса, поскольку велика вероятность попадания их в недопустимую область. Автор метода Бокс предложил строить множество пробных точек P случайным образом. Если заданы верхние и нижние границы изменения каждой переменной $x_i'' \leq x_i \leq x_i^g$, $i = 1, 2, \dots, N$ и последовательность псевдослучайных чисел A , равномерно распределенных на отрезке $[0, 1]$, то координаты точки определяются с помощью формулы $x_i = x_i'' + a_i(x_i^g - x_i'')$, $a_i \in A$, $i = 1, 2, \dots, N$. Таким образом, для получения точки в N -мерном пространстве требуется N случайных чисел. Каждая полученная точка проверяется на допустимость и если какое-нибудь из ограничений не выполняется, то она сдвигается к центру тяжести уже построенных точек до тех пор, пока не получится допустимая точка. Общее число точек P должно быть не меньше $N + 1$, но может быть и больше. После того как множество исходных P точек построено его преобразование осуществляется следующим образом. Определяется точка с наихудшим значением целевой функции и отбрасывается. Новая точка получается путем отражения исключаемой точки через центр тяжести

остальных точек. Если x^0 -исключаемая точка, x^c -центр тяжести остальных точек, то новая точка определяется как $x^m = x^c + \alpha(x^c - x^0)$. Параметр α задает расстояние отражения: при $\alpha = 1$ имеет место равенство $\|x^m - x^c\| = \|x^c - x^0\|$; $\alpha > 1$ соответствует растяжению; $\alpha < 1$ соответствует сжатию. $x^c = \frac{1}{k} \sum_{i=1}^k x^i$; где $k = P - 1$

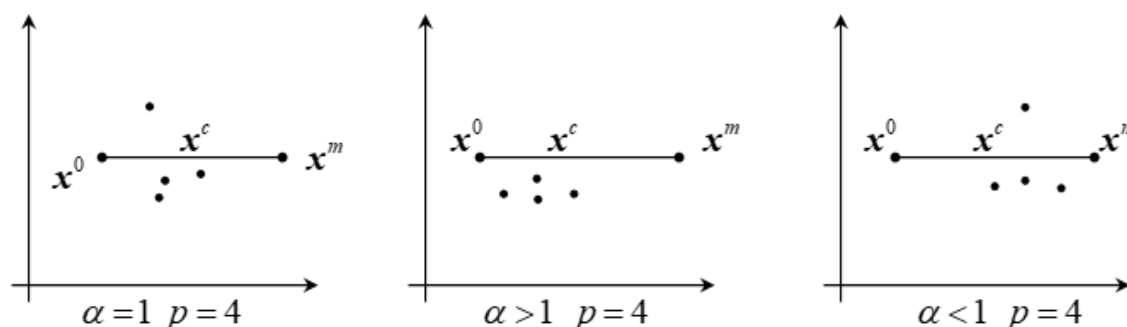


Рис. 3.21. Параметр α

После замены «худшей» точки x^0 на новую x^m возможны следующие ситуации:

1. Новая точка допустимая и значение целевой функции в ней лучше, чем в точке x^0 . В этом случае находим новую «худшую» точку и осуществляем операцию отражения.
2. Новая точка допустимая, но значение целевой функции в ней не лучше, чем в точке x^0 . Передвинем эту точку на половину расстояния до ранее найденного центра тяжести (иначе возможно заикливание).

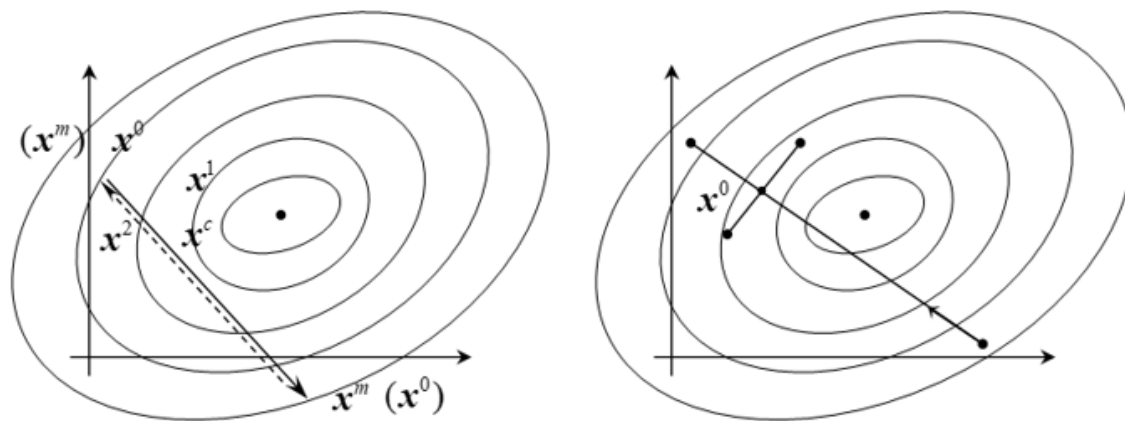


Рис. 3.22. Движение точки

3. Если $f(x^m) \geq f(x^0)$ то новый шаг будет выполнен в противоположном направлении и может привести к дальнейшему ухудшению значения функции

$$x^{m2} = \frac{1}{2}(x^m + x^c)$$

4. Новая точка недопустимая. Передвинем точку на половину расстояния до ранее найденного центра тяжести.

Очевидно, что при принятой стратегии поиска по мере приближения к оптимуму точки многогранника будут стягиваться к центру тяжести. Поэтому условием окончания процедуры будет достижение многогранником заданных малых размеров или получение достаточно малой разницы между значениями целевой функции в вершинах симплекса. Бокс провел численные эксперименты по описанному алгоритму и на основе их анализа рекомендует выбирать $\alpha = 1,3$ и $P \approx 2N$. Бокс также рекомендует, если новая точка выходит за границы допустимых значений переменных, то соответствующую координату принимать равной граничному значению.

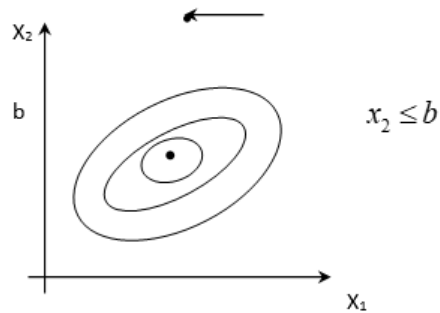
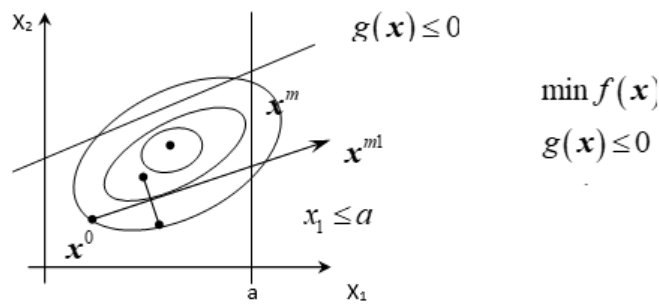


Рис. 3.23. Алгоритм метода комплексов

Выбор α больше единицы компенсирует сжатие комплекса, обусловленное сокращением в два раза расстояния до центра тяжести. Большое число вершин используется для предотвращения «уплощения» и вырождения комплекса, когда поиск ведется вблизи границы допустимой области.

Методы случайного поиска

Выборки с уменьшением интервала

В методе используются серии случайных выборок. Наилучшая точка каждой серии используется как начальная точка следующей серии. Точки каждой последующей серии выбираются из интервала меньшей величины.

Задается Q - количество серий; P - количество опытов в серии;

x^0 - начальное допустимое решение;

z^0 - оценка величины начального интервала;

$$\left(x^0 - \frac{1}{2} z^0 \right) \leq x \leq \left(x^0 + \frac{1}{2} z^0 \right).$$

ε - параметр уменьшения интервала $0 < \varepsilon < 1$.

Алгоритм:

Шаг 1. Для $i = 1, \dots, N$ вычислить $x_i^p = x_i^{q-1} + rz_i^{q-1}$, где r -случайная величина, равномерно распределенная на интервале $\left(-\frac{1}{2}; \frac{1}{2}\right)$.

Шаг 2. Если $\mathbf{x}^p$ - недопустимая точка и $p < P$, повторить шаг 1. Если $\mathbf{x}^p$ - допустимая точка, запомнить $\mathbf{x}^p$ и $f(\mathbf{x}^p)$. Если $p = P$, найти среди всех допустимых точек $\mathbf{x}^p$ точку с наименьшим значением $f(\mathbf{x}^p)$ и положить ее равной $\mathbf{x}^q$.

Шаг 3. Уменьшить интервал, полагая $z_i^q = (1 - \varepsilon)z_i^{q-1}$.

Шаг 4. Если $q > Q$, закончить вычисления. В противном случае, увеличить q и продолжить вычисления, начиная с шага 1.

Если положить $\varepsilon = 0,05$, количество опытов в серии $P = 100$; число серий $Q = 200$. Начальный единичный интервал сводится к интервалу $(1 - 0,05)^{200} \approx 3,5 \cdot 10^{-5}$. Для этого требуется примерно $2 \cdot 10^4$ испытаний.

Таким образом, для простой задачи небольшой размерности, в которой каждая проверка допустимости и подсчет значений целевой функции требует не более 10^{-3} секунды, общие затраты времени можно считать приемлемыми. Для более реальных задач описанный алгоритм может оказаться слишком дорогостоящим.

Адаптивный алгоритм случайного поиска с переменным шагом

Для повышения эффективности методов случайного поиска предлагаются эвристические алгоритмы, которые включают элементы методов спуска в общую схему поиска. В данном алгоритме случайные выборки используются для определения направления поиска. Длина шага изменяется в зависимости от результата поиска: если две последовательные итерации дают улучшение

целевой функции, величина шага увеличивается в α_s раз. Если же M последовательных итераций не дают улучшения, то величина шага уменьшается в α_f раз.

Алгоритм.

Даны параметры α_s, α_f, M и начальная точка x^0 . Для определения условия окончания вычислений вводится минимальное значение шага $\alpha_{\min}$.
Рекомендуется выбрать $\alpha_s = 1,618, \alpha_f = 0,618, M = 3N$.

Уровень 0.

Ввод $(\alpha_s, \alpha_f, M, x^0, \alpha_{\min})$

$\alpha := 1; \{ \alpha - \text{величина шага} \}$

$m := 0; \{ m - \text{количество неудачных точек выбранных подряд} \}$

repeat

(а) Найти случайным образом точку $x^{(1)}$ для которой $f(x^{(1)}) < f(x^0)$. l_1

случайная точка найдена.

Комбинаторный эвристический алгоритм

Алгоритм первоначально был разработан для решения задач проектирования машин и механизмов. Основная идея алгоритма состоит в разбиении интервалов значений каждой независимой переменной на участки дискретных значений, и реализации случайного поиска на полученной дискретной решетке.

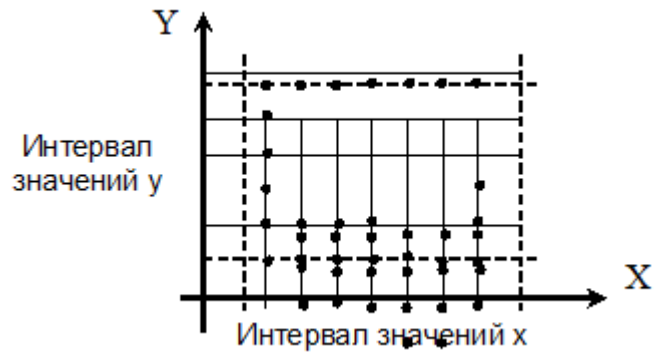


Рис. 3.24. Комбинаторный эвристический алгоритм

Непрерывная область значений переменных заменяется на узлы дискретной решетки.

Упреждающий поиск

- (а) Для каждого из Q допустимых решений, найденных для переменной i . Выбрать q допустимых значений переменной $(i + 1)$.
- (б) Выбрать наилучшую из $Q \times Q$ допустимых точек. Зафиксировать значение переменной i , соответствующее этой точке, как оптимальное.

Пояснения.

- а) Авторы алгоритма утверждают, что Q -т.е. число упреждающих точек, должно лежать между 3 и 5.
- б) Во всех случаях следует ограничивать количество попыток при поиске Q точек дающих улучшение целевой функции. Если Q точек получить не удастся надо переходить к следующей переменной.

Обсуждение методов прямого поиска

Подготовка задачи к решению

Общая задача условной оптимизации содержит ограничения в виде равенств, неравенств, а также верхние и нижние границы значений переменных:

$$\begin{aligned} \text{Минимизировать} \quad & f(\mathbf{x}), \mathbf{x} = (x_1, x_2, \dots, x_N) \\ & g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, 3, \dots, J, \\ \text{При ограничениях} \quad & h_k(\mathbf{x}) = 0, \quad k = 1, 2, 3, \dots, K, \\ & x_i^H \leq x_i \leq x_i^L, \quad i = 1, 2, \dots, N \end{aligned}$$

Существенная проблема заключается в том, что алгоритмы методов прямого поиска работают только с допустимыми точками пространства переменных. Как мы убедились при рассмотрении методов прямого поиска, если задача содержит ограничения только в виде неравенств, то существуют эвристические приемы, позволяющие перевести пробную точку $\mathcal{A}$ из недопустимой области в допустимую. Если же $\mathcal{A}$ недопустимая точка, то оперируя лишь значениями функций, входящих в ограничения, трудно изменить координаты точки так, чтобы выполнилось ограничение равенства.

Трудности получения допустимой точки в задачах с ограничениями в виде равенств приводят к тому, что все методы прямого поиска требуют, чтобы ограничения были заданы только в виде неравенств. Следовательно, ограничения в виде равенств должны быть явно или неявно исключены перед решением задачи.

Метод комплексов

Хотя метод комплексов не требует непрерывности функции, так как не используем значений производных, строго говоря, необходимо чтобы допускаемая область была выпуклой. Это условие оказывается существенным при нахождении центра тяжести, так как предполагается, что центр тяжести допускаемых точек также будет допускаемой точкой. Это может оказаться неверным, если допускаемая область невыпуклая.



Рис. 3.25. Допустимая область

Аналогичная ситуация может возникнуть при сдвиге допустимой, но неудовлетворительной по значению целевой функции точки. Сдвинутая точка может оказаться недопустимой даже если центр тяжести - недопустимая точка.

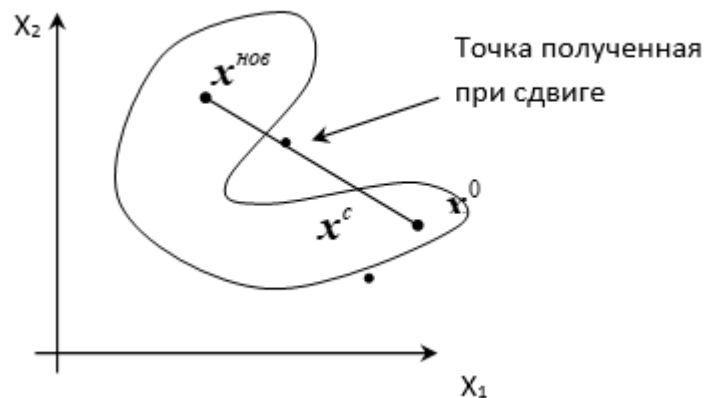


Рис. 3.26. Сдвиг

Следовательно, для невыпуклых областей метод комплексов расходится. Сходимость метода комплексов так же существенно замедляется, если пробные точки располагаются вдоль границы допустимой области. Поэтому используется специальный прием, при котором вычисления несколько раз прерываются по какому-либо критерию и затем алгоритм выполняется сначала из точки, имеющей наилучшее значение целевой функции.

Хотя метод, вообще говоря, сходится медленно, на практике он широко используется в технических приложениях и дает хорошие результаты для выпуклых задач.

Метод случайного поиска

По имеющимся данным о вычислительных экспериментах, алгоритм случайного поиска целесообразно использовать либо для задач небольшой размерности, либо как вспомогательный прием для определения хорошей

начальной точки. Известно, что любые процедуры случайного поиска, охватывающего большую часть допустимой области, существенно улучшают возможности обнаружения глобального оптимума, в случае многоэкстремальных задач.

Из рассмотренных методов, адаптивные процедуры являются более эффективными, так как они позволяют использовать накопленную информацию.

Раздел 4. Методы оптимизации в машинном обучении

Методы одномерной минимизации. Градиентные методы и метод Ньютона. Оптимизация в пространстве большой размерности: общий метод сопряжённых градиентов и неточный (безгессианный) метод Ньютона

Методы оптимизации лежат в основе решения многих задач компьютерных наук. Например, в машинном обучении задачу оптимизации необходимо решать каждый раз при настройке какой-то модели алгоритмов по данным, причём от эффективности решения соответствующей задачи оптимизации зависит практическая применимость самого метода машинного обучения.

Метод Хука-Дживса

Метод Хука-Дживса является одним из первых методов поиска (1961 год), в котором при определении нового направления учитывается информация, полученная на предыдущих итерациях. Поиск состоит из последовательности шагов исследующего поиска вокруг базисной точки, за которой, в случае успеха, следует поиск по образцу.

Исследующий поиск. Для проведения исследующего поиска выбирается базисная точка $\mathbf{x}_0$, и шаг длиной Δx_j для каждого из направлений $\mathbf{x}_j$.

Каждая переменная по очереди изменяется прибавлением длины шага. Если это приводит к уменьшению значения функции $f(\mathbf{x}_0 + \Delta x_i \mathbf{e}_i)$, где $\mathbf{e}_i$ -

единичный вектор, направленный вдоль оси X_i , то точка X_0 заменяется на точку $(X_0 + \Delta x_i e_i)$. В противном случае вычисляется значение функции в точке $(X_0 - \Delta x_i e_i)$. Когда будут рассмотрены все n координатных осей, мы будем иметь новую базисную точку X_1 .

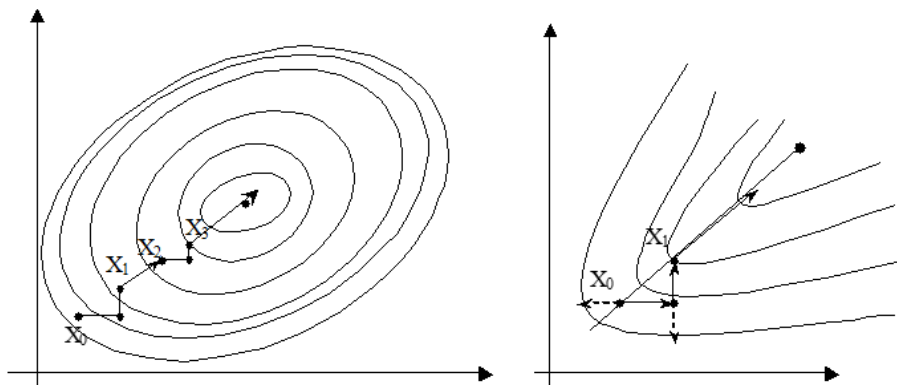


Рис. 4.1. Метод Хука-Дживса

Поиск по образцу заключается в реализации единственного шага из новой базовой точки вдоль прямой, соединяющей эту точку с предыдущей базовой точкой. Новая точка образца определяется в соответствии с формулой $x_{k+1}^p = x_k + 2(x_k - x_{k-1})$.

x_k — текущая базовая точка.

x_{k-1} — предыдущая базовая точка.

x_{k+1} — новая базовая точка.

x_{k+1}^p — точка, построенная при движении по образцу.

Алгоритм поиска:

1. Задать: начальную точку x_0 , приращение $\Delta x_i, i=1, \dots, N$ коэффициент уменьшения шага $\alpha > 1$ параметр окончания поиска ε
2. Провести исследующий поиск в точке x_k .

3. Если найдена точка x_{k+1} с меньшим значением целевой функции перейти к п.5.
4. Проверка на окончание поиска. Если выполняется неравенство $|\Delta x| < \varepsilon$ прекратить поиск, иначе уменьшить приращение по формуле $\Delta x_i = \Delta x_i / \alpha, i = 1, 2 \dots N$. Перейти к п.2.
5. Провести поиск по образцу $x_{k+1}^p = x_k + 2(x_k - x_{k-1})$.
6. Провести исследующий поиск в базовой точке x_{k+1}^p . Пусть x_{k+1} полученная в результате точка.
7. Если выполняется неравенство $f(x_{k+1}) < f(x_k)$, то положить $x_{k-1} = x_k; x_k = x_{k+1}$, т.е. принять точку x_{k+1} в качестве базовой и перейти к шагу 5, иначе перейти к шагу 4.

Метод Хука-Дживса отличается несложной стратегией поиска и невысоким уровнем требований к объему памяти ЭВМ, который даже ниже, чем в случае использования метода поиска по симплексу. Благодаря этому алгоритм находит широкое применение. Однако необходимо отметить, что алгоритм при наличии значительных нелинейных эффектов вырождается в последовательность исследовательских поисков без перехода к ускоряющему поиску по образцу.

Градиентные методы

В предыдущем разделе рассматривались методы, позволяющие получить решение задачи на основе использования только значений целевой функции.

Важность прямых методов несомненна, поскольку в ряде практических инженерных задач информация о значениях целевой функции является единственной надежной информацией, которой располагает исследователь.

С другой стороны, при использовании даже самых эффективных прямых методов для получения решения требуется иногда вычисление слишком большого количества значений целевой функции.

Это обстоятельство приводит к необходимости рассмотрения методов, основанных на использовании градиента целевой функции.

Все они основаны на итерационной процедуре, реализуемой в соответствии с формулой: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{S}(\mathbf{x})$, где $\mathbf{x}_k$ - текущее приближение к решению $\mathbf{x}^*$, α_k - параметр, характеризующий длину шага, $\mathbf{S}(\mathbf{x}_k) = \mathbf{S}_k$ - направление поиска в N -мерном пространстве оптимизируемых параметров.

Метод наискорейшего спуска

Определим в некоторой точке $\mathbf{x}_k$ -пространства переменных направление наискорейшего локального уменьшения целевой функции.

Разложим целевую функцию $f(\mathbf{x}_k)$ в окрестностях точки $\mathbf{x}_k$ в ряд Тейлора: $f(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x})^T \cdot \Delta \mathbf{x} + \dots$ и отбросим члены второго порядка и выше.

Очевидно, что локальное уменьшение целевой функции определяется вторым слагаемым, так как $f(\mathbf{x}_k)$ фиксировано.

В более простом виде скалярное произведение можно записать так: $\nabla f(\mathbf{x})^T \cdot \Delta \mathbf{x} = \|\nabla f(\mathbf{x})\| \cdot \|\Delta \mathbf{x}\| \cdot \cos \theta$, где θ - угол между векторами $\nabla f(\mathbf{x})$ и $\Delta \mathbf{x}$.

Для заданной величины $\|\Delta \mathbf{x}\|$ скалярное произведение будет иметь минимальное отрицательное значение при $\theta = 180^\circ$, т. е. когда направление $\Delta \mathbf{x}$ противоположно направлению градиента.

Поэтому в основе простейшего градиентного метода наискорейшего спуска лежит формула $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$, где α - заданный положительный параметр.

Метод обладает двумя недостатками:

- а) возникает необходимость выбора подходящего значения α ;
- б) методу свойственна медленная сходимость к точке минимума вследствие малости ∇f в окрестности этой точки

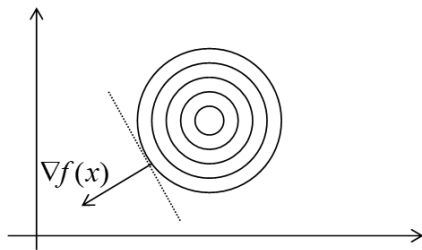


Рис. 4.2. Направление градиента

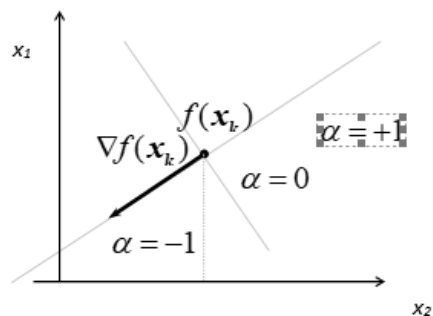


Рис. 4.3. Градиентный спуск

Таким образом, ∇f задает направление, в котором следует двигаться, а шаг перемещения определяется параметром α на каждой итерации $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)$. Значение α_k вычисляется путем решения задачи минимизации $f(\mathbf{x}_{k+1})$ вдоль направления $\nabla f(\mathbf{x}_k)$ с помощью того или иного метода одномерного поиска.

Задача минимизации функции $f(\mathbf{x})$ на прямой $\mathbf{x}_k - \alpha \mathbf{S}_k$, где $\mathbf{S}_k = \nabla f(\mathbf{x}_k)$ эквивалентна задаче минимизации функции $\varphi(\alpha) = f(\mathbf{x}_k - \alpha \mathbf{S}_k)$ при фиксированном значении точки $\mathbf{x}_k$.

Метод наискорейшего спуска обладает важным свойством, которое заключается в том, что при достаточно малом шаге итерации обеспечивается выполнение неравенства $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$.

Кроме того, метод, как правило, позволяет существенно уменьшить значение целевой функции при движении из точек, расположенных на значительном расстоянии от точки минимума, и поэтому часто используется при реализации градиентных методов в качестве начальной процедуры.

Метод Ньютона-Рафсона

Нетрудно видеть, что в методе наискорейшего спуска используется “наилучшая” локальная стратегия поиска с использованием градиента. Однако движение в направлении, противоположном градиенту, приводит в точку

минимума лишь в том случае, когда линии уровня функции представляют собой окружности.

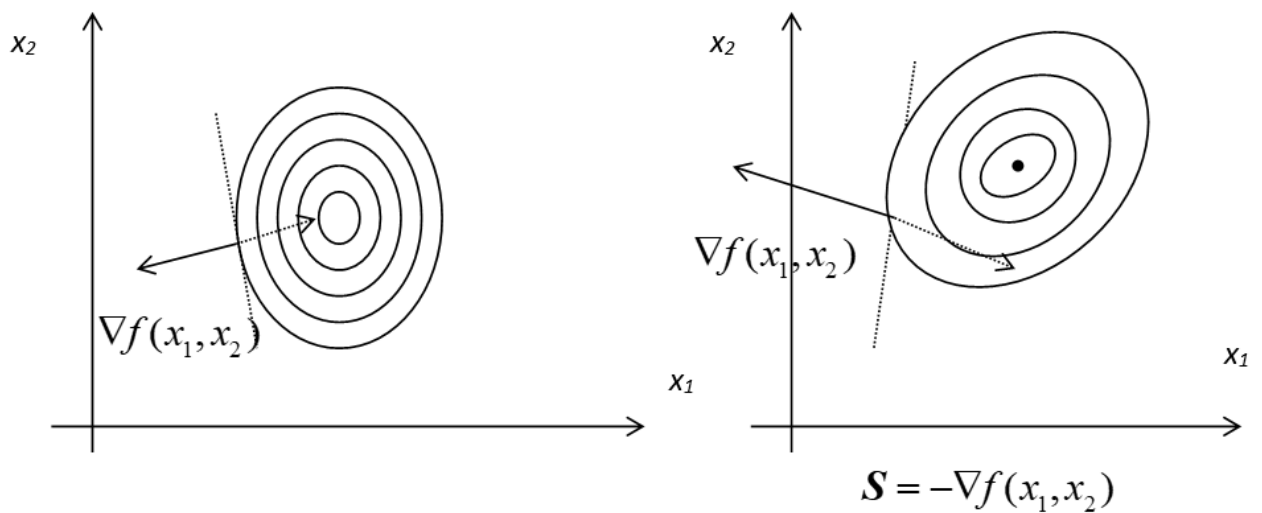


Рис. 4.4. Метод Ньютона-Рафсона

Таким образом, направление, противоположное градиенту, в общем случае не может быть приемлемым глобальным направлением поиска оптимума нелинейных функций.

В этом нет ничего удивительного, поскольку метод основывается на линейной аппроксимации целевой функции в окрестности текущей точки

$$f(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \cdot \Delta \mathbf{x}; \quad \Delta \mathbf{x} = (\mathbf{x} - \mathbf{x}_k)$$

Если в разложении непрерывной функции в ряд Тейлора сохранить еще один член разложения, то получим аппроксимацию любой функции в окрестности точки $\mathbf{x}_0$ квадратичной функцией следующего вида:

$$f(\mathbf{x}) \approx \varphi(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T \cdot \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{G} \Delta \mathbf{x}; \quad \text{где } \mathbf{G} - \text{матрица Гессе, матрица}$$

вторых частных производных, вычисленная в точке $\mathbf{x}_0$.

Например.

Если $f(\mathbf{x}) = f(x_1, x_2)$, то соответствующий Гессиан представляет собой

$$\text{матрицу } \mathbf{G} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix}.$$

Разумной аппроксимацией минимума функции $f(\mathbf{x})$ может быть минимум функции $\varphi(\mathbf{x})$. Если минимум функции $\varphi(\mathbf{x})$ находится в точке $\mathbf{x}_m$, то $\nabla \varphi(\mathbf{x}_m) = \mathbf{0}$; откуда $\nabla \varphi(\mathbf{x}_m) = \nabla f(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)(\mathbf{x}_m - \mathbf{x}_0) = \mathbf{0}$.

$$\mathbf{x}_m = \mathbf{x}_0 - (\mathbf{G}(\mathbf{x}_0))^{-1} \cdot \nabla f(\mathbf{x}_0)$$

Таким образом, можно модифицировать общее итерационное уравнение градиентных методов $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{S}_k$ и записать его в виде $\mathbf{x}_{k+1} = \mathbf{x}_k - (\mathbf{G}(\mathbf{x}_k))^{-1} \cdot \nabla f(\mathbf{x}_k)$

или в варианте

$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{G}(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k)$, где длина шага α_k определяется одномерным поиском вдоль оси направления $\mathbf{G}(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k)$.

Пример.

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

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; \nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 + 1 \\ 2x_2 \end{pmatrix}; \mathbf{G}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Обратная матрица определяется из условия:

$$\mathbf{G}(\mathbf{x}) \cdot (\mathbf{G}(\mathbf{x}))^{-1} = \mathbf{I}, \text{ в данном случае } \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Обращение диагональных матриц проблем не представляет

$$\mathbf{G}^{-1} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}.$$

При $\mathbf{x}_0 = (1, 1)^T$ по формуле метода Ньютона-Рафсона получаем

$$\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{G}^{-1} \cdot \nabla f(\mathbf{x}_0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} 3.5 \\ 2 \end{pmatrix} = \begin{pmatrix} -0.5 \\ 0 \end{pmatrix}$$

точка $\mathbf{x}_1 = (-0.5, 0)$ является точкой оптимума. Возьмем другую начальную точку $\mathbf{x}_0 = (10, 10)^T$, тогда $\mathbf{x}_1 = \begin{pmatrix} 10 \\ 10 \end{pmatrix} - \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} 21 \\ 20 \end{pmatrix} = \begin{pmatrix} -0.5 \\ 0 \end{pmatrix}$, т. е. снова $\mathbf{x}_1 = \mathbf{x}^*$.

Таким образом задача минимизации квадратичной функции решается по методу Ньютона-Рафсона с помощью одной итерации при любой начальной точке.

В случае произвольного вида функции потребуется большее количество итераций. Причем метод требует вычисления и обращения матрицы Гессе на каждом шаге, что часто является основной частью вычислений.

Если точка $\mathbf{x}_k$ близко расположена к точке оптимума, то сходимость метода будет быстрой, поскольку квадратичная функция $\varphi(\mathbf{x})$ хорошо аппроксимирует функцию общего вида в малой окрестности оптимальной точки.

Можно сказать, что, если метод наискорейшего спуска эффективен при поиске на значительных расстояниях от точки минимума $\mathbf{x}^*$ и плохо работает в окрестности этой точки, то метод Ньютона-Рафсона не отличается высокой надежностью при поиске $\mathbf{x}^*$ из удаленной точки, однако оказывается весьма эффективным, когда $\mathbf{x}_k$ находится вблизи точки минимума.

Метод сопряженных градиентов Флетчера-Ривса

Метод сопряженных градиентов обладает положительными свойствами двух других градиентных методов и основан на вычислении значений только первых производных.

Основная идея алгоритма заключается в том, что если квадратичная функция N переменных приведена к виду суммы полных квадратов, то ее оптимум может быть найден в результате реализации N одномерных поисков по преобразованным координатным направлениям.

Например.

$f(\mathbf{x}) = 4x_1^2 + 3x_2^2 - 4x_1x_2 + x_1$ - квадратичная функция общего вида;

$\varphi(\mathbf{x}) = 4x_1^2 + 2x_2^2 - x_1 + \frac{1}{2}x_2$ - функция вида суммы полных квадратов, т. е.

функция $\varphi(\mathbf{x})$ не содержит перекрестных произведений аргументов.

Процедура преобразования квадратичной функции $g(\mathbf{x}) = \mathbf{a} + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x}$ к виду суммы квадратов эквивалентна нахождению такой матрицы преобразования $\mathbf{T}$, которая приводила бы матрицу квадратичной формы к диагональному виду.

Таким образом заданная квадратичная форма $Q(\mathbf{x}) = \mathbf{x}^T \mathbf{C} \mathbf{x}$ путем преобразования координат $\mathbf{x} = \mathbf{T} \cdot \mathbf{z}$ приводится к виду $Q(\mathbf{z}) = \mathbf{z}^T \mathbf{T}^T \mathbf{C} \mathbf{T} \mathbf{z} = \mathbf{z}^T \mathbf{D} \mathbf{z}$, где $\mathbf{D}$ - диагональная матрица, т. е. ее элементы отличны от нуля только при $i=j$.

Пример.

Рассмотрим функцию $f(\mathbf{x}) = 4x_1^2 + 3x_2^2 - 4x_1x_2 + x_1$ и преобразование $x_1 = z_1 + \frac{1}{2}z_2$; $x_2 = z_2$.

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & 1/2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \text{ или } \mathbf{x} = \begin{pmatrix} 1 & 1/2 \\ 0 & 1 \end{pmatrix} \mathbf{z} \text{ где } \mathbf{T} = \begin{pmatrix} 1 & 1/2 \\ 0 & 1 \end{pmatrix}.$$

Подставив выражения для x_1 и x_2 в функцию $f(\mathbf{x})$ получим квадратичную функцию, преобразованную к виду суммы квадратов $f(\mathbf{z}) = 4z_1^2 + 3z_2^2 - z_1 + \frac{1}{2}z_2$.

Пусть t_j - j -я строка матрицы $\mathbf{T}$, тогда преобразование $\mathbf{x} = \mathbf{T} \cdot \mathbf{z}$ позволяет записать каждый вектор $\bar{\mathbf{x}}$ в виде линейной комбинации векторов t_j
 $\mathbf{x} = \mathbf{T}\mathbf{z} = t_1 z_1 + t_2 z_2 + \dots + t_N z_n$.

Иначе говоря, вместо координат вектора в стандартной координатной системе, где $\mathbf{x} = e_1 x_1 + e_2 x_2 + \dots + e_n x_n$ мы записываем функцию f от координат вектора $\mathbf{x}$ в новой координатной системе.

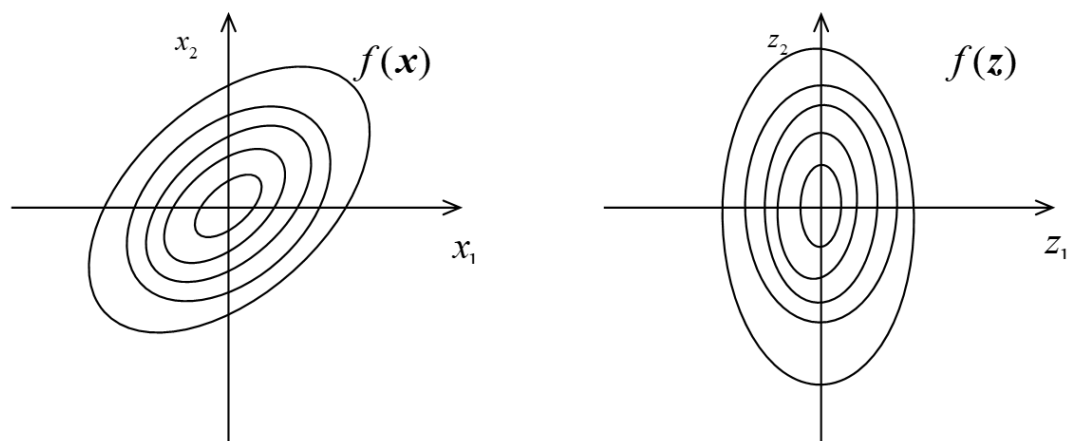


Рис. 4.5. Преобразования квадратичной формы

Поскольку матрица преобразований квадратичной формы имеет диагональный вид, то при записи функции в новой координатной системе оси эллипсов, изображающие линии уровня, будут осями координат.

Итак, с помощью преобразования переменных квадратичной функции строится новая система координат, параллельных осям квадратичной функции.

Следовательно, поиск методом покоординатного спуска в новой координатной системе дает максимальный эффект и позволяет достигнуть минимума функции ровно за N итераций (одномерных поисков).

Полученная система векторов, приводящих матрицу C к диагональному виду, называется системой C -сопряженных направлений.

Поскольку $T^T C T = D$, а все недиагональные элементы матрицы D равны нулю.

Отсюда вытекает следующее определение сопряженных направлений:

Пусть C - симметрическая матрица порядка $N \times N$;

Два направления p и q сопряжены относительно матрицы C , если $p^T C q = 0$.

В методе Флетчера-Ривса для получения сопряженных градиентов используется квадратичная аппроксимация функции $f(x)$ и значения компонент градиента.

Исходя из сказанного предположим, что целевая функция является квадратичной $f(x) = a + b^T x + \frac{1}{2} x^T \cdot C \cdot x$.

Итерации, как и в других градиентных методах, проводятся по формуле $x_{k+1} = x_k + \alpha_k S(x_k)$.

Для удобства записи введем обозначение градиента $\nabla f(x) = g(x)$.

Направление поиска на каждой итерации определяется с помощью следующих формул $S_k = -g_k + \sum_{i=0}^{k-1} \gamma_i S_i$.

Иначе говоря, формируется следующая последовательность N векторов-направлений.

$$S_0 = -g_0;$$

$$S_1 = -g_1 + \gamma_0 S_0;$$

$$S_2 = -g_2 + \gamma_0 S_0 + \gamma_1 S_1;$$

:

$$\mathbf{S}_n = -\mathbf{g}_n + \gamma_0 \mathbf{S}_0 + \dots + \gamma_{n-1} \mathbf{S}_{n-1}.$$

Коэффициенты γ_i выбираются таким образом, чтобы всякое направление $\mathbf{S}_k$ было сопряжено по матрице $\mathbf{C}$ со всеми построенными ранее направлениями поиска.

Рассмотрим первое направление

$$\mathbf{S}_1 = -\mathbf{g}_1 + \gamma_0 \mathbf{S}_0 = -\mathbf{g}_1 - \gamma_0 \mathbf{g}_0.$$

Наложим на него условие сопряженности с направлением $\mathbf{S}_0$:

$$\mathbf{S}_1^T \cdot (\mathbf{C}) \cdot \mathbf{S}_0 = 0, \text{ откуда } [\mathbf{g}_1 + \gamma_0 \mathbf{g}_0]^T \cdot \mathbf{C} \cdot \mathbf{S}_0 = 0.$$

На начальной итерации, согласно формуле (2): $\mathbf{S}_0 = (\mathbf{x}_1 - \mathbf{x}_0)/\alpha_0 = \Delta\mathbf{x}/\alpha_0$; следовательно, $[\mathbf{g}_1 + \gamma_0 \mathbf{g}_0]^T \cdot \mathbf{C} \cdot [\Delta\mathbf{x}/\alpha_0] = 0$ или $[\mathbf{g}_1 + \gamma_0 \mathbf{g}_0]^T \cdot \mathbf{C} \cdot \Delta\mathbf{x} = 0$.

Свойство 1 (квадратичной функции)

Пусть в пространстве переменных заданы 2 произвольные несовпадающие точки $\mathbf{x}_0$ и $\mathbf{x}_1$.

Градиент квадратичной функции равен $\nabla f(\mathbf{x}) = \mathbf{C} \cdot \mathbf{x} + \mathbf{b} = \mathbf{g}(\mathbf{x})$.

Таким образом $\mathbf{g}(\mathbf{x}_0) = \mathbf{C} \cdot \mathbf{x}_0 + \mathbf{b}$, $\mathbf{g}(\mathbf{x}_1) = \mathbf{C} \cdot \mathbf{x}_1 + \mathbf{b}$

Запишем изменение градиента при переходе от точки $\mathbf{x}_0$ к точке $\mathbf{x}_1$

$$\Delta\mathbf{g}(\mathbf{x}) = \mathbf{g}(\mathbf{x}_1) - \mathbf{g}(\mathbf{x}_0) = \mathbf{C} \cdot (\mathbf{x}_1 - \mathbf{x}_0) =$$

$$\Delta\mathbf{g}(\mathbf{x}) = \mathbf{C} \cdot |\Delta\mathbf{x}|.$$

Используя свойство 1 квадратичных функций из выражения, получаем $[\mathbf{g}_1 + \gamma_0 \mathbf{g}_0]^T \Delta\mathbf{g} = 0$; $[\mathbf{g}_1 + \gamma_0 \mathbf{g}_0]^T [\mathbf{g}_1 - \mathbf{g}_0] = 0$

$$\text{откуда } \mathbf{g}_1^T \cdot \mathbf{g}_1 + \gamma_0 \mathbf{g}_0^T \cdot \mathbf{g}_1 - \mathbf{g}_1^T \mathbf{g}_0 - \gamma_0 \mathbf{g}_0^T \mathbf{g}_0 = 0$$

Свойство 2.

Пусть из точки x_0 производится поиск минимума функции $f(x)$ в направлении $-g_0$. Иначе говоря, находится минимум функции $f(x_0 - \alpha g_0) = f(\alpha)$. Тогда в точке минимума производная функции равна нулю $\left(\frac{\partial f}{\partial \alpha}\right)_{x=x_1} = 0$.

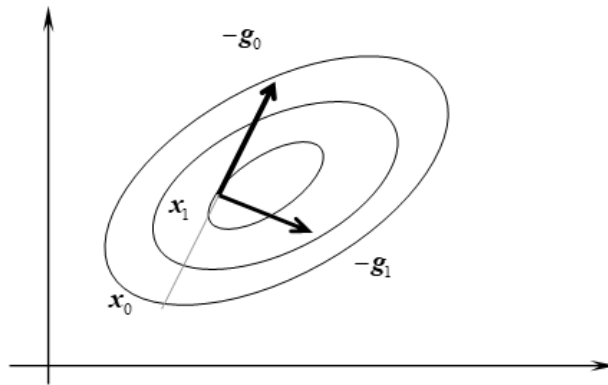


Рис. 4.6. Метод Флетчера-Ривса

Последнее означает, что вектор $-g_0$, задающий направление поиска, направлен в точку x_1 по касательной к линии уровня. В свою очередь, градиент функции $\nabla f(x_1)$ в точке x_1 перпендикулярен касательной, и следовательно, произведение векторов g_0 и g_1 равно нулю $g(x_0)^T \cdot g(x_1) = g_0^T \cdot g_1 = 0$.

Учитывая свойство 2 получаем $\gamma_0 = \|g_1\|^2 / \|g_0\|^2$.

В общем случае, используя свойство 1 и свойство 2 можно доказать, что в формуле $s_k = -g_k + \sum_{i=0}^{k-1} \gamma_i s_i$ коэффициенты $\gamma_i = 0$, $i = 0, 1, 2, \dots, k-2$.

Отсюда следует общая формула для направлений поиска, предложенная Флетчером и Ривсом: $s_k = -g_k + \left[\|g_k\|^2 / \|g_{k-1}\|^2 \right] \cdot s_{k-1}$.

Таким образом, если $f(x)$ - квадратичная функция, то для нахождения точки минимума требуется провести N одномерных поисков вдоль сопряженных направлений. Если же функция $f(x)$ не является квадратичной,

количество поисков возрастает. На основе поисков предполагается на каждом $N+1$ -ом шаге принимать за направление поиска направление наискорейшего спуска $\mathbf{S}_{n+1} = -\mathbf{g}(\mathbf{x}_n)$, то есть возвращаться к началу алгоритма.

О методах оптимизации в машинном обучении

Обучение нейросети включает в себя минимизацию функции ошибки (loss function) по отношению к ее параметрам (weights), которых может быть очень много. Поэтому существуют множество методов оптимизации для решения данной проблемы.

Градиентный спуск (Gradient Descent)

Градиентный спуск — это простейший метод для последовательного нахождения минимума дифференцируемой функции (в случае нейронных сетей это функция стоимости). Имея несколько параметров (веса сети) и дифференцируя по ним функцию получаем вектор частных производных или вектор градиента $\nabla f(\mathbf{x}) = \left\{ \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right\}$

Градиент всегда указывает по направлению максимального роста функции. Если же мы будем двигаться в противоположную сторону (т.е. $-\nabla f$) то со временем придем к минимуму, что нам и требовалось. Простейший алгоритм градиентного спуска:

1. Инициализация: случайно выбираем параметры $\mathbf{x}_0$
2. Вычисляем градиент: $\nabla f(\mathbf{x}_i)$
3. Изменяем параметры в сторону отрицательного градиента: $\mathbf{x}_{i+1} = \mathbf{x}_i - \alpha \nabla f(\mathbf{x}_i)$, где α — некоторый параметр скорости обучения (learning rate)
4. Повторяем предыдущие шаги пока градиент не станет достаточно близок к нулю

Градиентный спуск это довольно простой в реализации и хорошо зарекомендовавший себя метод оптимизации, но есть и минус — он первого порядка, это значит, что берется первая производная по функции стоимости, т.е. наша функция стоимости локально выглядит как плоскость и не учитывается ее кривизна.

Метод Ньютона (Newton's Method)

А что, если мы возьмем и будем использовать информацию, которую нам дает вторые производные по функции стоимости? Самый известный метод оптимизации с использованием вторых производных — это метод Ньютона. Основная идея этого метода заключается в минимизации квадратичной аппроксимации функции стоимости.

Рассмотрим многомерный случай. Положим у нас есть многомерная функция $f: R^n$, тогда: $x_{i+1} = x_i - H(f(x_i))^{-1} \nabla f(x_i)$

Проблемы с Методом Ньютона

Как мы можем видеть метод Ньютона — это метод второго порядка и будет работать лучше, чем обычный градиентный спуск, поэтому вместо того, чтобы в каждом шаге двигаться к локальному минимуму он движется к глобальному минимуму, если мы предполагаем, что функция f квадратичная и разложение второго порядка в ряд Тейлора её хорошо аппроксимирует. Но у данного метода есть **один большой минус**. Для оптимизации функции стоимости надо находить матрицу Гессе или гессиан H .

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

Как можно видеть гессиан это матрица вторых производных размера $n \times n$ и что бы ее посчитать потребуется $O(n^2)$ вычислительных операций, что может быть очень критично для сетей, у которых сотни или тысячи параметров. Помимо этого, для решения задачи оптимизации с помощью метода Ньютона необходимо найти обратную матрицу Гессе, для этого она должна быть положительно определенной для всех.

Hessian-Free оптимизация

Основная идея HF оптимизации заключается в том, что за основу мы берем метод Ньютона, но используем более подходящий способ минимизации квадратичной функции. Но для начала введем основные понятия, которые понадобятся в дальнейшем.

Пусть $\theta = (W, b)$ — параметры сети, где W — матрица весов (weights), b — вектор смещений (biases), тогда выходом сети назовем: $f(x, \theta)$, где x — входной вектор. $L(t, f(x, \theta))$ — функция потерь (loss function), t — целевое значение. А функцию которую будем минимизировать определим как среднюю от потерь по всем тренировочным примерам (training batch) $\underline{S}$:

$$h(\theta) = \frac{1}{|\underline{S}|} \sum_{(x,y) \in \underline{S}} L(t, f(x, \theta))$$

Далее согласно методу Ньютона определим квадратичную функцию, полученную путем разложения в ряд Тейлора второго порядка:

$$h(\theta + \delta) = M(\delta) = h(\theta) + \nabla h(\theta)^T \delta + \frac{1}{2} \delta^T H \delta \quad (4.1)$$

Далее взяв производную по δ от формулы выше и приравнявая её к нулю, получаем:

$$\nabla M(\delta) = \nabla h(\theta) + H \delta = 0 \quad (4.2)$$

Что бы найти δ будем использовать метод сопряженных градиентов (conjugate gradient method).

Метод сопряженных градиентов (CG) — это итерационный метод решения систем линейных уравнений типа $Ax=b$:

Краткий алгоритм CG:

Входные данные: $b, A, x_0, i = 0$ - шаг алгоритма CG

Инициализация:

1. $r_0 = b - Ax_0$ — вектор ошибки (residual)
2. $d_0 = r_0$ — вектор направления поиска (search direction)

Повторяем пока не выполнится условие остановки:

1. $\alpha_i = \frac{r_i^T r_i}{d_i^T A d_i}$
2. $x_{i+1} = x_i + \alpha_i d_i$
3. $r_{i+1} = r_i - \alpha_i A d_i$
4. $\beta_i = \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i}$
5. $d_{i+1} = r_{i+1} + \beta_i d_i$
6. $i = i + 1$

Теперь с помощью метода сопряженных градиентов мы можем решить уравнение $\nabla h(\theta) + H\delta = 0$ и найти δ , которая будет минимизировать

$h(\theta + \delta) = h(\theta) + \nabla h(\theta)^T \delta + \frac{1}{2} \delta^T H \delta$. В нашем случае это

$A = H, b = -\nabla h(\theta), x = \delta$.

Остановка CG алгоритма. Останавливать метод сопряженных градиентов можно исходя из разных критериев. Мы будем делать это на основе относительного прогресса в оптимизации квадратичной функции M :

$$s_i = \frac{M(\delta_i) - M(\delta_{i-w})}{M(\delta_i) - M(0)} \quad (4.3)$$

где w — размер «окна», по которому будем считать значение прогресса, $w = \max(10, \frac{i}{10})$. Условием остановки возьмем: $s_i < 0.0001$.

А теперь можно заметить, что **главная особенность НФ оптимизации** заключается в том, что нам не требуется находить гессиан напрямую, а надо лишь найти результат его произведения на вектор.

Умножение гессиана на вектор

Как уже говорилось ранее плюс данного метода заключается в том, что у нас нет необходимости считать гессиан напрямую. Надо лишь посчитать результат произведения матрицы вторых производных на вектор. Для этого можно представить $H(\theta)v$ как производную от $H(\theta)$ по направлению v :

$$H(\theta)v = \lim_{\varepsilon \rightarrow 0} \frac{\nabla h(\theta + \varepsilon v) - \nabla h(\theta)}{\varepsilon}$$

Но использование данной формулы на практике может вызвать ряд проблем связанные с вычислением при достаточно маленьком ε . Поэтому существует метод точного вычисления произведения матрицы на вектор. Введем дифференциальный оператор $R_v x$

Он обозначает производную некоторой величины x , зависящей от θ , по направлению v :

$$R_v x = \lim_{\varepsilon \rightarrow 0} \frac{x(\theta + \varepsilon v) - \nabla h(\theta)}{\varepsilon} = \frac{\delta x}{\delta \theta} v \quad (4.4)$$

Отсюда видно, чтобы посчитать произведение гессиана на вектор надо вычислить: $Hv = R(\nabla h(\theta))$.

Некоторые улучшения НФ оптимизации

1. Обобщенная матрица Ньютона-Гаусса (generalized Gauss-Newton matrix).

Неопределенность матрицы гессиана является проблемой для оптимизации не выпуклых (non-convex) функций, она может привести к отсутствию нижней границы для квадратичной функции M и как следствие невозможность нахождения ее минимума. Эту проблему можно решить множеством способов. Например, введение доверительного интервала будет ограничивать оптимизацию или же дампинг (damping) на основе штрафа, который добавляет положительный полуопределенный (positive semi-definite) компонент к матрице кривизны H и делает её положительно определенной.

Основываясь же на практических результатах наилучшим способом решить данную проблему, является использование матрицы Ньютона-Гаусса G вместо матрицы Гессе:

$$G = \frac{1}{|S|} \sum_{(x,y) \in S} J^T H_L J \quad (4.5)$$

где J — якобиан, H_L — матрица вторых производных от функции потерь $L(t, f(x, \theta))$. Для нахождения произведения матрицы G на вектор v : $Gv = J^T H_L J$, сначала найдем произведение якобиана на вектор $Jv = R_v(f(x, \theta))$, потом вычислим произведение вектора Jv на матрицу H_L и в конце умножим матрицу J на $H_L Jv$.

2. Дампинг (damping).

Стандартный метод Ньютона может плохо оптимизировать сильно нелинейные целевые функции. Причиной этому может быть то, что на начальных стадиях оптимизации она может быть очень большой и агрессивной, так как начальная точка находится далеко от точки минимума. Для решения этой проблемы используется дампинг — метод изменения квадратичной функции M или ограничения минимизации таким образом, что новая δ будет лежать в таких пределах, в которых M останется хорошей аппроксимацией.

Регуляризация Тихонова (Tikhonov regularization) или демпинг Тихонова (Tikhonov damping). (Не путать с термином «регуляризация», который обычно используется в контексте машинного обучения) Это самый известный метод демпинга, который добавляет квадратичный штраф к функции M :

$$\hat{M}(\delta) = M(\delta) + \frac{\lambda}{2} \delta^T \delta = f(\theta) + \nabla h(\theta)^T \delta + \frac{1}{2} \delta^T \hat{H} \delta$$

где, $\hat{H} = H + \lambda I$, λ — параметр демпинга. Вычисление Hv производится по

$$\hat{H}v = (H + \lambda I)v = Hv + \lambda Iv \quad (4.6)$$

3. Эвристика Левенберга-Марквардта (Levenberg-Marquardt heuristic).

Для демпинга Тихонова характерно динамическое подстраивание параметра λ . Изменять λ будем по правилу Левенберга-Марквардта, который часто используется в контексте LM — метода (метод оптимизации, является альтернативой методу Ньютона). Для использования LM — эвристики необходимо посчитать так называемый коэффициент уменьшения (reduction ratio):

$$\rho = \frac{h(\theta_k + \delta_k) - h(\delta_k)}{M_k(\delta_k) - M_k(0)} = \frac{h(\theta_k + \delta_k) - h(\delta_k)}{\nabla h(\theta_k)^T \delta_k + \frac{1}{2} \delta_k^T H \delta_k} \quad (4.7)$$

где k — номер шага HF алгоритма, δ_k — результат работы CG минимизации.

Согласно эвристики Левенберга-Марквардта получаем правило обновления λ :

$$\begin{cases} \text{if } \rho > \frac{3}{4} \text{ then } \lambda \leftarrow \frac{2}{3} \lambda \\ \text{if } \rho < \frac{1}{4} \text{ then } \lambda \leftarrow \frac{3}{2} \lambda \end{cases} \quad (4.8)$$

4. Начальное условие для алгоритма сопряженных градиентов (preconditioning).

В контексте HF оптимизации у нас есть некоторая обратимая матрица трансформации C , с помощью которой мы изменяем M таким образом, что

$\delta = C^{-1}\gamma$ и вместо δ минимизируем γ . Применение данной особенности в алгоритме CG требует вычисление трансформированного вектора ошибки $y_i = P^{-1}r_i$, где $P = C^T C$.

Краткий алгоритм PCG (Preconditioned conjugate gradient):

Входные данные: b , A , x_0 , P , $i=0$ — шаг алгоритма CG

Инициализация:

1. $r_0 = b - Ax_0$ — вектор ошибки (residual)
2. y_0 — решение уравнения $Py_0 = r_0$
3. $d_0 = y_0$ — вектор направления поиска (search direction)

Повторяем пока не выполнится условие остановки:

1. $\alpha_i = \frac{r_i^T y_i}{d_i^T A d_i}$
2. $x_{i+1} = x_i + \alpha_i d_i$
3. $r_{i+1} = r_i - \alpha_i A d_i$
4. y_i — решение уравнения $Py_i = r_i$
5. $\beta_i = \frac{r_{i+1}^T y_{i+1}}{r_i^T y_i}$
6. $d_{i+1} = y_{i+1} + \beta_i d_i$
7. $i = i + 1$

Выбор матрицы P довольно нетривиальная задача. Так же, на практике использование диагональной матрицы (вместо матрицы с полным рангом) показывает довольно хорошие результаты. Один из вариантов выбора матрицы P — это использование диагональной матрицы Фишера (Empirical Fisher Diagonal):

$$P = \text{diag}(\bar{F}) = \frac{1}{|S|} \sum_{(x,y) \in S} (\nabla L(t, f(x, \theta)))^2 \quad (4.9)$$

5. Инициализация CG — алгоритма

Хорошей практикой является инициализация начальной δ_0 , для алгоритма сопряженных градиентов, значением δ_k , найденным на предыдущем шаге алгоритма HF. При этом можно использовать некоторую константу распада: $\delta_0 = \varepsilon \delta_k$, $\varepsilon \in (0, 1)$. Стоит отметить, что индекс k относится к номеру шага алгоритма HF, в свою очередь индекс 0 в δ_0 относится к начальному шагу алгоритма CG.

Полный алгоритм Hessian-Free оптимизации

Входные данные: θ, λ параметр дампинга, i — шаг итерации алгоритма
Инициализация:

1. $\delta_0 = 0$

Основной цикл HF оптимизации:

1. Вычисляем матрицу P

2. Находим δ_k решая задачу оптимизации с помощью CG или PCG
$$\delta_k = CG(-\nabla h(\theta_k), H, \varepsilon \delta_k, P)$$

3. Обновляем параметр дампинга λ с помощью эвристики Левенберга-Марквардта

4. $\theta_{k+1} = \theta_k + \alpha \theta_k$ — параметр скорости обучения (learning rate)

5. $k = k + 1$

Таким образом метод Hessian-Free оптимизации позволяет решать задачи нахождения минимума функции большой размерности. При этом не требуется нахождения матрицы Гессе напрямую.