

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Государственное образовательное учреждение высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Е.А. Кочегурова

ТЕОРИЯ И МЕТОДЫ ОПТИМИЗАЦИИ

Издательство
Томского политехнического университета
2012

УДК 519.85
ББК 22.18
К-55

Кочегурова Е.А.

К-55 Теория и методы оптимизации / Е.А. Кочегурова; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2012. –157с.

ISBN: 978-5-9221-0170-7

Учебное пособие посвящено одному из важнейших направлений подготовки выпускника технического университета - математической теории оптимизации. Рассмотрены теоретические и алгоритмические аспекты методов конечномерной оптимизации. Структура всех разделов учебного пособия однотипна и ориентирована на полный цикл изучения задачи оптимизации от теоретического основ до алгоритмизации конкретных методов.

Учебное пособие предназначено для студентов направления 220400 «Управление в технических системах».

УДК 519.85
ББК 22.18

Рецензенты

Кандидат технических наук, доцент ТУСУР
Р.М. Мещеряков

Кандидат технических наук, доцент ТУСУР
А.А.Шелестов

ISBN: 978-5-9221-0170-7

© ГОУ ВПО НИ ТПУ, 2011
© Кочегурова Е.А., 2011
© Оформление. Издательство Томского
политехнического университета, 2011

ВВЕДЕНИЕ

Понятие оптимальности, как и понятие процесса оптимизации, — центральный момент не только в экономике, инженерном деле, менеджменте и бизнесе, оно также используется и во многих социальных и биологических науках. Термин «оптимальный» чаще всего трактуют как благоприятный, максимальный (минимальный), наиболее эффективный и др. Каждый человек ежедневно, не всегда осознавая это, решает проблему: как получить наибольший эффект, обладая ограниченными ресурсами.

Математическое программирование — раздел математики, посвященный теории и методам решения задач поиска экстремумов функций на множестве ограничений. Оно объединяет различные математические методы и дисциплины исследования операций: линейное программирование, нелинейное программирование, динамическое программирование, выпуклое программирование, геометрическое программирование, целочисленное программирование и др.

Здесь термин "программирование" обязано отчасти историческому недоразумению, отчасти неточному переводу с английского. В английском языке слово «programming» означает планирование, составление планов или программ. Вполне естественно, что терминология отражает тесную связь, существующую между математической постановкой задачи и её экономической интерпретацией. Поэтому наименование «Математическое программирование» связано с тем, что целью решения задач является выбор оптимальной программы действий.

1. ПОСТАНОВКА И КЛАССИФИКАЦИЯ ЗАДАЧ ОПТИМИЗАЦИИ

В самом общем случае, решить оптимизационную задачу это значит найти наилучшее решение среди возможных вариантов решения.

Если оптимизация связана с расчетом оптимальных значений параметров при заданной структуре объекта, то она называется параметрической. Задача выбора оптимальной структуры является структурной оптимизацией.

Решение любой оптимизационной задачи основано на построении математической модели исследуемого объекта и проведении вычислительного эксперимента. Проведение вычислительного (компьютерного) эксперимента не с самим объектом, а с его моделью дает возможность эффективно исследовать его свойства в любых ситуациях.

Основу вычислительного эксперимента составляет триада «модель—алгоритм—программа». Схема вычислительного эксперимента приведена на рисунке 1.1.

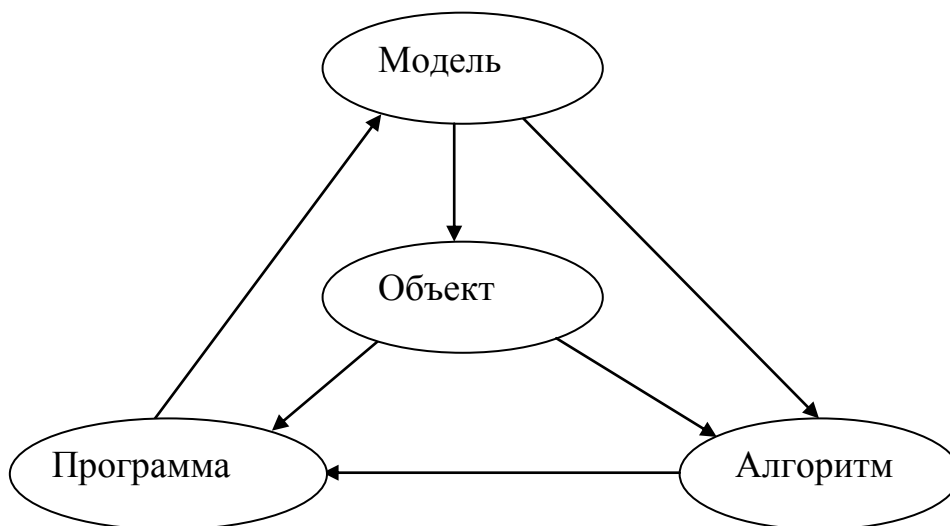


Рисунок 1.1. Схема вычислительного эксперимента

На первом этапе эксперимента строится некий эквивалент объекта, его модель, отражающий в математической форме важнейшие свойства объекта. Второй этап — разработка алгоритма для реализации модели на компьютере. Модель представляется в форме, удобной для

применения численных методов, определяется последовательность вычислительных и логических операций, которые нужно произвести, чтобы найти искомые величины с заданной точностью. На третьем этапе создаются программы, реализующие алгоритмы на доступном компьютеру языке.

Нахождение оптимальных значений параметров – это один из этапов вычислительного эксперимента, позволяющий выработать управляющее воздействие на объект исследования.

Теория оптимизации представляет собой совокупность фундаментальных математических результатов и численных методов, позволяющих избежать полного перебора всех решений.

Методы оптимизации – это методы построения алгоритмов нахождения оптимального (минимального или максимального) значения некоторой функции.

1.1. История становления и развития теории оптимизации

Термин "оптимум" был введен в 18 веке Готфридом В.Лейбницем и в основном рассматривался в применении к теологии, учении о религиозных догматах и религиозной культуре и их необходимости для человека.

Основа термина "optimus", в переводе с латинского означает наилучший. Его связывают с именем богини древнеиталийского племени сабинов Опы (богиня плодородия, урожая и богатства). Она – жена бога времени Сатурна и мать Юпитера (хранителя римского государства). В одной руке Она держит рог изобилия (мифический источник благ), а в другой - символ измерения и решения веса.

Лейбниц в своей философской теории излагал соображения о существующем мире как об оптимуме. Это переводилось как наилучший из всех возможных миров. Однако в философском учении Лейбница нет понятия допустимости. Но «наилучшее» может быть и недопустимым. Впоследствии его идеи взяты на вооружение философским течением «философский оптимизм», один из лозунгов которого звучит так: «Все что ни делается к лучшему».

Исторически выявлено несколько математических закономерностей лежащих в основе теории оптимизации.

17 век – Пьер Ферма установил закономерность, заключающуюся в том, что при приближении к точкам максимума и минимума скорость функции падает до нуля.

Ещё раньше практики – землеустроители использовали основные положения оптимального проектирования:

- кратчайшее расстояние между двумя точками – прямая;
- кривая заданной длины, ограничивающая максимальную площадь – окружность.

18 век – работы Даниила Бернулли, Леонарда Эйлера, Жозефа Л.Лагранжа, посвященные вариационному исчислению. Позже этими же задачами в 19 веке занимались Карл Вейерштрасс и Карл Г.Якоби.

Первыми, подробно изученными задачами поиска экстремума были задачи линейного программирования. Еще в 1820 г. Жозеф Фурье и затем Л. В. Канторович (1939 г.), Джордж Б.Данциг (1947 г.) сформулировали задачу линейного программирования и предложили метод ее решения - направленного перебора смежных вершин.

Так к середине 20 века произошло разделение теоретических разработок и практических нужд. Это несоответствие продолжалось вплоть до создания ЭВМ в конце 40-х годов прошлого столетия. После создания в 1947 году Д.Данцигом симплекс-метода и появления первых ЭВМ были сформулированы и решены тысячи прикладных задач. Несколько позже Р.Беллманом был разработан метод динамического программирования, который позволял решать задачи для систем, характеристики которых зависят от времени. Также существенный вклад в математическое программирование и оптимальное управление внес Л.С.Понтрягин, разработав раздел вариационного исчисления. Так к 70-м годам 20 века в основном был сформирован определенный раздел прикладной математики – теория и методы оптимизации.

1.2. Содержательная и формализованная постановка задачи

Решение оптимизационных задач (процесс оптимизации) лежит в основе всей инженерной деятельности.

Эффективность метода оптимизации связана с одной стороны с умелым использованием основных результатов таких математических дисциплин, как:

- математический анализ;
- линейная алгебра;
- математическая логика и теория алгоритмов;
- функциональный анализ;
- дифференциальное исчисление.

С другой стороны, в инженерной практике размер оптимизационных задач достаточно велик, следовательно, велики и затраты времени реализации алгоритмов. Поэтому оптимизационные задачи ориентированы в основном на реализацию на ЭВМ.

Ценность теории оптимизации состоит в её универсальности по отношению к различным классам задач.

Оптимизационная задача может быть частью задачи решаемой в:

- математической экономике: решение больших макроэкономических моделей, моделей предпринимательства и т.д.;
- в автоматике: оптимальное управление, робототехника, фильтрация, управление производством;
- в технике: оптимальное планирование информационных и компьютерных сетей, оптимизация структур и параметров технических систем;
- в численном анализе: аппроксимация, регрессия, решение линейных и нелинейных систем управления и т.д.

Важнейшим элементом во всех этих задачах является формализованная (математическая) постановка задачи. При этом задачи различаются содержательной стороной.

Рассмотрим ряд содержательных и формализованных задач оптимизации.

Пример 1.

Аппроксимация статической характеристики термопреобразователей.

Термопреобразователь сопротивления служит для измерения температуры среды. Его действие основано на изменении сопротивления R чувствительного элемента (Pt, Cu, AlKa, CrCa и т.д.) от температуры измеряемой среды T . Эта зависимость получена экспериментально в виде пары чисел

$$(R_i, T_i), \quad i = \overline{1, n},$$

и представляет собой, так называемую, статическую характеристику датчика температуры и хранится в виде градуировочных таблиц ГОСТ 6651-84. Каждая из таблиц (а их реально используют более 10) содержит ориентировочно 1000 значений. Хранить подобные объемы данных нецелесообразно для любого технического средства, а в

микропроцессорном контроллере невозможно. Поэтому используют аппроксимацию табличных данных в виде регрессионного управления.

Известно, что величины R и T связаны функциональной зависимостью:

$$T = \phi(R, \bar{a}) \quad (1.1)$$

где ϕ – вид функциональной связи;

\bar{a} - вектор неизвестных коэффициентов уравнения связи.

Предполагаем, что структура регрессионного управления известна (полиномиальная, гармоническая и др.).

Например, в случае использования полиномиального уравнения второго порядка, получаем аппроксимированные значения температуры \bar{T}

$$\bar{T} = a_0 + a_1 \cdot R + a_2 \cdot R^2. \quad (1.2)$$

Пример 2.

Транспортная задача

В городе имеются два кирпичных завода и три потребителя их продукции – строительные фирмы. Известны и приведены в таблице 1.1 суточные объемы производства кирпича, суточные потребности в нем строительных организаций и стоимость перевозки 1 тыс. кирпичей от каждого завода к каждой строительной фирме.

Таблица 1.1

Исходные данные для транспортной задачи

Заводы	Производство кирпича (тыс/сут)	Стоимость перевозки 1 т цемента (денежных ед.)		
		Комбинат 1	Комбинат	Комбинат
1	10	10	15	25
2	15	20	30	30
	Потребности в кирпичах (тыс/сут)	9	8	8

Требуется составить план суточных перевозок кирпича таким образом, чтобы транспортные расходы были минимальны.

Приведенные задачи различаются содержательной (описательной) постановкой задачи и имеют специфику конкретных прикладных

областей. Важнейшим моментом теории оптимизации является переход от содержательной постановки задачи к формализованной, то есть математической.

Стоит отметить, что решение задачи оптимизации это всего лишь один этап в процессе формирования оптимального проекта для эффективного функционирования системы. Процесс инженерного проектирования циклический и включает следующие этапы, приведенные на рисунке 1.2.

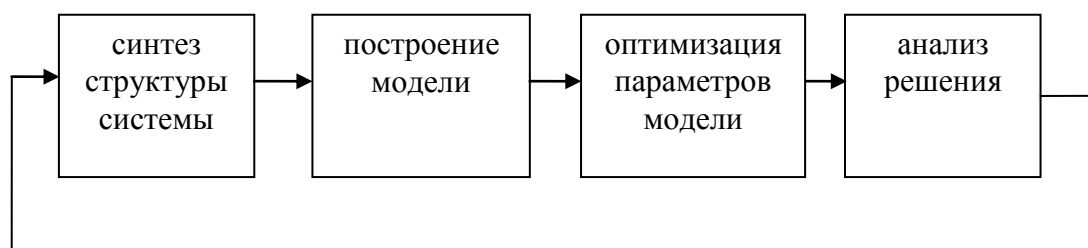


Рисунок 1.2. Процесс инженерного проектирования

Переход от содержательной к формализованной постановке задачи оптимизации включает ряд этапов, важнейшими из которых являются следующие.

1. Установить границы объекта оптимизации.

Границы объекта (системы) задаются пределами определяющими объект от внешней среды. Например, в задаче I мы не рассматривали процесс получения градуировочной таблицы. Однако при установлении границ должны быть уверены, что декомпозиция задачи не приведет к излишнему упрощению реального объекта.

2. Построение математической модели системы

Структура модели включает основные уравнения материальных и энергетических балансов и уравнения, описывающие физические процессы в системе. Эти уравнения обычно дополняются неравенствами, которые определяют область допустимых значений независимых переменных. Область допустимых значений определяет границы изменения характеристик объекта. Процесс построения математической модели – один из самых сложных этапов.

3. Выбор критерия оптимизации.

Зависит от решаемой задачи и может иметь экономический, точностный или надежность характер. Независимо от характера критерия, наилучшему решению задачи оптимизации всегда соответствуют минимум или максимум его значения.

Чаще всего критерий оптимальности носит экономическую оценку (производительность, себестоимость продукции, прибыль, рентабельность и др.).

Но если объект оптимизации является частью технологического процесса, то не всегда целесообразно выделять прямой экономический показатель, который бы полностью характеризовал эффективность работы рассматриваемого объекта. В таких случаях критерием оптимальности может служить технологическая характеристика, косвенно оценивающая экономичность работы устройства (технологические параметры и другие показатели качества продукции).

Основная задача при выборе критерия оптимизации – декомпозиция противоречивых целевых установок.

Например:

{
минимум затрат (вложений при проектировании);
минимум потребления энергии при эксплуатации;
максимум надежности.

В таком случае, чаще всего выделяют главный критерий, а остальные – вводятся в ограничение задачи оптимизирования. Подобные задачи относятся к классу задач векторной оптимизации.

4. Формирование целевой функции.

На основании выбранного критерия оптимизации необходимо составить целевую функцию (функцию качества), представляющую собой математическую зависимость критерия оптимальности от параметров, влияющих на ее значение. Вид целевой функции определяется конкретной задачей оптимизации.

Таким образом, задача оптимизации сводится к нахождению экстремума целевой функции.

Например, в примере I критерием оптимальности является точность аппроксимации (разность $[T - \bar{T}]$), а целевой функцией – квадрат расстояния $L(a_0, a_1, a_2) = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (T_i - \bar{T}_i)^2}$.

5. Построение оптимизационного алгоритма и решение экстремальной задачи.

Этот этап собственно и представляет выбор или разработку алгоритма решения инженерной задачи с использованием средств вычислительной техники.

1.3. Постановка задачи оптимизации

Несмотря на различные содержательные постановки задачи, структура оптимизационной задачи однотипна и содержит следующие компоненты.

1. Целевая функция $f(x)$ n -мерного векторного аргумента $x = (x_1, x_2, \dots, x_n)$, т.е.

$$F(x) \in R^n, \quad f(x) \in R^1.$$

2. Ограничения в виде неравенств $g_j(x) \geq 0$.

3. Ограничения в виде равенств $h_k(x) = 0$.

4. Область допустимых значений $x \in D \subset R^n$.

В общем виде задачи оптимизации имеет вид:

$$F(x) \rightarrow \min \tag{1.3}$$

ограничения I рода $h_k(x) = 0, \quad k = \overline{1, K},$

ограничения II рода $g_j(x) \geq 0, \quad j = \overline{1, J},$

$$x \in D \subset R^n.$$

1.4. Классификация задач оптимизации

В зависимости от вида целевой функции и соотношения ограничений выделяют различные задачи оптимизации, классификация которых приведена на рисунке 1.3.

Существует несколько признаков классификации. Основные критерии следующие.

1. По типу параметров задачи оптимизации. Различают непрерывные задачи оптимизации (continues optimization) и дискретные (discrete) и целочисленные (integer optimization).

2. По критерию размерности допустимого множества параметров D . Задачи оптимизации по этому критерию делятся на задачи одномерной оптимизации и задачи многомерной оптимизации.

3. Критерий наличия или отсутствия ограничений на допустимое множество D . Различают задачи условной (constrained) и безусловной (unconstrained) оптимизации. Этот признак классификации имеет место, как для одномерных, так и для многомерных задач оптимизации.

4. По характеру ограничений различают детерминированную оптимизацию и стохастическую. Если множество допустимых значений включает случайные компоненты, то имеет место стохастическое программирование. При этом стохастическая оптимизация может относиться и к дискретной задаче.

5. По виду целевой функции и виду ограничений различают линейное и нелинейное программирование.

Задача линейного программирования содержит линейную целевую функцию, ограничения в задаче также линейны.

При нарушении линейности целевой функции или ограничений имеет место нелинейная задача оптимизации. Классификация задач нелинейного программирования, в основном определена видом целевой функции и приведена на рисунке 1.4.

Ниже представлены постановки основных задач оптимизации в соответствии с рисунком 1.3.

Задачи одномерной безусловной оптимизации

$$\min f(x), x \in R^1 \quad (1.4)$$

Ограничения отсутствуют, $K = J = 0$, $D \equiv R^1$. Т.е. задача без ограничений с одномерным вектором. Вид $f(x)$ произвольный.

Задача многомерной безусловной оптимизации

$$\min f(x), x \in R^n \quad (1.5)$$

$K = J = 0$, $D \equiv R^n$, $x \in [-\infty, \infty]$ - ограничений нет

$f(x)$ – любого вида

Задача условной многомерной оптимизации

$$\begin{aligned} \min f(x), \\ h_k(x) = 0, \quad k = \overline{1, K} \\ g_j(x) \geq 0, \quad j = \overline{1, J} \\ x \in D, \quad i = \overline{1, N}. \end{aligned} \tag{1.6}$$

Задача линейного программирования

$$\begin{aligned} F(x) = \sum_{j=1}^n c_j \cdot x_j \longrightarrow \min \\ \sum_{j=1}^n a_{ij} \cdot x_j \leq b_i, \quad i = \overline{1, m}, \quad x_j \geq 0 \end{aligned} \tag{1.7}$$

Целевая функция – линейна, ограничения тоже линейны.

Наиболее известные классические задачи линейного программирования: транспортная задача, задача о диете и другие.

Задача целочисленного программирования

В задачах целочисленного программирования компоненты вектора $\{x\}$ принимают только целые значения. Известны классические задачи целочисленного программирования: задача о коммивояжере, задача раскраски графов, задачи теории расписания.

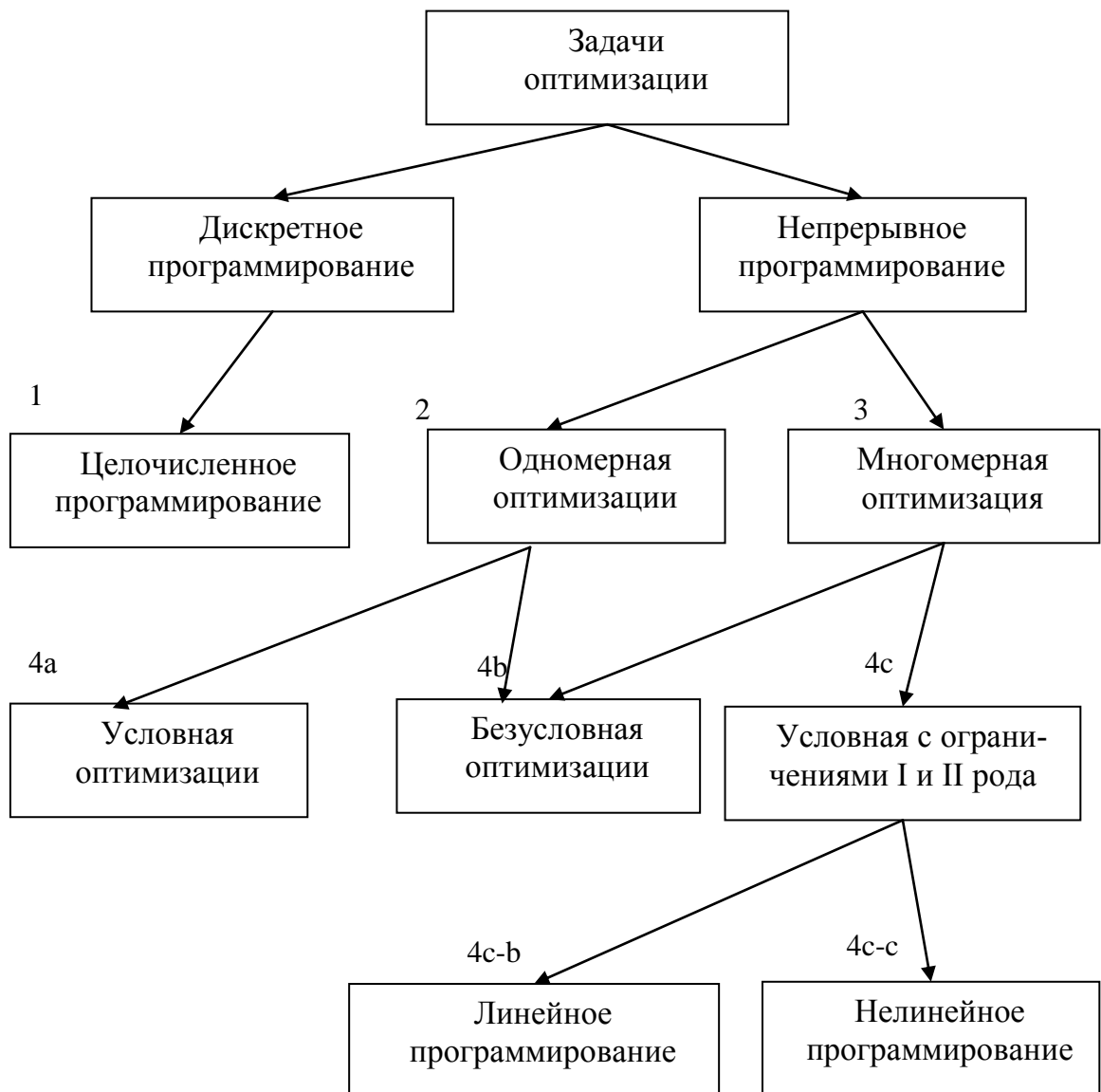


Рисунок 1.3. Классификация задач оптимизации

Задача нелинейного программирования

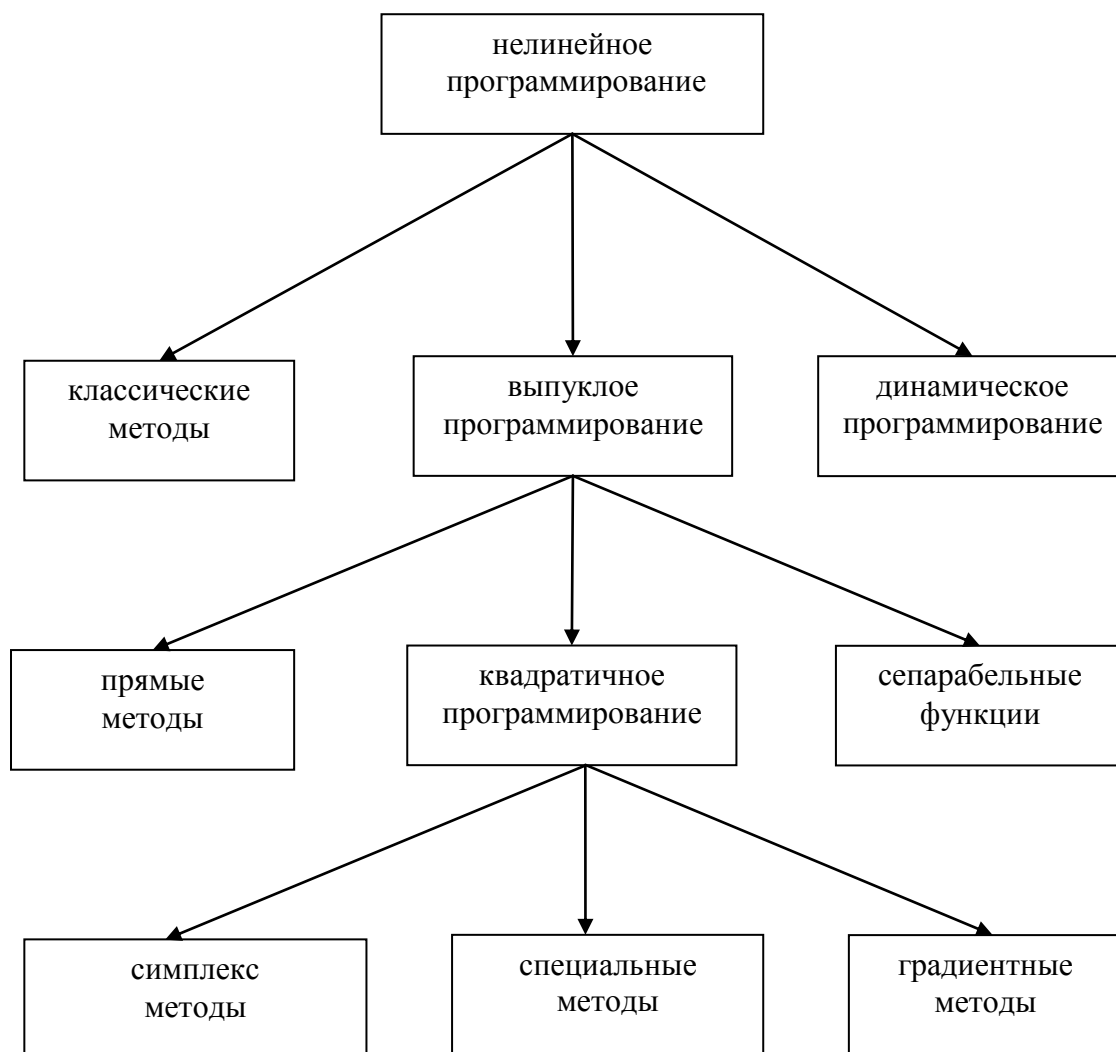


Рисунок 1.4. Классификация задач нелинейного программирования

Особо развито выпуклое и квадратичное программирование.

Приведенная классификация сделана по виду нелинейности целевой функции, при этом предполагалось, что ограничения $\phi(x)$ - линейны.

$$\left. \begin{array}{l} \min F(x) \\ \phi_i(x) \leq 0, i = \overline{1, m} \\ \phi_i(x) \leq 0, i = \overline{1, m} \end{array} \right\} \quad (1.8)$$

1.5. Анализ экстремальных задач условия существования экстремума

Определение. Функция $f(x)$, определенная на множестве D , достигает своего глобального \min в точке $x^* \in D$ в том и только том случае, если

$$f(x^*) \leq f(x), \forall x \in D.$$

Замечание. Если же можно указать ε – окрестность точки x^* в которой выполняется условие $f(x^*) \leq f(x)$, то говорят, что функция $f(x)$ имеет на множестве D локальный \min . Локальных экстремумов на интервале D может быть несколько.

Определение. Функция $f(x) \in \mathbb{R}^1$ является униmodalьной на отрезке $a \leq x \leq b$, если любой её локальный \min на отрезке $[a, b]$ одновременно является и глобальным.

Из определения унимодальной функции не следует правила проверки этого свойства функций. Однако в теории оптимизации можно выделить важный класс унимодальных функций – класс вогнутых и выпуклых функций, которые допускают такую проверку.

Определение: Функция $f(x)$, определенная на интервале $[a, b]$, называется выпуклой (выпуклая вниз функция) тогда и только тогда, когда для любых двух точек $x_1, x_2 \in [a, b]$ выполняется неравенство **Иенсена**

$$f[\lambda \cdot x_1 + (1 - \lambda) \cdot x_2] \leq \lambda \cdot f(x_1) + (1 - \lambda) \cdot f(x_2), \quad \lambda \in [0, 1]. \quad (1.9)$$

Т.е. прямая соединяющая две точки $f(x_1)$ и $f(x_2)$ лежит выше всех значений функции $f(x)$ при $x \in [x_1, x_2]$.

Геометрическая иллюстрация неравенства Иенсена приведена на рисунке 1.5.

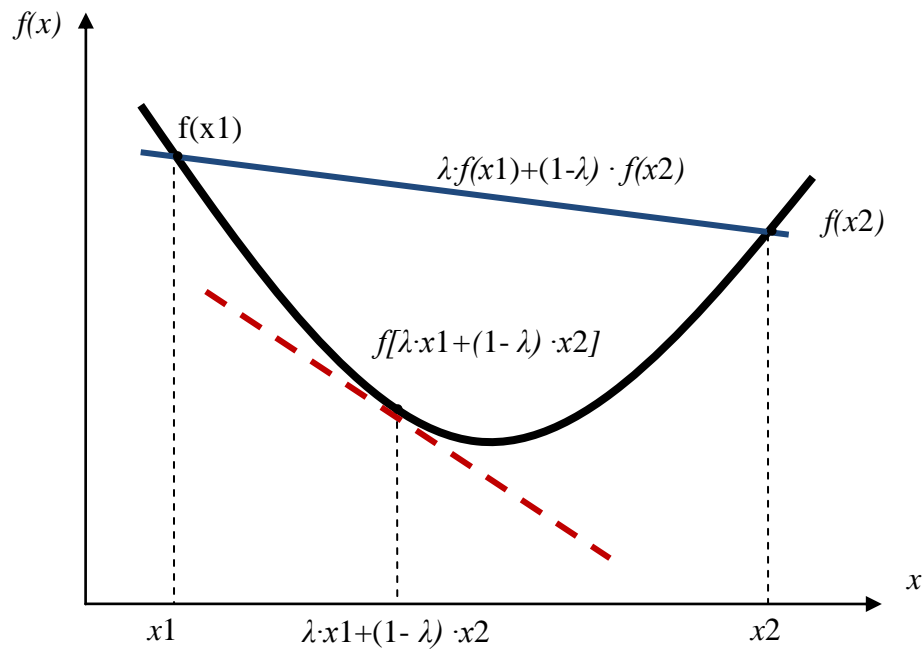


Рисунок 1.5. Геометрическая иллюстрация неравенства Иенсена

Свойства выпуклых функций

1. Хорда, соединяющая две любые точки выпуклой функции, проходит выше графика функций в интервале между этими точками.
2. Производная $f'(x)$ (или тангенс угла наклона касательной) возрастает (или по крайней мере не убывает) при увеличении $x \in [a, b]$.
3. $f''(x) \geq 0, x \in [a, b]$.
4. Для выпуклой функции локальный экстремум всегда является глобальным.
5. Функция $f(x)$ является вогнутой (выпуклой вверх) на интервале $[a, b]$ тогда и только тогда, когда $-f(x)$ есть выпуклая функция на этом интервале.
6. Касательная – в любой точке интервала $[a, b]$ лежит ниже графика функции $f(x)$.

На практике обычно для проверки выпуклости функции (а следовательно и свойства унимодальности) проверяют выполнение свойств 2 и 3.

При выполнении этих свойств функция имеет 1 экстремум, при невыполнении – несколько экстремумов, т.е. функция является мультимодальной.

Свойство унимодальности – весьма важное свойство функций. Многие методы оптимизации применимы только для унимодальных функций. Для многоэкстремальных функций стратегия поиска глобального оптимум сводится к перебору всех локальных экстремумов и выделению среди них глобального.

Определение. Стационарной точкой функции $f(x) \in R^1$ называется точка $x = x^*$, в которой:

$$\left. \frac{df(x)}{dx} \right|_{x=x^*} = 0 \quad (1.10)$$

Существует несколько видов стационарных точек, графическая иллюстрация которых приведена на рисунке 1.6. Из рисунка видно, что только для унимодальных функций стационарная точка определяет глобальный максимум или минимум.

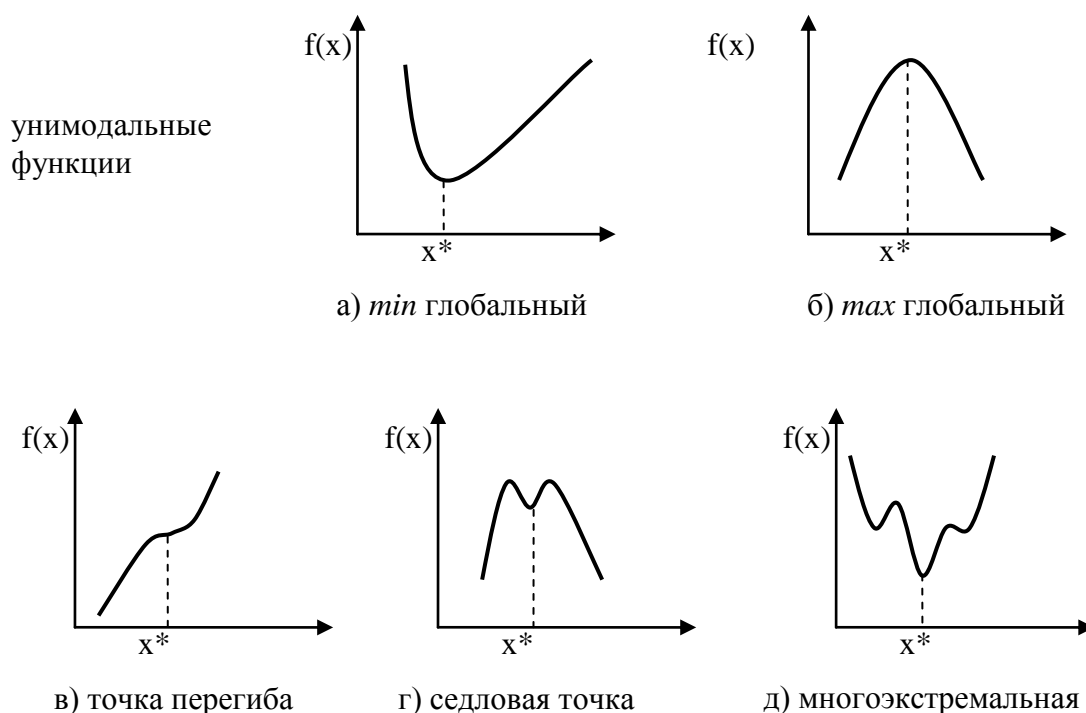


Рисунок 1.6. Типы стационарных точек

Сформулируем необходимые и достаточные условия существования экстремума функции $x \in R^1$.

Теорема (необходимые условия экстремума).

Для того, чтобы дифференцируемая на $[a, b]$ функция $f(x)$ имела в точке $x^* \in [a, b]$ локальный экстремум необходимо, чтобы точка x^* являлась стационарной, т.е.

$$f'(x^*) = 0.$$

Теорема (достаточные условия экстремума).

Пусть в точке x^* $(n-1)$ -е производные функции $f(x)$ обращаются в нуль, а производная n -го порядка отлична от нуля. Тогда

- если n - нечетное $\Rightarrow x^*$ - точка перегиба;
 - если n - четное $\Rightarrow x^*$ - точка локального экстремума.
- кроме того:
- если эта производная $> 0 \rightarrow x^*$ - локальный минимум;
 - если эта производная $< 0 \rightarrow x^*$ - локальный максимум.

Методика нахождения экстремума на интервале

1. Найти стационарную точку x^* функции из решения уравнения $f'(x^*) = 0$.
2. Проверить значение функции в граничных точках интервала и в стационарной точке: $f(a)$, $f(b)$, $f(x^*)$.
3. Проверить знак $f''(x^*)$ для оценки типа экстремума.

Пример 2. Найти экстремум (максимум) функции

$$f(x) = -x^3 + 3 \cdot x^2 + 9 \cdot x + 10, \quad x \in [-2, 4].$$

1) $\frac{df}{dx} = -3 \cdot x^2 + 6 \cdot x + 9 \quad x_1 = 3, x_2 = -1$

2) $f(-2) = 12, \quad f(4) = 30, \quad f(3) = 37, \quad f(-1) = 5 \rightarrow x^* = 3$

3) Оценим вид экстремума

$$\frac{d^2 f}{dx^2} = -6 \cdot x + 6 \Big|_{x=3} = -12 < 0.$$

Следовательно, $x^* = 3$ – глобальный максимум.

$$4) \frac{d^2 f}{dx^2} = -6 \cdot x + 6 \Big|_{x=-1} = 12 > 0.$$

Приведенное на рисунке 1.7 графическое решение совпадает с аналитическим.

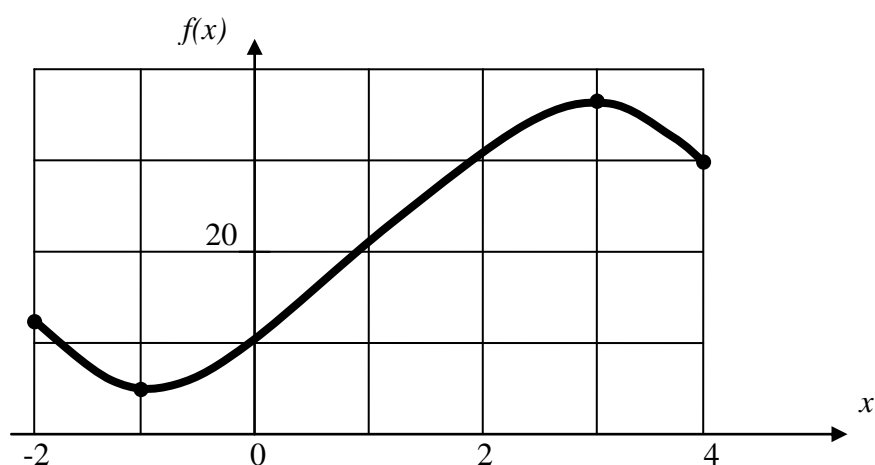


Рисунок 1.7. Графическая оценка экстремума функции

1.6. Характеристики алгоритмов оптимизации

Прежде чем приступить к решению конкретной задачи оптимизации следует выполнить переход от содержательной постановки задачи к формализованной. Следующий этап решения – выбор метода оптимизации. Для целенаправленного выбора алгоритма необходимо сопоставить алгоритмы и методы оптимизации по определенной группе характеристик.

Поэтому весьма важен выбор определенных сравнительных характеристик алгоритмов среди большого количества возможных.

Основными характеристиками численных алгоритмов являются:

- трудоёмкость вычислений;
- сходимость алгоритма;
- устойчивость метода к погрешностям в вычислениях;
- чувствительность метода к параметрам алгоритма;

- эффективность алгоритма.

Большинство методов оптимизации носит итерационный характер. Итерационные методы имеют ряд особенностей. Выделим основные.

1. Вычисления осуществляются по одной и той же итерационной (повторяющейся) формуле $x_{n+1} = \phi(x_n)$. Т.е. каждое последующее приближение к оптимуму определяется через предыдущее.

2. Для начала итерационного процесса необходимо задание начального приближения x_0 .

3. Итерационный процесс бесконечен $\{x_0, x_1, \dots, x_n\}$. Для его останова необходима точность вычисления решения ε и правило (формула) останова.

4. Важнейшая характеристика итерационного процесса количество итераций (шагов) n получения решения с заданной точностью. Чем шагов меньше, тем быстрее может быть получен результат и тем привлекательнее метод.

Именно эти особенности итерационных методов определяют перечисленные выше характеристики алгоритмов оптимизации. Рассмотрим основные из них.

Сходимость алгоритма.

Наиболее важный показатель метода оптимизации. Эта величина определяет, как быстро на каждой итерации уменьшается расстояние до точного значения корня $x^T - x_n$.

Различают сходимость глобальную и асимптотическую. Глобальная сходимость означает, что при любом выборе начальной точки x_0 последовательность $x_n = \phi(x_{n-1})$ сходится к точке, удовлетворяющей необходимым условиям оптимизации.

Иногда для унимодальных функций под глобальной сходимостью понимают попадание $\{x_0, x_1, \dots, x_n\}$ в окрестность стационарной точки x^* .

Под асимптотической сходимостью понимается поведение последовательности $\{x_0, x_1, \dots, x_n\}$ в окрестности предельной точки x^* . Это приводит к тому, что каждому алгоритму приписывается некоторый индекс эффективности, называемый скоростью сходимости r .

$$\frac{|x^m - x_n|}{|x^m - x_{n-1}|^r} = C, \quad (1.11)$$

где C – некоторая конечная ненулевая константа;
показатель степени r и называют скоростью сходимости.

Наиболее распространены случаи линейной и квадратичной сходимости, $r=1$ и $r=2$ соответственно линейная и квадратичная сходимости.

Методы более высокой скорости сходимости встречаются редко.

Замечание 1. При решении практических задач, когда x^* неизвестно, его заменяют последним значением x_{n+1} . Тогда, используя значение заданной точности нахождения оптимума ε получают такую оценку скорости сходимости

$$\alpha = \frac{|\varepsilon|}{|x_n - x_{n+1}|}. \quad (1.12)$$

Скорость сходимости – важнейший показатель итерационных алгоритмов и методов. Однако для оценки эффективности выбранных методов оптимизации можно использовать также такие характеристики:

- время получения результата;
- точность решения;
- чувствительность алгоритма к изменению параметра сходимости.

1.7 Критерии останова итерационных методов

Так как процесс нахождения оптимального решения чаще всего имеет итерационный характер, т.е. последовательность $\{x_0, x_1, \dots, x_n \rightarrow x^T\}$ стремится к точному решению при увеличении количества итераций n .

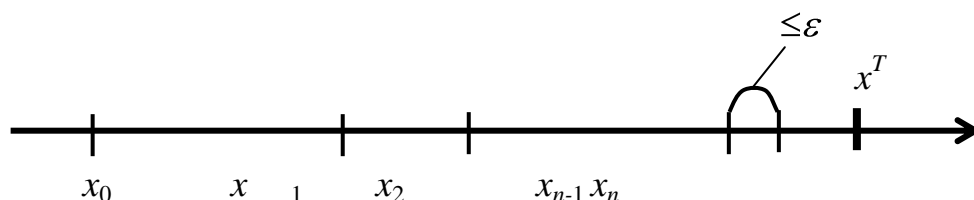


Рисунок 1.8. Иллюстрация сходимости итерационных методов

Весьма важным элементом всех итерационных методов является критерий (правило) останова итерационного процесса. Именно критерий определяет и точность достижения решения, его точность, а соответственно и эффективность метода.

Наиболее распространены следующие критерии останова.

$$1. \left| x_{n+1} - x_n \right| \leq \varepsilon. \quad (1.11a)$$

$$2. \left| f(x_{n+1}) - f(x_n) \right| \leq \varepsilon. \quad (1.11б)$$

$$3. \left| \frac{\partial f(x)}{\partial x} \right|_{x_n} \leq \varepsilon. \quad (1.11в)$$

Следует отметить, что как видно из рисунка 1.8, на n-ой итерации произошел останов итерационного процесса вычисления корня по определенному правилу. При этом точного решения x^T достигнуть не удалось. Итерационная процедура позволила найти приближенное значение решения, но при любом численном решении существует погрешность вычисления $\left| x^T - x_n \right|$. Это особенность и общее свойство всех итерационных методов.

2. МОДЕЛИ И МЕТОДЫ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ

Однопараметрическая оптимизация (поиск экстремумов функций одной переменной) - наиболее простой тип оптимизационных задач. Эти задачи могут быть как самостоятельными, так и частью более сложных задач поиска экстремума функции многих переменных.

Напомним постановку задачи одномерной безусловной оптимизации (1.4).

$$f(x) \rightarrow \min, \quad x \in R^1 \quad .$$

Ограничения отсутствуют, $K = J = 0$, $D \equiv R^1$. Т.е. задача без ограничений с одномерным вектором. Вид $f(x)$ произвольный.

Методы одномерной оптимизации могут быть разделены на три группы.

1. Методы исключения интервалов (сокращение интервала неопределенности).
2. Методы точечного оценивания на основе полиномиальной аппроксимации.
3. Методы с использованием производных.

2.1. Методы исключения интервала неопределенности

Эта группа методов может использоваться как для непрерывных, так и для разрывных и дискретных функций, рисунок 2.1.

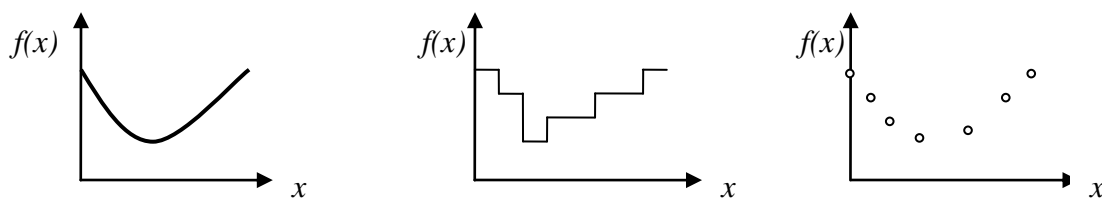


Рисунок 2.1. Типы данных

Методы исключения интервала предъявляют к оптимизируемой функции единственное требование унимодальности. Кроме того необходимо иметь возможность определения значений $f(x)$.

Основная идея интервальных методов основана на теореме исключения интервалов, графическая интерпретация которой приведена на рисунке 2.2.

Теорема исключения интервалов. Пусть функция $f(x)$ унимодальная функция, определенная на $[a, b]$ и x^* - минимум этой функции. Тогда для $\forall x_1 < x_2 \in [a, b]$ выполняется одно из условий:

- а) если $f(x_1) > f(x_2) \rightarrow x^* \in (x_1, b)$;
- б) если $f(x_1) < f(x_2) \rightarrow x^* \in (a, x_2)$;
- в) если $f(x_1) = f(x_2) \rightarrow x^* \in (x_1, x_2)$.

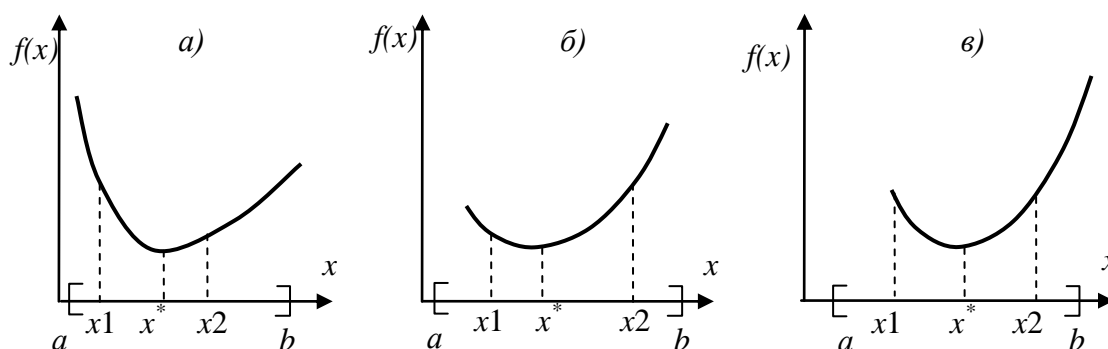


Рисунок 2.2. Геометрическая иллюстрация теоремы интервалов

Данная теорема дает правило последовательного исключения частей исходного интервала, включающее три этапа:

- I. установление первоначального интервала неопределенности;
- II. уменьшение интервала неопределенности по выбранному алгоритму;
- III. оценка экстремума с заданной точностью.

Установление первоначального этапа обычно носит эвристический характер, на основе опыта и интуиции исследователя, либо из физических характеристик процесса.

II этап – уменьшение границ оптимизации – собственно и включает сам метод поиска экстремума. Ниже рассмотрены основные методы одномерной оптимизации.

2.1.1. Метод половинного деления (дихотомии)

Как и все методы одномерного поиска метод дихотомии имеет хорошую геометрическую иллюстрацию, приведенную на рисунке 2.3. Первоначальный интервал $[a, b]$ делится на две равные части и заданная точность нахождения решения ε откладывается в обе стороны от точки деления. Затем интервал поиска сужается в соответствии с теоремой интервалов и процедура деления повторяется. Правила останова итерационной процедуры стандартные и определены формулами (1.14).

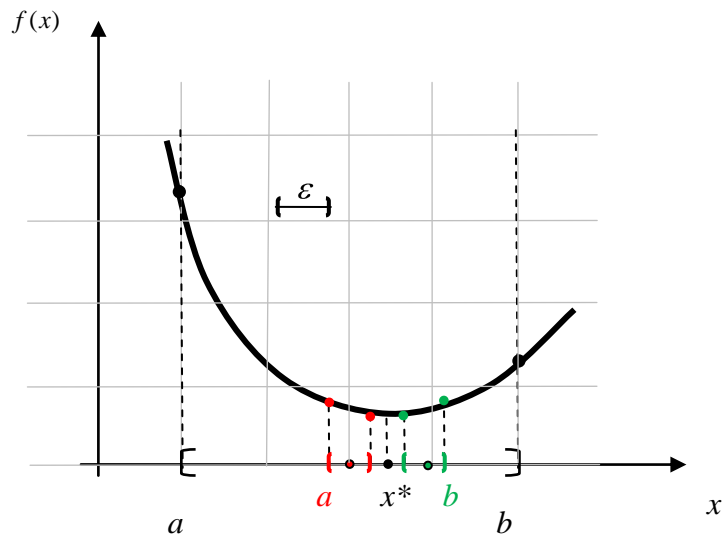


Рисунок 2.3. Геометрическая иллюстрация метода дихотомии

Алгоритмизация метода

Шаг 1. Задание интервала $[a, b]$ и точности вычисления ε .

Шаг 2. Вычисление пробных точек

$$x_1 = \frac{\left(a + b - \frac{\varepsilon}{2}\right)}{2}, \quad f_1 = f(x_1)$$

$$x_2 = \frac{\left(a + b + \frac{\varepsilon}{2}\right)}{2}, \quad f_2 = f(x_2)$$

$$n = n + 1 .$$

Шаг 3. Если $f_2 > f_1 \rightarrow b = x_2$ (a – старое значение);
 Если $f_2 < f_1 \rightarrow a = x_1$ (b – старое значение).

Шаг 4. Если $|b - a| < \varepsilon \rightarrow$ Шаг 5;
 Если $|b - a| > \varepsilon \rightarrow$ Шаг 1.

Шаг 5. $x^* = \min\{x: f(x_1), f(x_2)\}$.

Шаг 6. Печать результатов $x^*, f(x^*), n$.

Замечание 1. Если длина первоначального интервала (начальная стратегия)

$$L_1 = [a, b].$$

То после n итераций интервал (конечная стратегия) сокращается

$$L_n = \frac{1}{2^{n/2}} \cdot L_1 \quad (2.1)$$

Замечание 2. Иногда в Шаге 2 алгоритма не используется величина точности, т.е. $\varepsilon=0$. Логика от этого не меняется, все шаги алгоритма сохраняются, однако увеличивается количество итераций n .

2.1.2. Минимаксная стратегия поиска оптимума

Формализуем стратегию поиска оптимума для интервальных методов. В методе дихотомии величина исключаемого интервала зависит от пробных точек x_1 и x_2 . Т.к. местонахождение x^* неизвестно, то целесообразно уменьшить интервал в одном и том же соотношении и желательно это отношение сделать максимальным.

Подобную стратегию называют минимаксной стратегией поиска. Цель подобной стратегии -минимизация максимально возможного интервала неопределенности, в котором находится минимум (максимум) некоторой функции.

Математически это может быть записано в виде:

$$L_n = \min_{\alpha} \max_q (x_{q+1} - x_{q-1}), \quad (2.2)$$

где α - множество стратегий, удовлетворяющих условию максимизации;

L_n – величина интервала после n шагов.

Рассмотрим минимаксную стратегию для двух точек x_1 и x_2 , $x_1 < x_2$ т.е. $n=2$, рисунок 2.4.

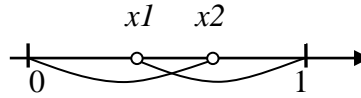


Рисунок 2.4. Схема минимаксной стратегии метода дихотомии

Здесь всего два интервала $(x_2 - 0)$ и $(1 - x_1)$. Тогда в соответствии с критерием (2.2) необходимо сначала найти $L_n = \max[x_2 - 0, 1 - x_1]$.

Из рисунка 2.5 следует, что линия $x_2 = x_1$ ограничивает условие $x_1 < x_2$. Но по условию (2.2) нужно выбрать стратегию $L_n \rightarrow \min$, что таковой стратегией является $L_n = 0.5$. Это выполняется при $x_1 = x_2 = 0.5$. Но для выполнения условия $x_1 < x_2$ вводится некоторое значение $\varepsilon = x_2 - x_1$, при котором:

$$f(x_1) = f(0.5 - \frac{\varepsilon}{2}) \neq f(x_2) = f(0.5 + \frac{\varepsilon}{2}).$$

Следовательно, искомая стратегия определяется как:

$$x_1 = \frac{(1 - \varepsilon)}{2}, \quad x_2 = \frac{(1 + \varepsilon)}{2}, \quad L_n = \frac{(1 + \varepsilon)}{2}.$$

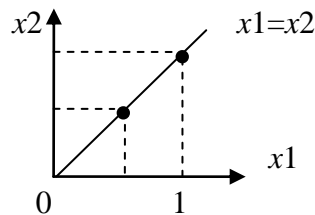


Рисунок 2.5. Схема минимаксной стратегии в плоскости пробных точек

При практическом использовании минимаксной стратегии для двух точек даны такие рекомендации:

- 1) точки x_1 и x_2 следует размещать на одинаковом расстоянии от середины интервала (симметричная схема);

- 2) размещение пробных точек x_1 и x_2 необходимо выполнить так, чтобы отношение длины исключаемого интервала к длине интервала поиска была величиной постоянной.

2.1.3. Метод золотого сечения

На основе минимаксной стратегии более эффективные схемы сокращения интервала неопределенности. Это достигается делением интервала на неравные части.

На рисунке 2.6 графически приведены два первых шага сокращения интервала поиска в методе золотого сечения.

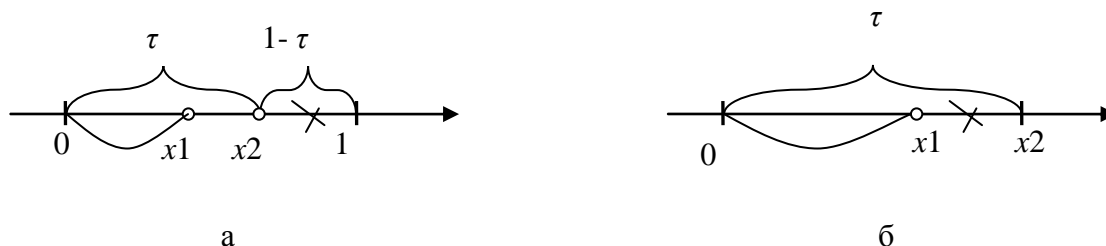


Рисунок 2.6. Геометрическая иллюстрация метода золотого сечения:

а-сокращение интервала неопределенности на начальном шаге;
б-сокращение интервала неопределенности на втором шаге.

Рассмотрим сокращение интервала неопределенности.

а) Пусть задан интервал единичной длины.

Точки x_1 и x_2 выбраны в соответствии с рекомендациями 1 минимаксной стратегии.

Пусть $f(x_2) > f(x_1)$, $x_1 = 0.4$ и $x_2 = 0.6$, исключаем интервал $[x_2, 1]$, т.к. ищем минимум.

б) Рассмотрим интервал $[0, x_2]$.

Выберем вторую точку x_2 в соответствии с рекомендацией 2 минимаксной стратегии и запишем отношение исключаемого интервала ко всей длине интервала, т.е.

$$\frac{1-\tau}{1} = \frac{\tau \cdot (1-\tau)}{\tau} = \frac{\tau^2 \cdot (1-\tau)}{\tau^2} \dots$$

Т.к. в числителе длина исключаемого интервала, то с одной стороны (из числителя) она равна $\tau \cdot (1-\tau)$, а из рисунка $\tau - (1-\tau)$.

Исходя из этого равенства запишем уравнение для нахождения

$$\begin{aligned}\tau \cdot (1 - \tau) &= \tau - (1 - \tau), \\ \tau - \tau^2 &= 2 \cdot \tau - 1, \\ \tau^2 + \tau - 1 &= 0.\end{aligned}$$

Положительный корень этого уравнения $\tau = \frac{(-1 + \sqrt{5})}{2} \approx 0.61803$.

Схема поиска, при котором пробные точки делятся в таком отношении, называются методом золотого сечения.

Замечания

1. Схема сокращения интервала неопределенности приведена на рисунке 2.7.

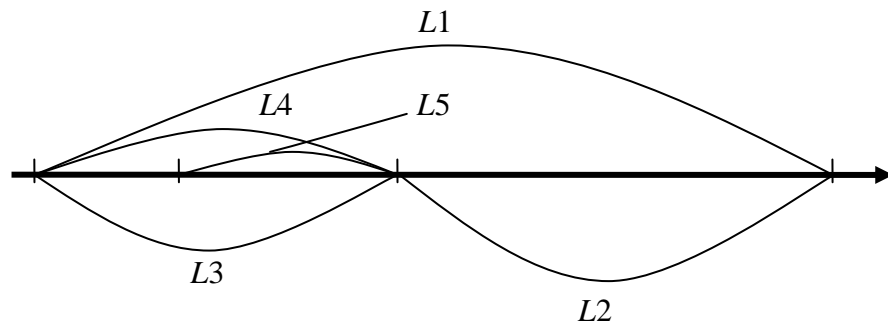


Рисунок 2.7. Геометрическая иллюстрация сокращения интервала
в методе золотого сечения

Длина интервалов от шага к шагу уменьшается в одном и том же отношении τ

$$\frac{L_1}{L_2} = \frac{L_2}{L_3} = \frac{L_3}{L_4} \dots = \tau.$$

2. После n шагов величина интервала равна

$$L_n = \tau^{N-1} \cdot L_1. \quad (2.3)$$

3. Любой интервал $[a, b]$ может быть сведен к единичному интервалу. Если $x \in [a, b]$, то перейдем к новой переменной

$$y = \frac{(x - a)}{(b - a)}.$$

Перепишем пробные точки с учетом введенной величины τ

$$x_1 = a + (b - a) \cdot (1 - \tau),$$

$$x_2 = a + (b - a) \cdot \tau$$

3. В методе золотого сечения отношение интервалов постоянно и равно ≈ 0.61803 . Однако, существует метод Фибоначчи, в котором это отношение меняется как числа последовательности Фибоначчи:

$$F_0 = F_1 = 1, \quad F_k = F_{k-1} + F_{k-2}, \quad \{1, 1, 2, 3, 5, 8, 13, 21 \dots\}.$$

Отношение интервалов в методе равно $\frac{F_{n-1}}{F_{n+1}}$, т.е. $\{1/2, 1/3, 2/5, 3/8, 5/13 \dots\}$.

2.1.3. Метод равномерного поиска

На интервале $[a, b]$ (рисунок 2.8) выбирается k равноотстоящих точек и в них вычисляются значения функции $f(x)$.

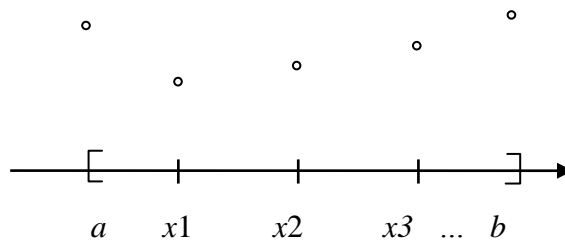


Рисунок 2.8.

Если L_1 - длина интервала $[a, b]$, то длина каждого частичного интервала равна $\frac{L_1}{k}$. Сравнивая значения функции $f(x)$, выбираем такое значение функции, при котором $\bar{x} = \min_k (f(x_k))$. Далее рассматривается интервал $\left[\bar{x} - \frac{L_1}{k}, \bar{x} + \frac{L_1}{k} \right]$. Т.е. длина интервала $L_2 = \frac{2 \cdot L_1}{k}$. Далее интервал L_2 вновь делится на k частей и процесс сокращения интервала продолжается до тех пор, пока не будет выполнено условие останова (1.11).

2.1.4. Сравнительный анализ методов исключения интервалов

В соответствии с введенными выше характеристиками алгоритма оптимизации, сравним методы поиска минимума функции одной переменной по двум характеристикам:

- скорость сходимости (сходимость);
- число шагов k получения минимума с точностью ε (эффективность);.

Пусть показателем скорости сходимости будет величина

$$\alpha(N) = \frac{L_N}{L_1} . \quad (2.4)$$

Тогда для интервальных методов одномерного поиска существуют следующие оценки.

Метод половинного деления

$$\alpha(N) = \frac{L_1 \cdot 0.5^{N/2}}{L_1} = 0.5^{N/2} .$$

Метод золотого сечения

$$\alpha(N) = \frac{\tau^{N-1}}{1} = 0.618^{N-1} .$$

Метод равномерного поиска

$$\alpha(N) = \frac{2 \cdot L_1}{(N+1) \cdot L_1} = \frac{2}{N+1} = \left(\frac{2}{K} \right)^{n-1} .$$

Для нахождения числа шагов N воспользуемся тем же соотношением

$$L_N = L_1 \cdot \alpha(N) .$$

Приравняв правую часть уравнения заданной точности ε , можно получить следующие соотношения.

Метод половинного деления

$$N = \frac{2 \cdot \ln(\varepsilon)}{\ln(0.5)} , \quad \varepsilon = 0.5^{N/2}$$

Следовательно

$$\ln \mathcal{E} = \frac{N}{2} \cdot \ln 0.5.$$

Метод золотого сечения

$$N = 1 + \left\lceil \frac{\ln(\mathcal{E})}{\ln(0.618)} \right\rceil.$$

Метод равномерного поиска

$$N = \frac{2}{\mathcal{E}} - 1, \quad N = 1 + \frac{\ln(\mathcal{E})}{\ln\left(\frac{2}{K}\right)}.$$

В таблице 2.1 приведены количественные характеристики интервальных методов.

Таблица 2.1

Основные характеристики интервальных методов

Метод	$\alpha(N)$			$N(\varepsilon)$		
	$N = 2$	$N = 5$	$N = 10$	$\varepsilon = 10^{-1}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$
Половинного Деления	0.5	0.177	0.031	7	14	26
Золотого сечения	0.618	0.147	0.013	6	11	18
Равномерного поиска		0.0016	$5 \cdot 10^{-7}$		4	6

2.2. Методы точечного оценивания

При рассмотрении методов исключения интервалов к оптимизируемой функции предъявлялось лишь одно требование – унимодальность. Если увеличить количество требований, то можно построить более эффективные алгоритмы поиска.

Потребуем от целевой функции выполнения свойств унимодальности и непрерывности на интервале определения $[a, b]$.

Основная идея этой группы методов связана с возможностью аппроксимации гладкой функции полиномом. Затем полученный полином, как легко дифференцируемая функция, используется для оценки экстремума.

Метод базируется на теореме Вейерштрасса.

Теорема. Любая непрерывная на интервале $[a, b]$ функция может быть аппроксимирована на нем полиномом с любой степенью точности.

Точность аппроксимации достигается либо увеличением степени полинома, либо уменьшением интервала аппроксимации. На практике редко используют полином выше 3-й степени, а вот уменьшение интервала – конструктивный путь.

Метод квадратичной аппроксимации (метод Пауэлла)

Для построения квадратичного полинома необходимо три точки x_1, x_2, x_3 , и знание функции в этих точках $f_1 = f(x_1), f_2 = f(x_2), f_3 = f(x_3)$. Обозначим полином $g(x)$:

$$g(x) = a_0 + a_1 \cdot (x - x_1) + a_2 \cdot (x - x_1) \cdot (x - x_2). \quad (2.5)$$

В формуле (2.5) неизвестными являются коэффициенты интерполяции a_0, a_1, a_2 .

Введем критерий интерполяции $g(x_i) = f(x_i)$.

На основе введенного критерия найдем коэффициенты интерполяции a_0, a_1, a_2 .

$$\left\{ \begin{array}{l} x = x_1, \quad f_1 = g(x_1) = a_0; \\ x = x_2, \quad f_2 = a_0 + a_1 \cdot (x_2 - x_1) \approx g(x_2); \end{array} \right.$$

$$x = x_3, \quad f_3 = a_0 + a_1 \cdot (x_3 - x_1) + a_2 \cdot (x_3 - x_1) \cdot (x_3 - x_2) \approx g(x_3).$$

Следовательно

$$a_0 = f_1,$$

$$a_1 = \frac{f_2 - f_1}{x_2 - x_1},$$

$$a_2 = \frac{1}{x_3 - x_2} \cdot \left[\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right].$$

Для оценки точки оптимума функции $f(x)$ найдем стационарную точку для аппроксимированной функции $g(x)$ из условия $g'(x) = 0$.

$$g'(x) = a_1 + a_2 \cdot (x - x_2) + a_2 \cdot (x - x_1) = 0.$$

Отсюда уравнение для нахождения стационарной точки

$$a_1 + a_2 \cdot (2 \cdot x - x_1 - x_2) = 0.$$

Отсюда стационарная точка по методу Пауэлла определяется следующим соотношением

$$x_{\text{ст}}^* = -\frac{a_1}{2 \cdot a_2} + \frac{x_1 + x_2}{2}. \quad (2.6)$$

Пример. Найти экстремум методом Пауэлла для функции

$$f(x) = 2 \cdot x^2 + \frac{16}{x}, \quad x \in [1, 5].$$

Пусть $x_1 = 1$, $x_2 = 5$, $x_3 = 3$.

Значения функции в этих точках равно

$$f_1 = 18, \quad f_2 = 23.33, \quad f_3 = 53.2.$$

Запишем квадратичную аппроксимацию

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

Отсюда

$$a_1 = \frac{f_2 - f_1}{x_2 - x_1}, \quad a_2 = \frac{1}{x_3 - x_2} \cdot \left[\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right].$$

$$a_1 = 2.665, \quad a_2 = 3.067.$$

И значение стационарной точки

$$x^* = \frac{3+1}{2} - \frac{2.665}{2 \times 3.067} = 1.565.$$

Значение оптимума

$$f(1.565) = 15.122.$$

Оценим точное значение стационарной точки и оптимум функции
 $f'(x) = 0.$

$$4 \cdot x - \frac{16}{x^2} = 0, \quad x = \sqrt[3]{4} = 1.5874.$$

Таким образом, точное значение стационарной точки

$$x^* = 1.5874, \quad f(1.5874) = 15.119.$$

Оценим тип стационарной точки.

$$f''(x) = 4 + \frac{32}{x^3}, \quad f''(x) > 0, \text{ при } x > 0.$$

Полученная стационарная точка является минимумом.

Этот метод, разработанный Пауэллом, лежит в основе схемы алгоритмизации, начальными условиями которой являются начальная точка x_1 и Δx – величина шага по оси ОХ, ε – точность вычисления оптимума.

Алгоритмизация метода

Шаг 1. Вычислить $x_2 = x_1 + \Delta x$.

Шаг 2. Вычислить $f(x_1), f(x_2)$.

Шаг 3. Если $f(x_1) > f(x_2)$, то $x_3 = x_1 + 2 \cdot \Delta x$,
 иначе $x_3 = x_1 - 2 \cdot \Delta x$.

Шаг 4. Вычислить $f(x_3)$ и найти минимальное значение
 $F_{\min} = \{f_1, f_2, f_3\}.$

Шаг 5. По точкам x_1, x_2, x_3 строим полином по формуле (2.5) и находим стационарную точку x^* по формуле (2.6).

Шаг 6. Проверка на окончание итерационного процесса по формулам (1.11)

Если

$$|F_{\min} - F(x_{cm}^*)| \leq \varepsilon$$

или

$$|x_{\min} - x_{cm}^*| \leq \varepsilon,$$

то поиск закончен, иначе – шаг 7.

Шаг 7. Выбрать наилучшую точку x_{\min} или x^* (по минимуму $f(x)$) и две точки по обе стороны от неё

$$x_2 = \{x^*, x_{\min} : \min(f(x^*), f_{\min})\}$$

$$x_1 = x_2 - \Delta x, \quad x_3 = x_2 + \Delta x,$$

Вычислить значения функции в этих точках f_1, f_2, f_3 .

Перейти к шагу 5.

Замечания.

1. При первой реализации шага 5 точка x^* может выйти за пределы $[x_1 - x_3]$. Поэтому необходимо провести проверку $x^* \in [x_1, x_3]$. И в случае неудачи проверки заменить её любой точкой x_1, x_2 .

2. Если шаг 3 приведет к выходу из интервала $[a, b]$, то задать точку $x_3 = x_1 + 2 \cdot \Delta x$.

2.3. Методы одномерного поиска с использованием производных

Если к требованиям унимодальности и непрерывности целевой функции добавить требование дифференцируемости, то эффективность поисковых процедур может существенно повыситься.

Известно, что необходимым условием существования экстремума является наличие стационарной точки $f'(x)|_{x=x^*} = 0$.

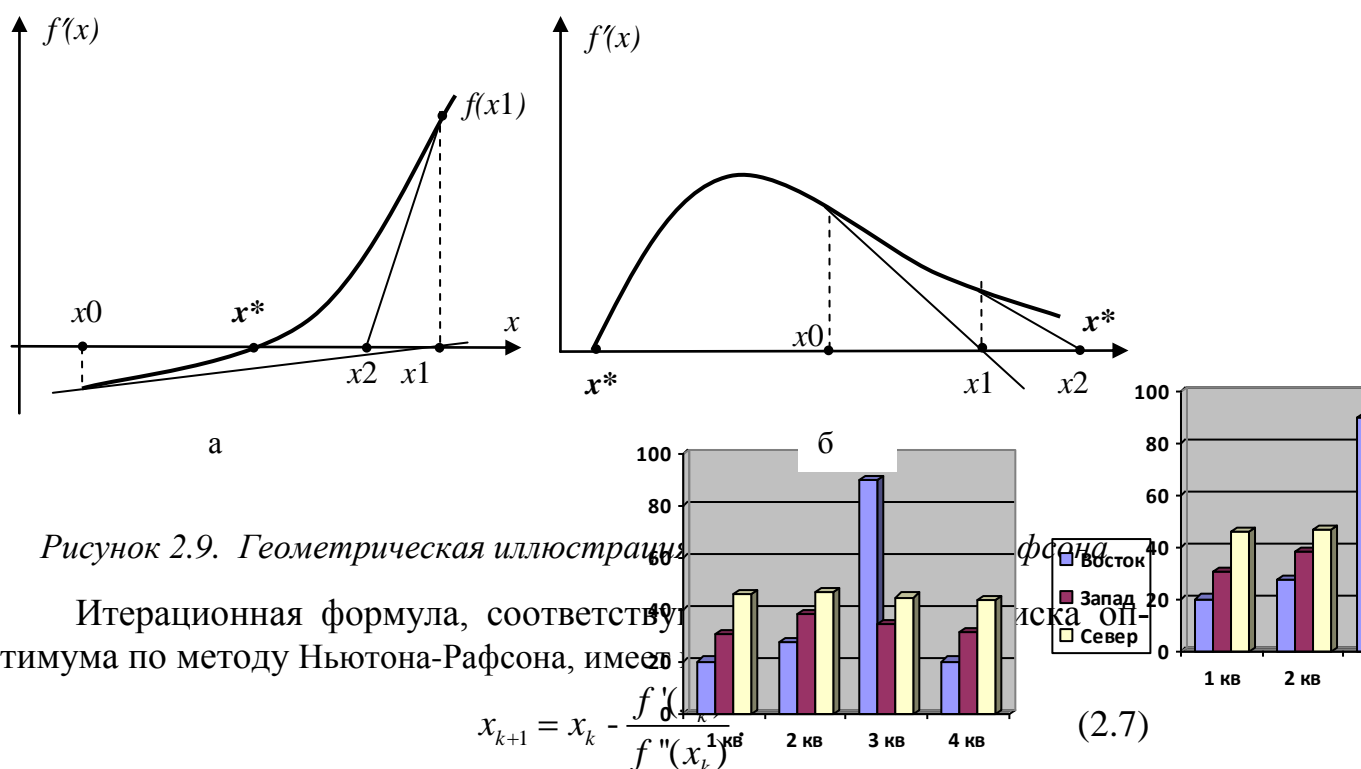
Если вид уравнения $f'(x) = 0$ является трудноразрешимым непосредственным образом, то для поиска стационарной точки могут быть использованы поисковые процедуры последовательного приближения к решению уравнения $f'(x) = 0$. Прежде всего это методы решения,

нелинейных уравнения: половинного деления, дихотомии, Ньютона, хорд и т.д.

2.3.1. Метод Ньютона – Рафсона

В рамках метода предполагается, что функция $f(x)$ дважды дифференцируема. После выбора начального приближения x_0 строится касательная в точке $f'(x_0)$ - рисунок 2.9. Из решения уравнения $f'(x_1) = 0$ найдем стационарную точку. Геометрически это соответствует точке пересечения касательной $f'(x_0)$ с осью ОХ.

Далее строим касательную к функции $f(x)$ в точке $f'(x_1)$ и находим стационарную точку x_2 и т.д.



Итерационная формула, соответствующая методу Ньютона-Рафсона, имеет вид

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \quad (2.7)$$

В основе формулы лежит разложение в ряд Тейлора $f'(x_{k+1}) = f'(x_k + h)$ и отбрасывание нелинейных членов разложения.

Критерии останова метода могут быть стандартные (1.11). Но с учетом, что находится экстремум функции $f(x)$ по значению производной $f'(x)$, целесообразно использовать станов по правилу

$$|f'(x_k)| \leq \varepsilon.$$

Замечания.

1. Метод Ньютона может расходиться (рисунок 2.9- б) при неудачном выборе начального приближения x_0 . Метод хорошо работает в окрестности стационарной точки. Поэтому рекомендуется сузить интервал неопределенности, а затем использовать метод Ньютона (комбинация методов).

2. Удачный выбор начальной точки x_0 обеспечивает следующее условие $f(x_0) \cdot f''(x_0) > 0$.

2.4. Сравнение методов одномерного поиска оптимума

Успех использования того или иного метода одномерной оптимизации зависит от следующих моментов:

- вида исходной целевой функции;
- целей решения конкретной задачи.

Так, методы оптимизации, основанные на полиномиальной аппроксимации и вычислении производных более эффективны, чем методы исключения интервалов. Но это выполняется при условии, что целевая функция достаточно гладкая и непрерывная.

Цели решения задачи накладывают ограничения на основные характеристики методов:

- время получения решения;
- точность полученного экстремума;
- чувствительность алгоритма к начальным условиям.

Поэтому приоритет метода может определяться критичностью к той или иной характеристике. Так, например, тот же точечный метод обеспечивает высокую точность для гладких функций. Однако на быстроизменяющихся функциях работает весьма медленно и, следовательно, не удовлетворяет показателю времени получения решения. Методы с использованием производных работают только при аналитически заданных функциях, для которых возможно оценить производную без дополнительных погрешностей. Но в технических приложениях целевая функция чаще задается таблично, как результат эксперимента и измерения некоторого показателя. В этом случае возможно использование только интервальных методов. И, наконец, объединение некоторых методов может принести ощутимое улучшение показателей оценки экстремума.

3. МЕТОДЫ БЕЗУСЛОВНОЙ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ

В общем виде задачи безусловной оптимизации имеет вид

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

$$x \in D \subset R^n.$$

Т.е. ограничения I рода ($h_k(x) = 0$) и II рода ($g_j(x) \geq 0$) отсутствуют или являются пустыми множествами.

Методы многомерной безусловной оптимизации аналогичны методам одномерной оптимизации и определены необходимыми и достаточными условиями экстремума. В таблице 3.1 приведено сопоставление условий существования экстремума

Таблица 3.1

Условия существования экстремума

условия экстремума	R^1	R^n
необходимые	стационарная точка <u>$f'(x^*) = 0$</u>	вектор-градиент <u>$\nabla f(x_0) = 0$</u>
необходимые и достаточные	1. стационарная точка $f'(x^*) = 0$ 2. $f''(x^*) > 0$	1. вектор-градиент $\nabla f(x_0) = 0$ 2. матрица Гессе - положительно определена

Градиент – это вектор, проекции которого на оси координат являются частными производными.

Введем вектор-градиент:

$$\nabla f(\bar{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T, \quad (3.1)$$

и вещественную матрицу Гессе (гессиан):

$$H(x) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right], \quad i, j \in [1, n] \quad (3.2)$$

Например, для $f(\bar{x}) \in R^2$

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

Матрица Гессе $H(x)$ - симметрична.

Теорема 1. Для того чтобы дифференцируемая на множестве $D \in R^n$ функция $f(\bar{x})$ имела в точке $\bar{x} \in D$ локальный экстремум необходимо, чтобы \bar{x} была стационарной точкой, т.е.

$$\nabla f(\bar{x}) = 0 \quad (3.3)$$

Теорема 2. Критерий Сильвестра.

Если $f(x)$ – дважды дифференцируемая на $D \in R^n$ функция и ее гессиан положительно определен при всех $x \in D$, то функция $f(x)$ является выпуклой на множестве D .

Критерии определенности матрицы H(x)

Положительная определенность матрицы означает:

- все диагональные элементы положительны;
- определители всех главных миноров матрицы положительны.

Положительная полуопределённость матрицы означает:

- все диагональные элементы матрицы ≥ 0 ;
- определители всех главных миноров матрицы ≥ 0 .

Пример. Оценить определенность гессиана для функции

$$f(x_1, x_2) = 2 \cdot x_1^2 + x_2^2 + \sin(x_1 + x_2).$$

Найдем производные первого и второго порядка.

$$f'(x_2) = 2 \cdot x_2 + \cos(x_1 + x_2),$$

$$f'(x_1) = 4 \cdot x_1 + \cos(x_1 + x_2).$$

$$\frac{\partial^2 f}{\partial x_2^2} = 2 - \sin(x_1 + x_2),$$

$$\frac{\partial^2 f}{\partial x_1^2} = 4 - \sin(x_1 + x_2),$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = -\sin(x_1 + x_2).$$

Запишем матрицу Гессе

$$H(x) = \begin{bmatrix} 4 - \sin(x_1 + x_2) & -\sin(x_1 + x_2) \\ -\sin(x_1 + x_2) & 2 - \sin(x_1 + x_2) \end{bmatrix}.$$

Главные миноры

$$\Delta_1 = 4 - \sin(x_1 + x_2) > 0,$$

$$\Delta_2 = 8 - 6 \cdot \sin(x_1 + x_2) > 0.$$

Следовательно, гессиан положительно определен и функция $f(x)$ – выпуклая.

Теорема 3. Необходимые и достаточные условия существования минимума многомерной функции $f(x) \in R^n$

- существует стационарная точка, т.е. вектор-градиент $\nabla f(x_0) = 0$;
- матрица Гессе положительно определена.

Методы поиска экстремума многомерной функции не имеют столь наглядной геометрической иллюстрации, как в одномерном случае. Имеет геометрическую иллюстрацию только двумерный случай. Рассмотрим графическое решение для следующего примера.

Пример.

Дана сепарабельная функция

$$z(x, y) = (x - 1)^2 + 4 \cdot (y - 2)^2.$$

Вид функции приведен на рисунке 3.1, а линии уровня на рисунке 3.2.

Найти экстремум функции при начальных условиях

$$x_0, y_0 = \{2.5; 3\}.$$

Отметим, что точное решение

$$(x, y)^T = (1, 2), \quad z(x^T, x^T) = 0).$$

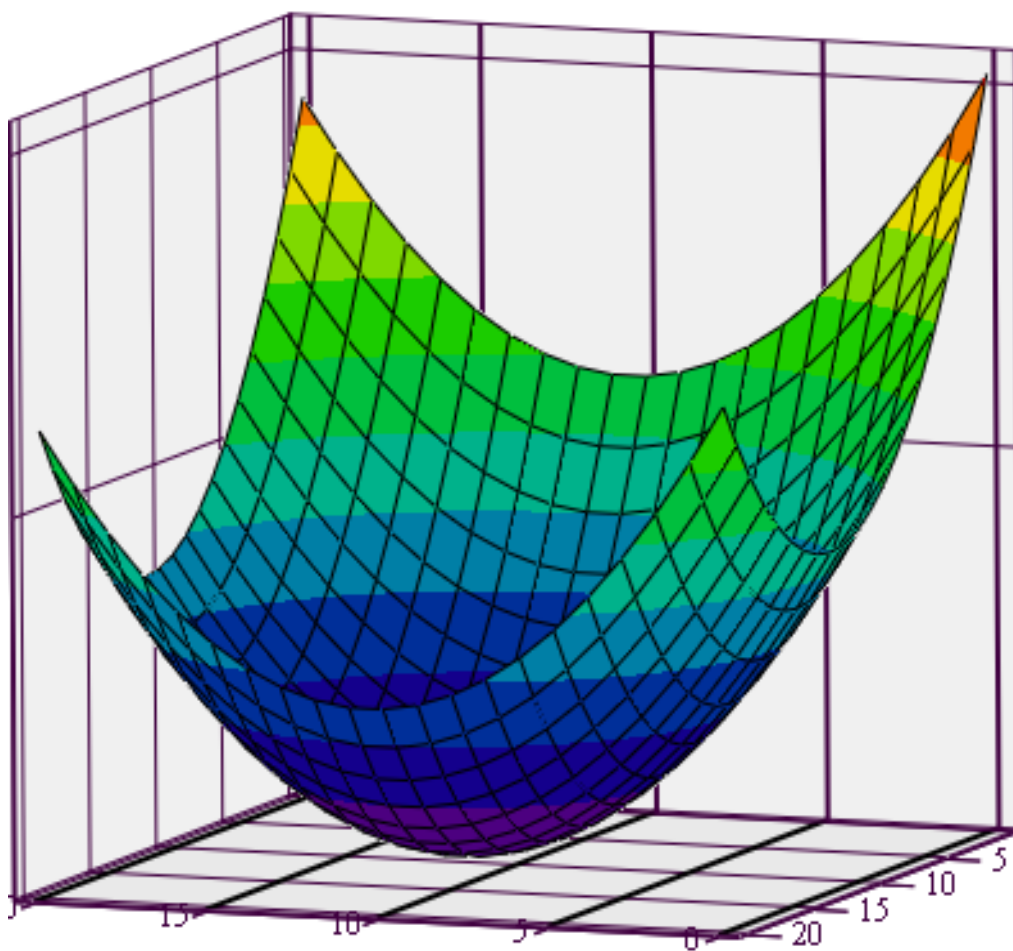


Рисунок 3.1.

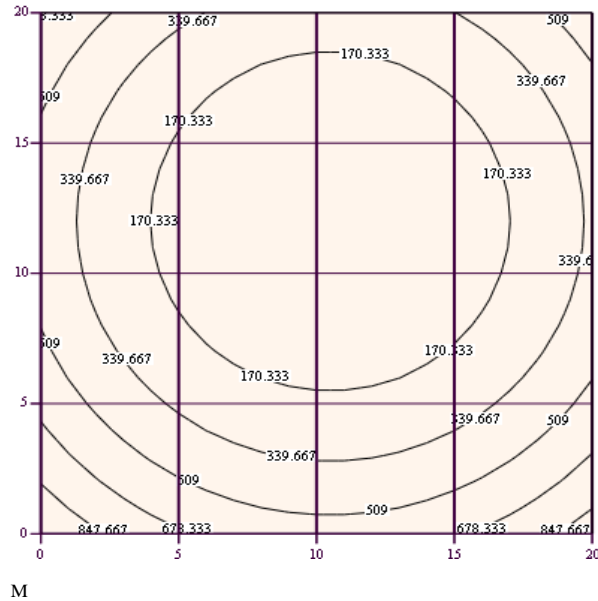


Рисунок 3.2.

Рассмотрим графическое решение, представленное на рисунке 3.3.

$$z(x, y) = (x - 1)^2 + 4 \cdot (y - 2)^2.$$

1) При заданных начальных условиях фиксируем одну координату $y_0=3$. Это равносильно проведению плоскостей Π_0 , параллельно плоскости $(Z, 0, X)$.

Линия пересечения плоскости Π_0 с поверхностью $z(x, y)$ определяется уравнением:

$$z = (x - 1)^2 + 4 \cdot (3 - 2)^2 = (x - 1)^2 + 4.$$

Полученное уравнение определяет линию пересечения плоскости Π_0 с поверхностью $z(x, y)$. Экстремальная кривая является параболой, стационарная точка которой

$$z'(x) = 2 \cdot (x - 1) = 0, \quad x^1 = 1.$$

2) Фиксируем теперь $x^1 = 1$ т.е. проводим плоскость Π_1 параллельную $(Z, 0, Y)$. В данной плоскости экстремальная кривая:

$$z = 4 \cdot (y - 2)^2 \rightarrow z' = 8(y - 2) = 0 \rightarrow y^2 = 2.$$

Итак, за два шага найдено оптимальное значение. Это объясняется тем, что оси эллипсов (линии уровня) параллельны осям координат.

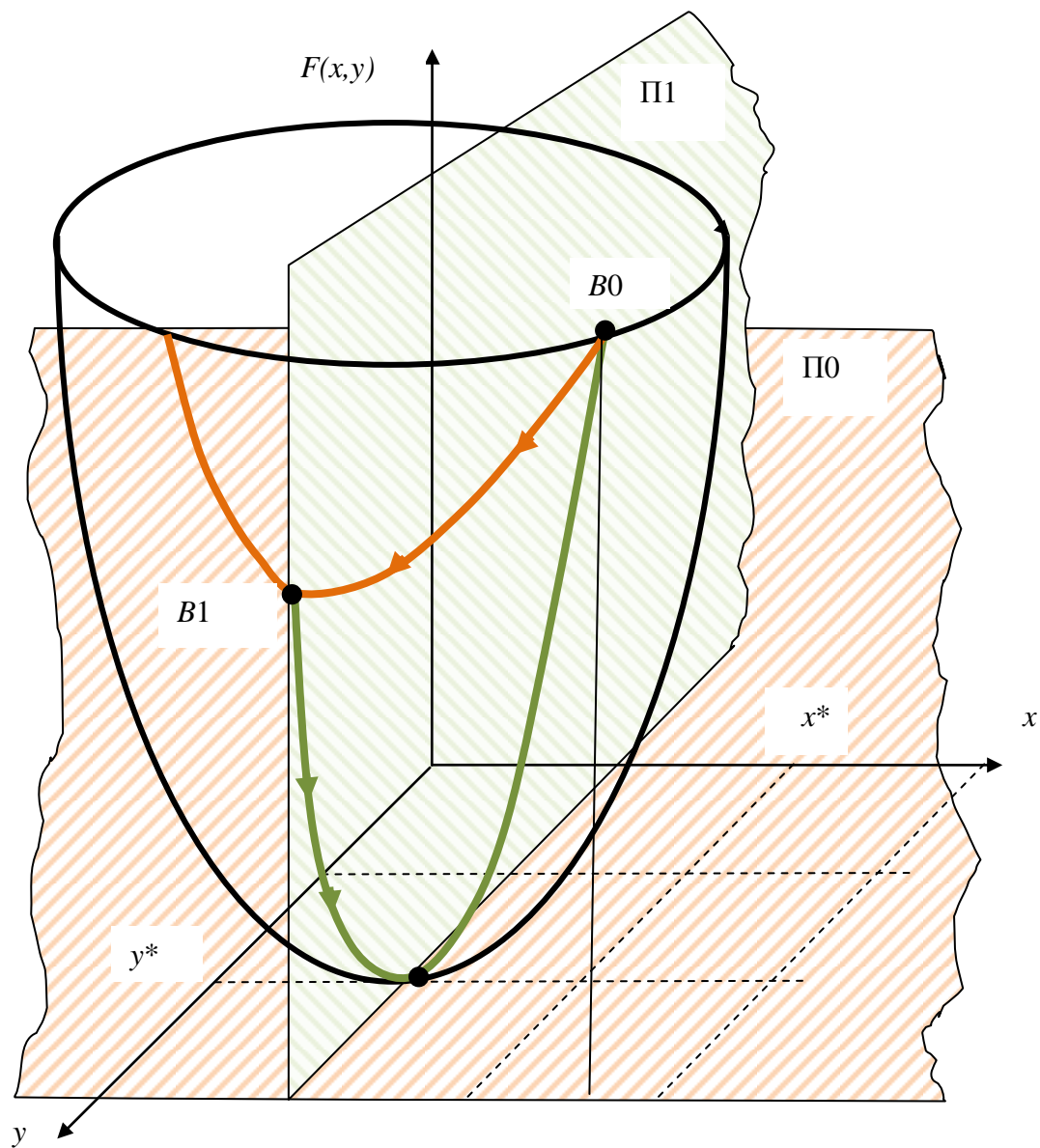


Рисунок 3.3. Графическое решение

3.1. Метод покоординатного спуска

Формализуем рассмотренную графическую процедуру поиска, которая носит название метода покоординатного спуска

Пусть $z = f(x_1, x_2)$, $\bar{x}^0 = \{x_1^0, x_2^0\} = B_0$.

Процесс оптимизации проходит следующим образом.

Фиксируем переменную x_2 в точке x_2^0 . Тогда $z = f(x_1, x_2^0)$ является функцией одной переменной x_1 .

Решаем задачу одномерной оптимизации по переменной x_1 и попадаем в точку $B_1 = (x_1^1, x_2^1)$

$$B_0 = x_1^0, x_2^0 \rightarrow B_1 = x_1^1, x_2^1.$$

Далее фиксируем точку x_1^1 и спускаемся по координате $x_2 \rightarrow B_2(x_1^2, x_2^2)$ и т.д.

Получаем сходящуюся последовательность

$$f(B_0) > f(B_1) > f(B_2) > \dots$$

Замечания.

1. Метод координатного спуска не применим в случае наличия изломов в линиях уровня, т.е. поверхностей с оврагами и гребнями.

2. Достоинством метода является возможность использования простых алгоритмов одномерной оптимизации

3. Метод использует только необходимое условие экстремума. Это может привести к локальному экстремуму.

Отметим, что для приведенной выше сепарабельной целевой функции $z(x, y) = (x-1)^2 + 4 \cdot (y-2)^2$ линии уровня имели оси, параллельные главным осям координат. Это облегчает поиск экстремума и сокращает количество итераций.

Для квадратичной несепарабельной функции оси эллипсов (линий уровня) расположены произвольно по отношению к координатным осям. На рисунке 3.4а) приведена целевая функция $f(x_1, x_2) = x_1^2 + 2 \cdot x_2^2 + x_1 \cdot x_2 - 7 \cdot x_1 - 7 \cdot x_2$, а на рисунке 3.4б) - линии уровня.

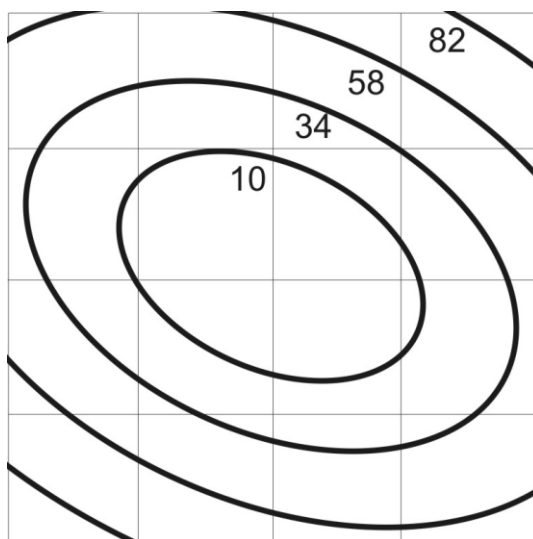
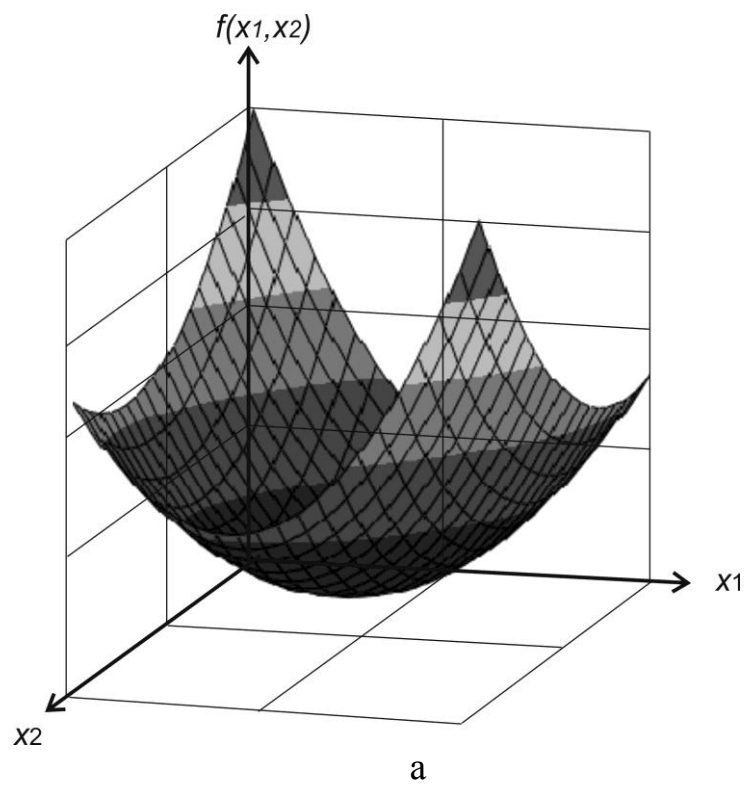


Рисунок 3.4. Целевая функция $f(x_1, x_2) = x_1^2 + 2 \cdot x_2^2 + x_1 \cdot x_2 - 7 \cdot x_1 - 7 \cdot x_2$

3.2. Методы поиска безусловного экстремума

Пусть в R^n задан вектор \bar{x}^0 . Требуется построить такую последовательность векторов $\bar{x}^0, \bar{x}^1, \dots, \bar{x}^k \in R^n$, чтобы выполнялось условие:

$$f(\bar{x}^0) > f(\bar{x}^1) > \dots > f(\bar{x}^k) > \dots \quad (3.4)$$

Методы поиска экстремума, обеспечивающие выполнение условия (3.4), разделяются на ряд этапов.

- I. Выбор начальной точки \bar{x}^0 . Так как ограничений нет, следовательно, значения вектора \bar{x}^0 любые, но целесообразнее выбрать \bar{x}^0 близкое к минимуму.
- II. Выбор направления движения в сторону экстремума.
- III. Выбор шага перемещения.

При любом методе спуска (поиска) последовательность $\{\bar{x}^k\}$, $k = 0, 1, \dots$, подчиняется условию:

$$\bar{x}^{k+1} = \bar{x}^k + \alpha_k \cdot \bar{d}_k, \quad k = 0, 1, \dots, \quad (3.5)$$

где d_k - направление спуска.

$\|\alpha_k d_k\|$ - величина шага.

Если d_k нормализовано, т.е. $\|d_k\| = 1$, то величина шага равна $|\alpha_k|$.

Компоненты формулы (3.5) представлены на рисунке 3.5.

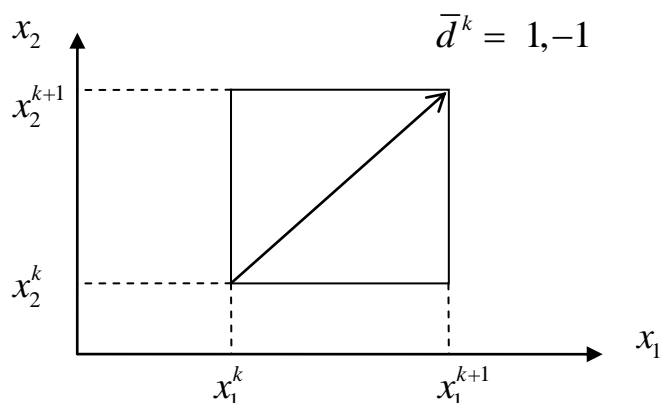


Рисунок 3.5. Направление движения и величина шага

Т.к. в задачах без ограничений все направления возможны, то ограничение определяется лишь смыслом поиска (например, поиск минимума). Таким образом, возможное направление определяется формулой

$$f(\bar{x}^k + \alpha_k \bar{d}_k) < f(\bar{x}^k) .$$

Разложим в ряд Тейлора $f(\bar{x}^k + \alpha_k \bar{d}_k)$, ограничиваясь линейными членами разложения

$$f(\bar{x}^k + \alpha_k \bar{d}_k) \approx f(\bar{x}^k) + \alpha_k \cdot \nabla f(\bar{x}^k)^T \cdot \bar{d}_k + \dots$$

Отсюда, чтобы направление было возможным (приемлемым) требуется, чтобы

$$\nabla f(\bar{x}^k)^T \cdot \bar{d} < 0. \quad (3.6)$$

Механизм поиска экстремума в значительной степени определяется видом целевой функции и дополнительной информацией о её свойствах. В зависимости от этого различают следующие группы методов безусловного многомерного поиска.

1. Прямые методы поиска, использующие только значения.
2. Градиентные методы, использующие значения первых производных целевой функции.
3. Методы второго порядка, использующие значения первых и вторых производных целевой функции.

3.3. Методы прямого поиска

Методы прямого поиска не используют никакой информации кроме значений целевой функции.

К этой группе относятся следующие методы.

1. Метод покоординатного спуска.
2. Метод Нелдора-Мида.
3. Метод Хука-Дживса.

Самая простая идея прямого поиска заключается в сканировании возможного пространства переменных $\bar{x} = \{x_1, x_2, \dots, x_n\}$.

Пусть D - двумерная область изменения переменных $\bar{x} = \{x_1, x_2\}$, ε – заданная точность нахождения экстремума. В результате сканирова-

ния вычисляем значения целевой функции $f_1(\bar{x}^1), f_2(\bar{x}^2), \dots, f_n(\bar{x}^n)$, $n = \frac{D}{\mathcal{E}} + 1$. Затем находим экстремальное значение

функции $f(\bar{x}^*) = \min\{f_i(\bar{x}^i)\}$. Здесь верхний индекс означает номер шага сканирования.

Для направленного сканирования существуют два способа обхода области D. Для случая двух переменных они приведены на рисунке 3.6.



Рисунок 3.6. Способы сканирования области двух переменных

а- строчная развертка; б- спираль.

Также проста идея с выбором базовой точки и оценивания значений целевой функции в окружении базовой точки, например в квадратном образце.

Методы прямого поиска малоэффективны. Большая трудоемкость методов связана с большим количеством вычислений целевой функции.

3.3.1. Метод Хука Дживса

Метод Хука-Дживса это комбинация исследующего поиска с циклическим изменением переменных и ускоряющего поиска по образцу. Схематично стратегия поиска изображена на рисунке 3.7.

Данный метод является аналогом метода циклического покоординатного спуска с изменяющимся шагом.

В 1961 году Хук и Дживс придумали эвристический метод n - мерного прямого поиска, но метод является весьма эффективным и оригинальным и популярен среди методов прямого поиска до сих пор. Суть метода такова.

1. Исследующий поиск

В начале для выбора базовой точки исследуется окрестность $\bar{x} = \{x_1, x_2, \dots, x_n\}$ по различным координатам вектора x (может быть с

различным шагом). После того, как выбрано приемлемое направление (фиксируется новая базовая точка) осуществляется переход к этапу 2. Если значение целевой функции $f(x)$ увеличилось, то необходимо вернуться в предыдущую точку и сделать половинный шаг.

2. Установление конфигураций (поиск по образцу)

Из базовой точки в приемлемом направлении делаешься большой шаг, и шаг увеличивается в этом направлении до тех пор пока целевая функция уменьшается. Когда функция перестанет уменьшаться, то от принятой конфигурации необходимо отказываются и размер шага уменьшить. Производится новое обследование окрестности. Таким образом, делается попытка поиска оврага целевой функции, а затем движение по этому оврагу. Поиск по образцу – это движение вдоль прямой, соединяющей две базовые точки.

Введем обозначения:

x^k - текущая базовая точка.

x^{k-1} - предыдущая базовая точка.

x^{k+1} - новая базовая точка.

x_n^{k+1} - точка, построенная при движении по образцу.

$$x_n^{k+1} = x^k + (x^k - x^{k-1})$$

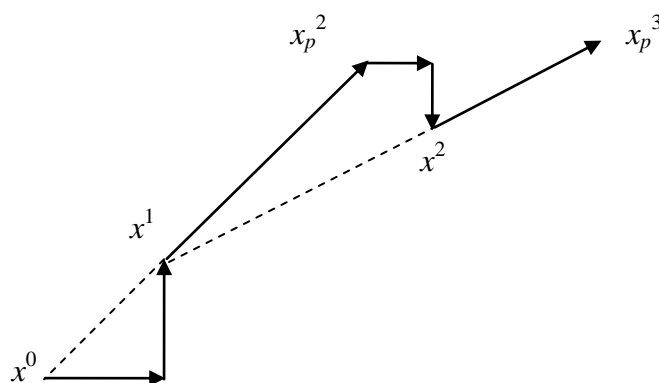


Рисунок 3.7. Геометрическая иллюстрация метода Хука-Дживса

Как только движение по образцу не приводит к уменьшению целевой функции, точка x_p^{k+1} фиксируется в качестве временной базовой точки и вновь проводится исследующий поиск.

Если в результате получается точка с меньшим значением целевой функции, чем в точке x^k , то она рассматривается как новая базовая точка x^{k+1} . С другой стороны, если исследующий поиск неудачен, необходимо вернуться в точку x^k и провести исследующий поиск с целью выявления нового направления минимизации. В конечном итоге возникает ситуация, когда такой поиск не приводит к успеху. В этом случае требуется уменьшить величину шага и возобновить исследующий поиск. Поиск завершается, если величина шага меньше заданной точности вычисления экстремума ε .

Алгоритмизация метода Хука-Дживса

Шаг 1. Задать начальные условия - \bar{x}^0 , шаг перемещения - Δx , точность вычисления экстремума - ε , коэффициент шага убывания - $\beta > 1$, число итераций $k = 0$.

Шаг 2. Провести исследовательский поиск

$$k = k + 1.$$

$$x_1^k = x_1^{k-1} \pm \Delta x$$

$$x_2^k = x_2^{k-1} \pm \Delta x.$$

в зависимости от убывания $f(x_i \pm \Delta x)$, $i = \overline{1, 2}$.

Шаг 3. Оценить произошло ли убывание целевой функции.

если $f(\bar{x}^{k+1}) < f(\bar{x}^k) \rightarrow$ Шаг 5;

если $f(\bar{x}^{k+1}) > f(\bar{x}^k) \rightarrow$ Шаг 4.

Шаг 4. Окончание поиска

если $\|\Delta x\| < \varepsilon \rightarrow$ выход из процедуры,

печать результатов $\bar{x}^k, f(\bar{x}^k)$.

если $\|\Delta x\| > \varepsilon, \Delta_i = \frac{\Delta_i}{\beta}, i = \overline{1, 2} \rightarrow$ переход на Шаг 2.

Шаг 5. Провести поиск по образцу

$$x_p^{k+1} = x^k + (x^k - x^{k-1}).$$

Шаг 6. Оценить убывание целевой функции.

если $f(\bar{x}_p^{k+1}) < f(\bar{x}^k)$, $x^{k+1} = x_p^{k+1} \rightarrow$ Шаг 5.

если $f(\bar{x}_p^{k+1}) > f(\bar{x}^k) \rightarrow$ Шаг 4.

Замечание. Достоинством метода Хука-Дживса является то, что он позволяет восстанавливать направление движение вдоль оврага, когда из-за искривления оврага установленная конфигурация теряется.

Пример.

Найти методом Хука-Дживса минимум функции

$$f(\bar{x}) = 8 \cdot x_1 + 4 \cdot x_1 \cdot x_2 + 5 \cdot x_2^2$$

с точностью $\varepsilon = 10^{-1}$, при заданных начальных условиях

$$\bar{x}^0 = \begin{pmatrix} -4 \\ -4 \end{pmatrix}, \quad \Delta x = 1, \quad \alpha = 2.$$

1. Исследовательский поиск

$$f\left(\begin{pmatrix} -4 \\ -4 \end{pmatrix}\right) = 272.$$

Фиксируем $x_2^0 = -4$, $x_1^1 = -4 + 1 = -3$

$$f\left[\begin{pmatrix} -3 \\ -4 \end{pmatrix}\right] = 200 < f(x^0) = 272 \rightarrow \text{успех}$$

Фиксируем $x_1^1 = -3$, $x_2^1 = -4 + 1 = -3$

$$f\left[\begin{pmatrix} -3 \\ -3 \end{pmatrix}\right] = 153 < f(x^0) = 272 \rightarrow \text{успех}$$

2. Поиск по образцу

$$\bar{x}_p^{-2} = \bar{x}^{-1} + (\bar{x}^{-1} - \bar{x}^{-0}) = \begin{bmatrix} -3 + (-3 + 4) \\ -3 + (-3 + 4) \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$$

$$f(\bar{x}_p^{-2}) = 68 \rightarrow \text{успех}$$

Временная базовая точка $\bar{x}^{-2} = \bar{x}_p^{-2}$

$$\bar{x}_p^{-3} = \bar{x}^{-2} + (\bar{x}^{-2} - \bar{x}^{-1}) = \begin{bmatrix} -2 + (-2 + 3) \\ -2 + (-2 + 3) \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$f(\bar{x}_p^{-3}) = 17 \rightarrow \text{успех}$$

Временная базовая точка $\bar{x}^{-3} = \bar{x}_p^{-3}$

$$\bar{x}_p^{-4} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad f(\bar{x}_p^{-4}) = 0 \rightarrow \text{успех}$$

Временная базовая точка $\bar{x}^{-4} = \bar{x}_p^{-4}$

$$\bar{x}_p^{-5} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad f(\bar{x}_p^{-5}) = 17 \rightarrow \text{неуспех}$$

3. Исследовательский поиск

возврат к предыдущей базовой точке

$$\bar{x}^4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \Delta x = 1, \quad \bar{x}^{-5} = \bar{x}^{-4} \pm \Delta x.$$

$$f\begin{bmatrix} 1 \\ 1 \end{bmatrix} = f\begin{bmatrix} -1 \\ -1 \end{bmatrix} = 17, \quad f\begin{bmatrix} -1 \\ 1 \end{bmatrix} = f\begin{bmatrix} 1 \\ -1 \end{bmatrix} = 9$$

неуспех $\rightarrow \Delta x = 0.5, \Delta x > \varepsilon.$

4. Исследовательский поиск

$$\Delta x = 0.5, \quad \bar{x}^{-5} = \bar{x}^{-4} \pm \Delta x$$

$$f\begin{bmatrix} 0.5 \\ 0 \end{bmatrix} = f\begin{bmatrix} -0.5 \\ 0 \end{bmatrix} = 2, \quad f\begin{bmatrix} 0 \\ 0.5 \end{bmatrix} = f\begin{bmatrix} 0 \\ -0.5 \end{bmatrix} = 1.25$$

неуспех $\rightarrow \Delta x = 0.25, \rightarrow \Delta x = 0.25, \Delta x > \varepsilon.$

5. Исследовательский поиск

$$\Delta x = 0.25, \quad \bar{x}^{-5} = \bar{x}^{-4} \pm \Delta x$$

$$f\begin{bmatrix} 0.25 \\ 0 \end{bmatrix} = f\begin{bmatrix} -0.25 \\ 0 \end{bmatrix} = 0.5, \quad f\begin{bmatrix} 0 \\ 0.25 \end{bmatrix} = f\begin{bmatrix} 0 \\ -0.25 \end{bmatrix} = 0.313$$

неуспех $\rightarrow \Delta x = 0.125, \rightarrow \Delta x = 0.125, \Delta x > \varepsilon.$

6. Исследовательский поиск

$$\Delta x = 0.125, \bar{x}^5 = \bar{x}^4 \pm \Delta x$$

$$f \begin{bmatrix} 0.125 \\ 0 \end{bmatrix} = f \begin{bmatrix} -0.25 \\ 0 \end{bmatrix} = 0.125,$$

$$f \begin{bmatrix} 0 \\ 0.125 \end{bmatrix} = f \begin{bmatrix} 0 \\ -0.125 \end{bmatrix} = 0.078$$

неуспех $\rightarrow \Delta x = 0.0625, \Delta x < \varepsilon.$

Выход из итерационной процедуры. Решением задачи оптимизации является последняя базовая точка \bar{x}^4

$$\bar{x}^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad f \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0.$$

Стратегия поиска приведена на рисунке 3.7.

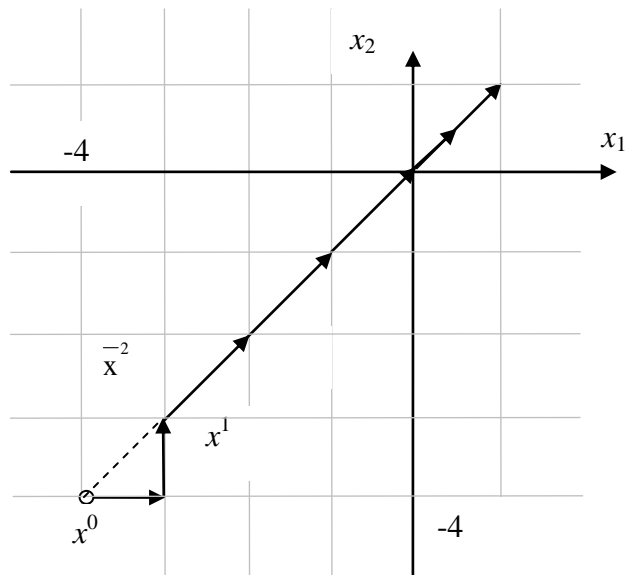


Рисунок 3.7.

3.3.2. Метод оврагов

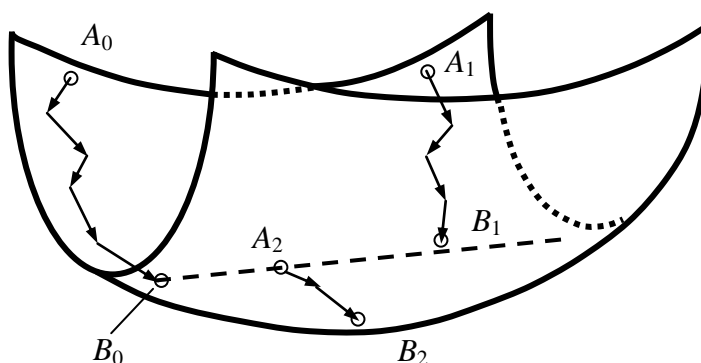
Существует большое количество модификаций метода Хука – Дживса. Однако во всех процессах локального поиска существует опасность "зацикливания", т. е. блуждания в окрестности локального экстремума, в котором значения целевой функции меньше, чем в близлежащих точках, но весьма далеки от абсолютного минимума. Аналогичная идея лежит в основе метода оврагов, предложенный для отыскания экстремумов функций многих переменных И. М. Гельфандом и М. Л. Цетлиным.

Метод оврагов имеет хорошую геометрическую иллюстрацию, представленную на рисунках 3.8-а и 3.8-б на поверхности и линиях уровня (дне оврага) соответственно.

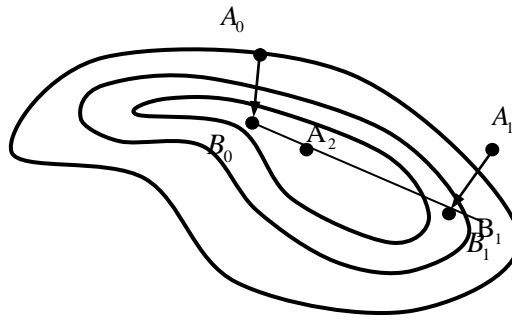
Эффективность метода особенно проявляется, если переменные, определенные целевой функцией, можно разделить на существенные и несущественные. В этом состоит наглядная ассоциация с оврагом, сильно зависящем от координаты y и слабо зависящем – от координаты x .

Геометрическая интерпретация метода следующая.

Из произвольной точки A_0 осуществляется ряд итераций локального спуска, пока уменьшение целевой функции $f(x, y) > \varepsilon$. Пусть локальный спуск привел нас в некоторую точку B_0 . Зафиксируем ее и выберем некоторую точку A_1 , из которой также осуществим локальный спуск и попадем в точку B_1 .



а



б

Рисунок 3.7. Геометрическая иллюстрация метода оврагов

Далее соединим точки B_0 и B_1 и сделаем большой шаг вдоль луча B_0B_1 в сторону уменьшения целевой функции. Попадаем в точку A_2 . Из точки A_2 вновь осуществим локальный спуск. Таким образом, процедура повторяется.

Таким чередованием существующий и несуществующих переменных хорошо отслеживается дно оврага и путь к глобальному экстремуму.

Замечание. Недостаток метода - сложно разделить на существенные и несущественные переменные.

Некоторое развитие метода оврагов - метод релаксаций.

3.4. Градиентные методы

Будем считать, что компоненты вектора градиента целевой функции могут быть записаны в аналитическом виде.

Основа стратегии поиска безусловного экстремуму по-прежнему состоит в построении последовательности векторов $\bar{x}^0, \bar{x}^1, \dots, \bar{x}^k \in R^n$ чтобы для целевой функции выполнялось условие $f(\bar{x}^0) > f(\bar{x}^1) > \dots > f(\bar{x}^k) > \dots$

Пусть в результате поиска экстремума найдена точка $\bar{x}^k \in R^n$ и требуется определить направление наискорейшего локального спуска, т.е. наибольшего локального уменьшения целевой функции. Разложим целевую функцию в ряд Тейлора, ограничиваясь линейными членами:

$$\begin{aligned} f(\bar{x}) &= f(\bar{x}^{\bar{k}}) + \nabla f^T(\bar{x}^{\bar{k}}) \cdot \Delta \bar{x}^{\bar{k}} + \dots \\ \forall x &\in [\bar{x}^{\bar{k}}, \bar{x}^{\bar{k}} + \Delta \bar{x}^{\bar{k}}] \end{aligned} \quad (3.7)$$

Локальное уменьшение целевой функции определяется вторым слагаемым, т.к. $f(\bar{x}^{\bar{k}}) = const$. Наибольшее уменьшение целевой функции ассоциируется с выбором такого направления, которому соответствует наиболее отрицательная величина скалярного произведения $(\nabla f, \Delta \bar{x})$. Отсюда следует формула простейшего метода градиентного спуска.

$$\bar{x}^{\bar{k}+1} = \bar{x}^{\bar{k}} - \alpha^k \cdot \nabla f(\bar{x}^{\bar{k}}), \quad (3.8)$$

где α^k - некоторый положительный параметр, характеризующий длину шага на каждой итерации и обеспечивающий выполнение цели оптимизации

$$f(\bar{x}^{\bar{k}+1}) < f(\bar{x}^{\bar{k}}).$$

Реализация простейшего алгоритма на основе формулы поиска (3.8) носит название метода **градиентного поиска** и включает два этапа.

1. Пробный шаг для определения направления градиента.
2. Одновременное смещение в сторону градиента всех координат вектора \bar{x} .

Если на некоторой итерации поиска условие $f(\bar{x}^{\bar{k}+1}) < f(\bar{x}^{\bar{k}})$ не выполняется, то шаг α^k в формуле (3.8) дробят для выполнения условия и вычисления продолжают.

В качестве окончания итерационного процесса обычно используют близость к нулю градиента, которая оценивается для всех координат вектора-градиента

$$\left| \frac{\partial f(\bar{x}^{\bar{k}})}{\partial x_i} \right| \leq \varepsilon, \forall i = \overline{1, n} \quad (3.9-a)$$

$$\sqrt{\sum_{i=1}^n \left[\frac{\partial f(\bar{x}^k)}{\partial x_i} \right]^2} \leq \mathcal{E}. \quad (3.9-б)$$

Формулы (3.9) являются правилами останова итерационной процедуры поиска в градиентных методах и при выполнении условий останова (3.9) считается, что оптимум найден $\bar{x}^* = \bar{x}^k$, $f^* = f(\bar{x}^k)$.

Замечание. Недостатком градиентного метода является долгое движение к экстремуму при наличии в линиях уровня оврагов и гребней.

Алгоритмизация метода градиентного спуска

1. Задание исходных данных

\bar{x}^0 - начальные условия,
 \mathcal{E} - точность вычисления экстремума,
 α - длина шага движения,
число итераций $k = 0$.

2. Пробный шаг - для определение величины и направления градиента

$$\nabla f(\bar{x}^k) \Big|_{\bar{x} = \bar{x}^k}.$$

3. Проверка условий останова

если $\left| \nabla f(\bar{x}^k) \right| < \mathcal{E} \rightarrow$ печать результатов $\bar{x}^k, f(\bar{x}^k)$,
выход из итерационной процедуры.
если $\left| \nabla f(\bar{x}^k) \right| > \mathcal{E} \rightarrow$ Шаг 4.

4. Одновременное смещение в сторону градиента всех координат

$$\bar{x}^{k+1} = \bar{x}^k - \alpha_k \cdot \nabla f(\bar{x}^k), \quad k = k + 1.$$

5. Проверка цели оптимизации

если $f(\bar{x}^k) < f(\bar{x}^{k-1}) \rightarrow$ Шаг 2 (вычисление нового градиента).

если $f(\bar{x}^k) > f(\bar{x}^{k-1}) \rightarrow \alpha = \frac{\alpha}{2} \rightarrow$ Шаг 4.

Пример.

Найти минимум целевой функции

$$f(x, y) = x_1^2 + 2 \cdot x_2^2 + e^{x_1 + x_2}$$

методом градиентного спуска при начальных приближениях

$$x^0 = (0, 0); \quad \alpha_0 = 1; \quad \varepsilon = 5 \cdot 10^2.$$

Целевая функция представлена на рисунке 3.9.

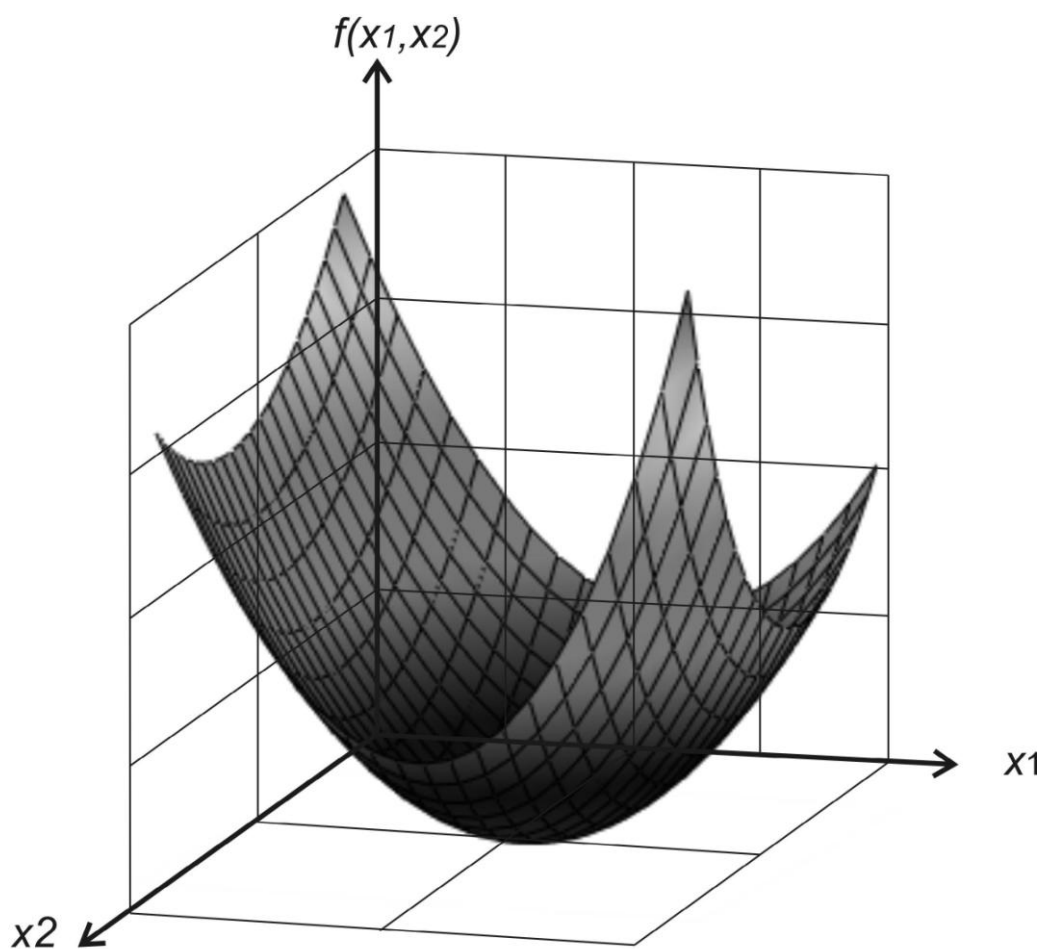


Рисунок 3.9.

1. Вычислим $f \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 1$.

1.1. Пробный шаг – для определения величины и направления градиента

$$\nabla f(\bar{x}^0) = \left[\frac{2 \cdot x_1 + e^{x_1+x_2}}{4 \cdot x_2 + e^{x_1+x_2}} \right] \bigg|_{x^0=(0,0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} > \varepsilon.$$

1.2. Смещение в сторону градиента всех координат вектора x .

$$\begin{aligned} \bar{x}^1 &= \bar{x}^0 - \alpha_0 \cdot \nabla f(\bar{x}^0); \\ \bar{x}^1 &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 1 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}. \end{aligned}$$

1.3. Проверим условие цели поиска $f^1 < f^0$

$$f \begin{bmatrix} -1 \\ -1 \end{bmatrix} = 3.145 > f \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 1.$$

Условие не выполняется, следовательно необходимо дробить шаг

$$\alpha_0 = \frac{\alpha_0}{2}, \quad \alpha_0 = 0.5.$$

1.4. Градиент не был изменен, т.е. $\nabla f(x^0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Смещение в сторону градиента координат вектора x

$$\bar{x}^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 0.5 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -0.5 \\ -0.5 \end{pmatrix}.$$

Проверим условие цели поиска

$$f(\bar{x}^1) = 1.118 > f(\bar{x}^0) = 1$$

Условие не выполняется, необходимо дробить шаг

$$\alpha_0 = \frac{\alpha_0}{2}, \quad \alpha_0 = 0.25.$$

1.5. Смещение в сторону градиента координат x

$$\bar{x}^{-1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 0.25 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -0.25 \\ -0.25 \end{pmatrix}$$

Проверим условие цели поиска

$$f(\bar{x}^{-1}) = 0.794 < f(\bar{x}^{-0}) = 1$$

условие поиска выполняется.

Необходимо менять направление градиента.

2. Пробный шаг – для определения величины и направления градиента

$$\nabla f(\bar{x}^{-1}) = \left[\frac{2 \cdot x_1 + e^{x_1+x_2}}{4 \cdot x_2 + e^{x_1+x_2}} \right] \bigg|_{\bar{x}^{-1} = (-0.25, -0.25)} = \begin{bmatrix} 0.106 \\ -0.393 \end{bmatrix} > \varepsilon$$

2.1. Смещение в сторону градиента координат вектора x .

$$\bar{x}^{-2} = \begin{pmatrix} -0.25 \\ -0.25 \end{pmatrix} - 0.25 \cdot \begin{pmatrix} 0.106 \\ -0.393 \end{pmatrix} = \begin{pmatrix} -0.276 \\ -0.151 \end{pmatrix}.$$

Проверим условие цели поиска

$$f(\bar{x}^{-2}) = 0.774 < f(\bar{x}^{-1}) = 0.794$$

- условие поиска выполняется.

Необходимо менять направление градиента.

3. Пробный шаг – для определения величины и направления градиента

$$\nabla f(\bar{x}^{-2}) = \left[\frac{2 \cdot x_1 + e^{x_1+x_2}}{4 \cdot x_2 + e^{x_1+x_2}} \right] \bigg|_{\bar{x}^{-2} = (-0.276, -0.151)} = \begin{bmatrix} 0.0982 \\ 0.0451 \end{bmatrix} > \varepsilon$$

3.1. Смещение в сторону градиента координат вектора x .

$$\bar{x}^{-3} = \begin{pmatrix} -0.301 \\ -0.162 \end{pmatrix}.$$

Проверим условие цели поиска

$$f(\bar{x}^{-3}) = 0.772 < f(\bar{x}^{-2}) = 0.774$$

условие поиска выполняется.

4. Пробный шаг – для определения величины и направления градиента

$$\nabla f(\bar{x}^{-3}) = \left[\frac{2 \cdot x_1 + e^{x_1+x_2}}{4 \cdot x_2 + e^{x_1+x_2}} \right]_{\bar{x}^{-3} = (-0.301, -0.162)} = \begin{bmatrix} 0.0262 \\ -0.023 \end{bmatrix} < \varepsilon$$

Заданная точность достигнута, конец поиска.

При выполнении расчетов результаты целесообразно заносить в таблицу. В таблице 3.1 представлены пошаговые результаты поиска экстремума.

Таблица 3.1

Метод градиентного поиска

k	x^k	$f(x^k)$	$\nabla f(x^k)$	α_k	Примечание
0	0, 0	1	$1 > \varepsilon$ $1 > \varepsilon$	1	
1	-1, -1	3.145	-		$f \uparrow$ $\alpha/2$
	0, 0	1	$1 > \varepsilon$ $1 > \varepsilon$	0.5	
	-0.5, -0.5	1.118	-		$f \uparrow$ $\alpha/2$
	0, 0	1	$1 > \varepsilon$ $1 > \varepsilon$	0.25	

2	-0.25, -0.25	0.794	$0.106 > \varepsilon$ $-0.393 > \varepsilon$	0.25	$f \downarrow$
3	-0.276, -0.1516	0.774	$0.0982 > \varepsilon$ $0.0451 < \varepsilon$	0.25	$f \downarrow$
4	-0.301, -0.163	0.772	$0.0262 < \varepsilon$ $-0.023 < \varepsilon$		$\varepsilon < 0.05$

Графическая иллюстрация стратегии поиска приведена на рисунке 3.10.

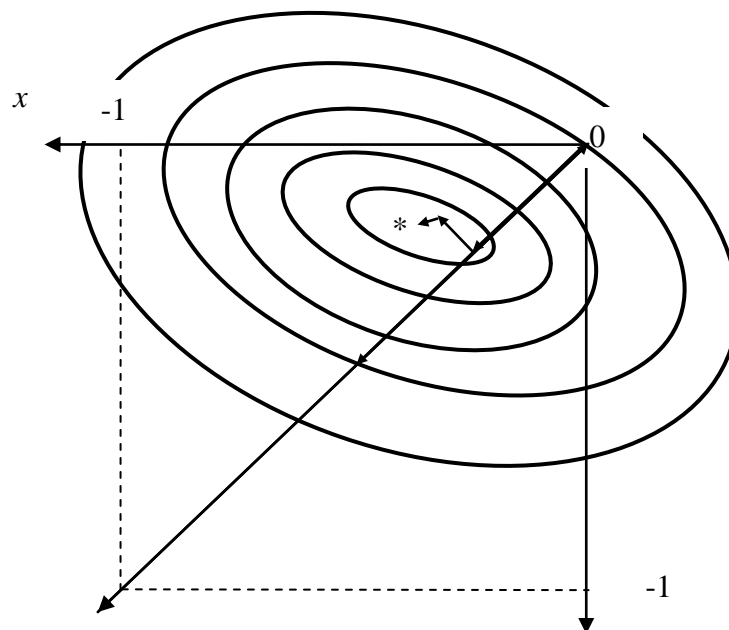


Рисунок 3.10. Геометрическая иллюстрация

3.4.1. Метод наискорейшего спуска (метод Коши)

Метод наискорейшего спуска это модификация градиентного метода. Отличается от него способом нахождения длины шага α^k в формуле (3.8).

Предлагается находить длину шага α^k на каждой итерации из условия:

$$\varphi_k(\alpha) = \min_{\alpha > 0} f[\bar{x}^k - \alpha \cdot \nabla f(\bar{x}^k)]. \quad (3.10)$$

Из (3.10) следует, что величина α^k находится, как решение задачи минимизации функции $f(\bar{x}^{k+1})$ вдоль направления $\nabla f(\bar{x}^k)$. Т.е. длина шага α^k может быть найдена как стационарная точка из решения уравнения.

$$\frac{\partial f[\bar{x}^k - \alpha \cdot \nabla f(\bar{x}^k)]}{\partial \alpha} = 0$$

Основные этапы метода Коши.

1. Нахождение градиента

$$\nabla f(\bar{x}^k) \Big|_{\bar{x}=\bar{x}_k}.$$

2. Если $|\nabla f(\bar{x}^k)| < \varepsilon \rightarrow$ печать результатов $\bar{x}^k, f(\bar{x}^k)$, конец.

если $|\nabla f(\bar{x}^k)| > \varepsilon \rightarrow s$ Шаг 3.

3. Из решения уравнения

$$\frac{\partial f[\bar{x}^k - \alpha \cdot \nabla f(\bar{x}^k)]}{\partial \alpha} = 0$$

находим α^k (аналитически, либо методами одномерного поиска).

4. Движение вдоль градиента

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

переход на шаг 1.

Пример.

Найти методом наискорейшего спуска минимум функции

$$f(\bar{x}) = x_1^2 + 2 \cdot x_2^2 + e^{x_1+x_2}$$

при начальных приближениях

$$\bar{x}^0 = (0,0); \quad \alpha_0 = 1; \quad \varepsilon = 5 \cdot 10^{-2}.$$

Вычислим

$$f \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 1.$$

Запишем критерий минимизации (3.10) относительно неизвестного

α

$$\begin{aligned}
\varphi_0(\alpha) &= f[\bar{x}^0 - \alpha \cdot \nabla f(\bar{x}^0)] = \\
&= f[0 - \alpha \cdot 1, 0 - \alpha \cdot 1] = \\
&= f[-\alpha, -\alpha] = \\
&= \alpha^2 + 2 \cdot \alpha^2 + e^{-2\alpha} = 3 \cdot \alpha^2 + e^{-2\alpha}.
\end{aligned}$$

Продифференцируем полученное уравнение относительно переменной α

$$\varphi'_0(\alpha) = 6 \cdot \alpha - 2 \cdot e^{-2\alpha} = 0.$$

Решение этого уравнения дает оптимальное значение длины шага $\alpha_0 = 0.2188$. На рисунке 3.11 приведено графическое решение.

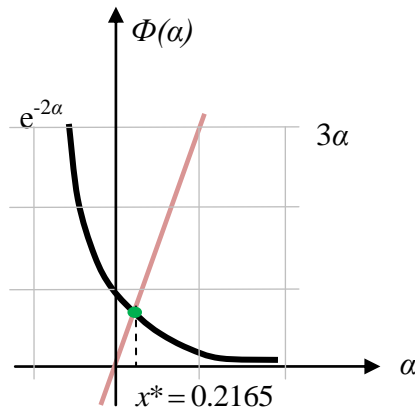


Рисунок 3.11. Графическое нахождение шага α

Вычислим

$$\begin{aligned}
\bar{x}^1 &= [\bar{x}^0 - \alpha_0 \cdot \nabla f(\bar{x}^0)] = [-0.2163, -0.2163] \\
f[\bar{x}^1] &= 0.7892
\end{aligned}$$

Отметим, что в методе простого градиентного спуска путем дробления шага было найдено значение

$$\bar{x}^1 = [-0.25, -0.25], \quad f[\bar{x}^1] = 0.794$$

Замечание.

Сравнивая рассмотренные градиентные методы, нужно отметить, что метод простого градиентного спуска с неопределенным шагом. Метод Коши – с шагом оптимальной длины для выбранного направления.

Полностью результаты решения примера методом Коши приведены в таблице 3.2.

Таблица 3.2

Промежуточные вычисления методом Коши

k	\bar{x}^k	$\nabla f(x^k)$	α	\bar{x}^{k+1}	$f(x^k)$
0	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	0.2163	$\begin{bmatrix} -0.2163 \\ -0.2163 \end{bmatrix}$	1
1	$\begin{bmatrix} -0.2163 \\ -0.2163 \end{bmatrix}$	$\begin{bmatrix} 0.2163 \\ -0.2163 \end{bmatrix}$	0.0333	$\begin{bmatrix} -0.288 \\ -0.1442 \end{bmatrix}$	0.7892
2	$\begin{bmatrix} -0.2884 \\ -0.1442 \end{bmatrix}$	$\begin{bmatrix} 0.0721 \\ 0.0721 \end{bmatrix} < \varepsilon$	-	-	0.7736

Градиент является чрезвычайно важным направлением.

Пусть $\bar{d} = (d_1, \dots, d_n)$ - произвольное заданное направление. Тогда минимальная скорость изменения функции $f(x)$ вдоль направления \bar{d} выражается скалярным произведением

$$(\nabla f, d) = |\nabla f(x)| \cdot |d| \cdot \cos \beta$$

где β – угол между направлениями векторов ∇f и d , рисунок 3.12.

Скалярное произведение принимаем максимальное значение, когда $\cos(\beta) = 1$, т.е. угол $\beta = 0$ градусов.

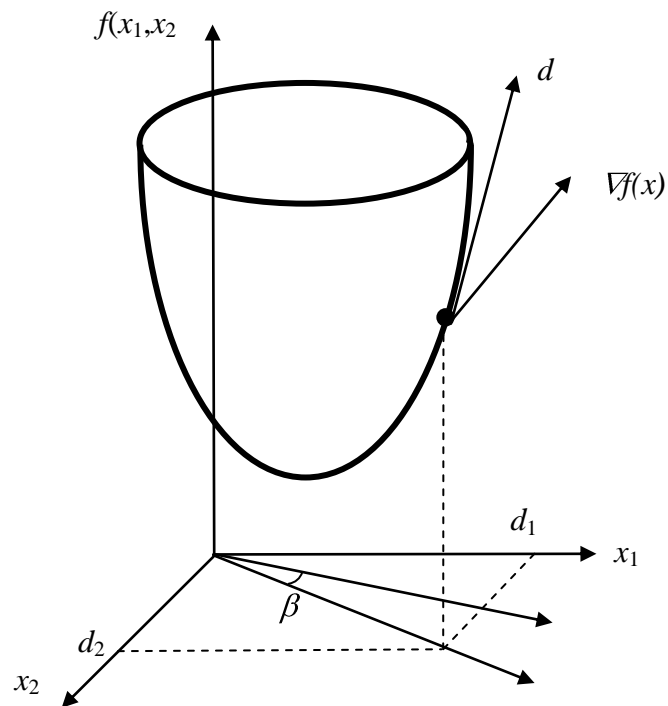


Рисунок 3.12.

Таким образом, скорость изменения $f(x)$ максимальна, когда ∇f совпадает с заданным направлением d , а градиент ∇f – направление наискорейшего возрастания функции $f(x)$.

Направление градиента перпендикулярно в каждой точке линии постоянного уровня, рисунок 3.13.

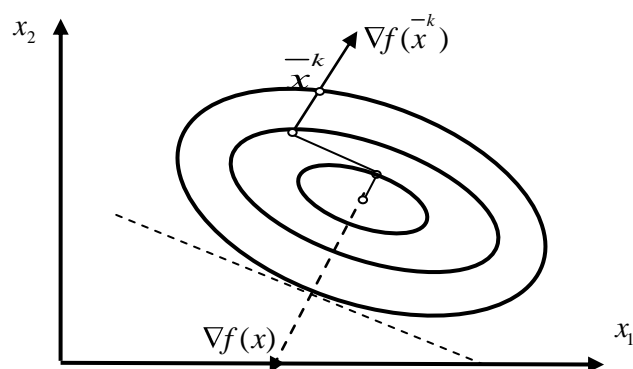


Рисунок 3.13.

В методе наискорейшего спуска используется это свойство градиента. Из некоторой точки \bar{x}^k движение в направлении минимума осуществляется в сторону антиградиента $(-\nabla f(x_k))$.

Если ввести вектор $\bar{P}^k = \bar{x}^{k+1} - \bar{x}^k$ тогда вектор $\nabla f(x^{k+1})$ перпендикулярен вектору P^k , поскольку x^{k+1} была выбрана путем минимизации функции $f(\bar{x}^k - \alpha \cdot \nabla f(\bar{x}^k))$, рисунок 3.14.

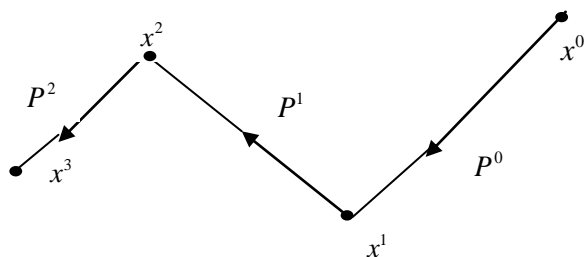


Рисунок 3.14.

$$\bar{P}^k = \bar{x}^{k+1} - \bar{x}^k = \alpha_k \cdot \nabla f(\bar{x}^k), \quad (3.11)$$

т.е. P^k - это градиент оптимальной длины.

Замечания.

1. Метод Коши хорошо работает на больших расстояниях от оптимума. Поэтому часто используется в качестве начальной процедуры. Однако в окрестности оптимума сходимость низкая. Здесь рекомендуется подключать метод Ньютона (II порядка).

2. Для квадратичной функции $f(\bar{x}) = \frac{1}{2} \cdot \bar{x}^T \cdot A \cdot \bar{x}$ формула Коши примет вид

$$\bar{x}^{k+1} = \bar{x}^k - \alpha_k \cdot A \cdot \bar{x}^k.$$

Величина α_k может быть найдена в явном виде

$$\alpha_k = \frac{(\nabla f(\bar{x}^k), \nabla f(\bar{x}^k))}{(A \cdot \nabla f(\bar{x}^k), \nabla f(\bar{x}^k))}. \quad (3.12)$$

3. Для сепарабельных функций за оптимальное решение достигается за n шагов.

3.4.2. Метод сопряженных градиентов

Понятие сопряженности векторов

В разделе 3 была введена матрица Гессе $H = \frac{\partial^2 f(\bar{x}^k)}{\partial x_i \partial x_j}$, элементы которой для n - мерного пространства R_n имеют вид.

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

Пусть имеем два направления, которые характеризуются векторами \bar{P}^0 и \bar{P}^1 .

Если скалярное произведение $(\bar{P}^1, \bar{P}^0) = 0$, то вектора ортогональны, т.е. взаимно перпендикулярны.

Если скалярное произведение $(\bar{P}^1, H \cdot \bar{P}^0) = 0$, то вектора называются сопряженными относительно матрицы H .

Здесь H – положительно определенная квадратная матрица.

Приведенные определения имеют определенную геометрическую иллюстрацию, приведенную на рисунке 3.15. Матрица H , умноженная на вектор \bar{P}^0 , изменяет его длину и поворачивает на некоторый угол. И этот новый полученный вектор ортогонален к вектору \bar{P}^1 .

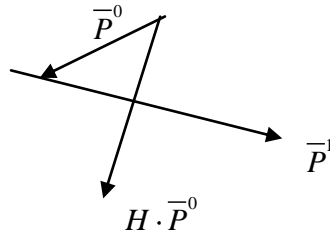


Рисунок 3.15.

Свойство сопряженности векторов используется в методе сопряженных градиентов, который отличается от метода наискорейшего спуска только выбором направления уменьшения функции на каждом шаге. Вместо вектора $-\nabla f(\bar{x}^k)$ метод использует вектор $-\bar{P}^k$.

Формула метода сопряженных градиентов имеет вид.

$$\bar{x}^{k+1} = \bar{x}^k - \alpha_k \cdot \bar{P}^k, \quad k = 0, 1, \dots \quad (3.13)$$

$\bar{x}^0 \in R^n$ - вектор начальных приближений. Шаг α_k выбирается аналогично (3.10) из решения задачи одномерной оптимизации:

$$\varphi(\alpha_k) = \min_{\alpha > 0} f[\bar{x}^k - \alpha \cdot \bar{P}^k]. \quad (3.14)$$

Направление спуска \bar{P}_k определяется по формуле

$$\bar{P}_k = \nabla f(\bar{x}^k) + \beta_k \cdot \bar{P}^{k-1}, \quad (3.15)$$

где

$$\bar{P}^0 = \nabla f(x^0),$$

$$\beta_k = \frac{|\nabla f(\bar{x}^k)|^2}{|\nabla f(\bar{x}^{k-1})|^2} = \frac{\sum_{i=1}^n \left(\frac{\partial f(\bar{x}^k)}{\partial x_i} \right)^2}{\sum_{i=1}^n \left(\frac{\partial f(\bar{x}^{k-1})}{\partial x_i} \right)^2}.$$

Вектора P_1, P_2, P_3 , - являются сопряженными.

Замечания.

1. В данном методе направление спуска определяет не только анти-градиент $-\nabla f(\bar{x}^k)$, но и направление спуска на предыдущем шаге \bar{P}^{k-1} .

2. Для снижения влияния накапливающихся ошибок вычисления рекомендуется через n шагов поиска обнулять длину шага $\beta_n = 0$.

3. Если целевая функция - выпуклая квадратичная функции, принадлежащая пространству R^n , то для нахождения экстремума методом сопряженных градиентов требуется не более n итераций. Конечное число шагов – большое достижение для практики. Это обеспечивается поворотом главных осей линий уровня таким образом, чтобы одна из главных осей прошла через точку экстремума.

4. Для функции общего вида (а не квадратичной) метод сопряженных градиентов еще не разработан. Основная трудность в том, что матрица Гессе получается функциональной, т.е. зависящей от переменных.

Пример. Найти методом сопряженных градиентов минимум функции

$$f(\bar{x}) = x_1^2 + 2 \cdot x_2^2 + x_1 \cdot x_2 - 7 \cdot x_1 - 7 \cdot x_2$$

при начальных приближениях $\bar{x}^0 = (0,0)$; $\varepsilon = 5 \cdot 10^{-2}$.

Так как функция $f(\bar{x})$ - квадратичная, следовательно, оптимальное решение \bar{x}^*_{opt} должно быть найдено за 2 шага.

$$\nabla f(\bar{x}) = (2 \cdot x_1 + x_2 - 7, x_1 + 4 \cdot x_2 - 7) .$$

Шаг 1. по формулам (3.15) и (3.14) находим направление спуска и формируем функцию для нахождения длины шага

$$\bar{p}^0 = \nabla f(\bar{x}^0) = 2 \cdot x_1 + x_2 - 7, x_1 + 4 \cdot x_2 - 7 \Big|_{\bar{x}^0} = (-7, -7) .$$

$$\begin{aligned}\varphi_0(\alpha_0) &= f[\bar{x}^0 - \alpha^0 \cdot \nabla f(\bar{x}^0)] = f(7 \cdot \alpha, 7 \cdot \alpha) = \\ &= 49 \cdot \alpha^2 + 2 \cdot 49 \cdot \alpha^2 + 49 \cdot \alpha^2 - 49 \cdot \alpha - 49 \cdot \alpha = \\ &= \alpha^2(49 + 98 + 49) - \alpha(49 + 49) = 98(2 \cdot \alpha^2 - \alpha).\end{aligned}$$

Из уравнения

$$\varphi'_0(\alpha) = 0$$

находим значение α

$$4 \cdot \alpha - 1 = 0,$$

$$\alpha = 0.25.$$

По формуле (3.13) найдем новое значение вектора решений

$$\bar{x}^1 = \bar{x}^0 - \alpha \cdot \bar{P}^0 = (0, 0) - 0.25(-7, -7) = \left(\frac{7}{4}, \frac{7}{4}\right).$$

Шаг 2.

$$\nabla f(\bar{x}^1) = \left(-\frac{7}{4}, \frac{7}{4}\right).$$

$$\beta_1 = \frac{\sum_{i=1}^2 \left(\frac{\partial f(\bar{x}^1)}{\partial x_i} \right)^2}{\sum_{i=1}^2 \left(\frac{\partial f(\bar{x}^0)}{\partial x_i} \right)^2} \bigg|_{\bar{x}^k} = \frac{\nabla f(\bar{x}^1)}{\nabla f(\bar{x}^0)} = \frac{1}{16}.$$

$$\bar{P}^1 = \nabla f(\bar{x}^1) + \beta_1 \cdot \bar{P}^0 = \left(-\frac{7}{4}, \frac{7}{4}\right) + \frac{1}{16}(-7, -7) = \left(-\frac{35}{16}, \frac{21}{16}\right)$$

$$\varphi_1(\alpha) = f[\bar{x}^1 - \alpha_1 \cdot \bar{x}^1] = \frac{49}{32} \cdot \left(\frac{7}{2} \cdot \alpha^2 - \alpha - 392\right).$$

Следовательно

$$\alpha = \frac{4}{7}.$$

$$\bar{x}^{-2} = \bar{x}^{-1} - \alpha \cdot \bar{P}^1 = \left(\frac{7}{4}, \frac{7}{4}\right) - \frac{4}{7} \cdot \left(-\frac{35}{16}, \frac{21}{16}\right) = (3, 1);$$

$$x_{opt}^* = (3, 1).$$

Проверим условия сходимости итерационного процесса (3.9)

$$\nabla f(\bar{x}^{-2}) = (2 \cdot x_1 + x_2 - 7, x_1 + 4 \cdot x_2 - 7) \Big|_{\bar{x}^{-2}} = \\ (2 \cdot 3 + 1 - 7, 3 + 4 - 7) = (0, 0) < \varepsilon.$$

На рисунке 3.16 приведены стратегии поиска для метода Коши и сопряженных градиентов, а в таблицах 3.3 и 3.4 – численные результаты поиска.

Таблица 3.3

Метод наискорейшего спуска (Коши)

Номер итерации	$\nabla f(\bar{x}^k) = (x_1^k, x_2^k)$	α_k	$\bar{x}^k = (x_1^k, x_2^k)$
1	-7 -7	0.25	1.748 1.748
2	-1.755 1.742	0.503	2.631 0.873
3	-0.866 -0.879	0.25	2.847 1.092
4	-0.214 0.215	0.497	2.953 0.985
5	-0.108 -0.107	0.25	2.980 1.012
6	-0.028 -0.027	0.503	2.994 0.998
7	-0.013 -0.013	0.25	2.998 1.001

Таблица 3.4

Метод сопряженных градиентов

Номер итерации	$\nabla f(\bar{x}^k) = (x_1^k, x_2^k)$	\bar{P}^k	β_k	α_k	$\bar{x}^k = (x_1^k, x_2^k)$
	-7 -7	-7 -7			7/4 7/4

2	$-\frac{3}{4}$ $\frac{7}{4}$	$-\frac{35}{16}$ $\frac{21}{16}$	$\frac{1}{16}$	$\frac{4}{7}$	$\frac{3}{1}$

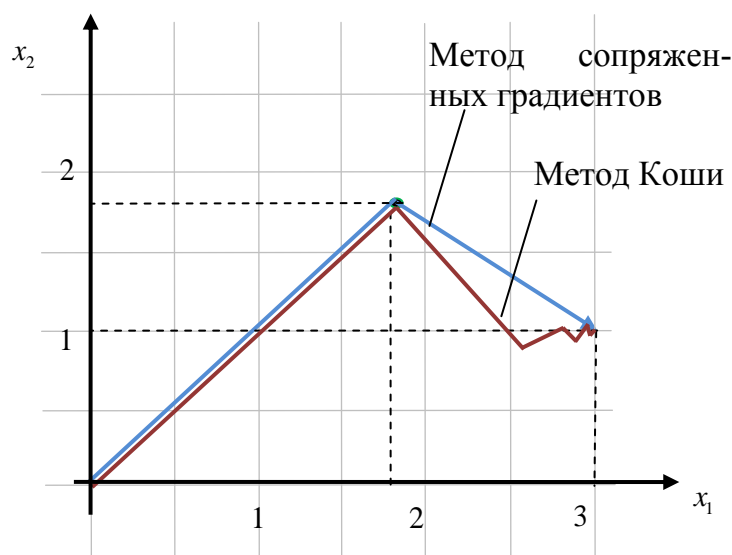


Рисунок 3.16.

3.5. Градиентные методы второго порядка

Рассмотренные выше методы используют информацию лишь о первых производных целевой функции $f(\bar{x})$, $\bar{x} \in R_n$.

В окрестностях точки локального минимума составляющие градиенты (частные производные) $\frac{\partial f(\bar{x})}{\partial x_i}$, $i = 1, n$ малы, и поэтому приходится делать большее число шагов для попадания в \mathcal{E} -окрестность экстремума.

Ускорить этот процесс помогает использование вторых производных функции $f(\bar{x})$. Нахождение последующее приближение к экстремуму по-прежнему осуществляется по формуле:

$$\bar{x}^{-k+1} = \bar{x}^{-k} + \Delta \bar{x}^{-k}.$$

Требуется найти приращение $\Delta \bar{x}^{-k}$.

Пусть $f(\bar{x})$ - выпуклая, дважды дифференцируемая в пространстве R^n функция.

Градиентные методы первого порядка использовали разложение целевой функции $f(\bar{x})$ в ряд Тейлора в окрестности точки \bar{x}^k с отбрасыванием нелинейных членов, формула (3.7) .

$$f(\bar{x}) = f(\bar{x}^k) + \nabla f(\bar{x}^k)^T \cdot \Delta \bar{x}.$$

Добавим в разложение в ряд Тейлора квадратичные члены, получим.

$$f(\bar{x}) = f(\bar{x}^k) + \nabla f(\bar{x}^k) \cdot \Delta \bar{x} + 0.5 \cdot \Delta \bar{x}^T \cdot \nabla^2 f(\bar{x}^k) \cdot \Delta \bar{x}$$

Пусть $\bar{x} = \bar{x}^{k+1}$, тогда

$$\nabla f(\bar{x}^{k+1}) = \nabla f(\bar{x}^k) + \nabla^2 f(\bar{x}^k) \cdot \Delta \bar{x},$$

где

$$\Delta \bar{x} = \bar{x}^{k+1} - \bar{x}^k,$$

$$\nabla f(\bar{x}^{k+1}) = \frac{f(\bar{x}^{k+1}) - f(\bar{x}^k)}{\Delta \bar{x}}.$$

Будем предполагать, что в точке \bar{x}^{k+1} находится экстремум целевой функции, поэтому градиент функции $\nabla f(\bar{x}^{k+1}) = 0$.

Перепишем условия, используя $H(\bar{x}^k)$ - матрицу Гессе.

$$\nabla f(\bar{x}^{k+1}) = \nabla f(\bar{x}^k) + H(\bar{x}^k) \cdot \Delta \bar{x} = 0.$$

Откуда искомое приращение

$$\Delta \bar{x} = -H^{-1}(\bar{x}^k) \cdot \nabla f(\bar{x}^k). \quad (3.16)$$

А последовательность приближений к точке экстремума можно построить по формуле:

$$\bar{x}^{k+1} = \bar{x}^k + H^{-1}(\bar{x}^k) \cdot \nabla f(\bar{x}^k), \quad (3.17)$$

где $H^{-1}(\bar{x}^k)$ - обратная матрица Гессе в точке \bar{x}^k .

Полученная формула носит название формулы Ньютона.

Замечания.

1. Если начальное приближение \bar{x}^{-0} , которое выбрано удачно, то сходимость метода на порядок выше сходимости градиентных методов. Поэтому часто метод Ньютона используется для уточнения экстремума.

2. Сходимость метода Ньютона квадратичная. А само направление сходимости неоднозначно от итерации к итерации.

Если $H(\bar{x}^k)$ положительно определена, то спуск целевой функции.

$H(\bar{x}^k)$ отрицательно определена, то подъем целевой функции.

$H(\bar{x}^k)$ не определена, то нет однозначности в стратегии поиска.

Пример. Дана целевая функция

$$f(\bar{x}) = 8 \cdot x_1^2 + 5 \cdot x_2^2 + 4 \cdot x_1 \cdot x_2.$$

Найти минимум функции при методом Ньютона при начальных условиях $\bar{x}^{-0} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$. Отметим, что точное решение (0,0).

Решение.

Формула метода Ньютона

$$\bar{x}^{-k+1} = \bar{x}^{-k} + H^{-1}(\bar{x}^{-k}) \cdot \nabla f(\bar{x}^{-k})$$

Шаг 1.

$k = 0$

$$\nabla f(\bar{x}) = \begin{bmatrix} 2 \cdot 8 \cdot x_1 + 4 \cdot x_2 \\ 2 \cdot 5 \cdot x_2 + 4 \cdot x_1 \end{bmatrix}_{10,10} = \begin{bmatrix} 200 \\ 140 \end{bmatrix}.$$

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

$$H(\bar{x}) = \begin{pmatrix} 16 & 4 \\ 4 & 10 \end{pmatrix}.$$

$$H^{-1}(\bar{x}) = \frac{1}{144} \cdot \begin{bmatrix} 10 & -4 \\ -4 & 16 \end{bmatrix}.$$

$$\bar{x}^{-1} = \bar{x}^{-0} - H^{-1}(\bar{x}^{-0}) \cdot \nabla f(\bar{x}^{-0}).$$

$$\bar{x}^{-1} = \begin{bmatrix} 10 \\ 10 \end{bmatrix} - \frac{1}{144} \begin{bmatrix} 10 & -4 \\ -4 & 16 \end{bmatrix} \cdot \begin{bmatrix} 200 \\ 140 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Решение получено за 1 шаг.

Пример. Дана целевая функция

$$f(\bar{x}) = x_1^2 + 2 \cdot x_2^2 + x_1 \cdot x_2 - 7 \cdot x_1 - 7 \cdot x_2.$$

Найти минимум функции при $\bar{x}^{-0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ методом Ньютона.

Отметим, что точное решение (3,1).

Решение.

Формула метода Ньютона

$$\bar{x}^{-k+1} = \bar{x}^{-k} + H^{-1}(\bar{x}^{-k}) \cdot \nabla f(\bar{x}^{-k})$$

Шаг 1.

$$k = 0$$

$$\nabla f(\bar{x}) = \begin{bmatrix} 2 \cdot x_1 + x_2 - 7 \\ 4 \cdot x_2 + x_1 - 7 \end{bmatrix}_{0,0} = \begin{bmatrix} -7 \\ -7 \end{bmatrix}.$$

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

$$H(\bar{x}) = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}.$$

$$H^{-1}(\bar{x}) = \frac{1}{7} \cdot \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix}.$$

$$\bar{x}^{-1} = \bar{x}^{-0} - H^{-1}(\bar{x}^{-0}) \cdot \nabla f(\bar{x}^{-0}).$$

$$\bar{x}^{-1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \frac{1}{7} \begin{bmatrix} 4 & -1 \\ -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} -7 \\ -7 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Решение получено за 1 шаг.

3.5.1. Модифицированный метод Ньютона

Обеспечивает более устойчивую сходимость, чем метод Ньютона. Метод обеспечивает уменьшение целевой функции от итерации к итерации за счет поиска α_k вдоль направления $H(\bar{x}_k)^{-1} \nabla f(\bar{x}^k)$, аналогично методу Коши.

Формула модифицированного метода Ньютона получена на основе обычной формулы метода Ньютона дополнением адаптивного коэффициента α_k . Формула имеет вид

$$\bar{x}^{-k+1} = \bar{x}^{-k} - \alpha_k \cdot H(\bar{x}_k)^{-1} \nabla f(\bar{x}^k), \quad (3.18)$$

где α_k находится из решения задачи одномерной оптимизации

$$\varphi_k(\alpha_k) = \min_{\alpha > 0} \varphi_k(\alpha).$$

$$\varphi_k(\alpha) = f[\bar{x} - \alpha \cdot H(\bar{x})^{-1} \cdot \nabla f(\bar{x})].$$

Замечания.

1. Метод устраняет неоднозначность сходимости метода, обусловленной определенностью матрицы $H(\bar{x})$.

2. Параметр α_k , выбираемый на каждом шаге и обеспечивает выполнение условия поиска по целевой функции $f(\bar{x}) > f(\bar{x}^{k+1})$, которое в методе Ньютона может нарушаться.

3. Метод эффективен, если вычисление первой и второй производных не вызывает сложностей. Вторая сложность метода – построение и решение уравнения $\frac{\partial \varphi}{\partial \alpha} = 0$, содержащего матрицу Гессе.

4. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Задачей линейного программирования (ЛП) называется оптимизационная задача, в которой целевая функция – линейна на множестве линейных ограничений.

$$\begin{aligned} f(\bar{x}) &= \sum_{j=1}^n c_j \cdot x_j \rightarrow \min \\ \sum_{j=1}^n a_{ij} \cdot x_j &\leq b_i, i = \overline{1, m}, x_j \geq 0 \end{aligned} \quad (4.1)$$

Ограничения, накладываемые на координаты x_j , могут быть равенствами и неравенствами (I и II рода).

Задачи ЛП – самая обширная часть оптимизационных задач по некоторым литературным источникам составляет примерно 70% всех оптимизационных задач. Используется при решении экономических, военных, промышленных и организационных задач. Столь широкое использование объясняется доступностью математического обеспечения для задач большой размерности и наглядностью решений.

Решение любой задачи ЛП начинается с построения математической модели практической задачи. Это важный и трудоемкий процесс, своего рода искусство и интуиция.

Построение модели включает следующие три этапа.

- 1) Определение переменных задачи.
- 2) Представление ограничений в виде линейных уравнений или неравенств.
- 3) Задание линейной целевой функции и смысла оптимизации.

4.1. Классические задачи линейного программирования

Существует ряд классических задач ЛП, постановки и решения которых известны и формализованы.

4.1.1. Задача технического контроля

В отделе технического контроля (ОТК) некоторой фирмы работают контроллеры 1 и 2 разрядов.

Норма выработки ОТК за 8 часов (день) не менее 1800 изделий.

Контролер 1 разряда проверяет 25 изделий/час (точность 98%);

Контролер 2 разряда проверяет 15 изделий/час (точность 95%).

Заработная плата:

Контролер 1 разряда – 4\$ / час;

Контролер 2 разряда – 3\$ / час.

Ошибка контроллера приносит убыток фирме 2\$.

Фирма может использовать не более:

8 контроллеров 1 разряда;

10 контроллеров 2 разряда.

Требуется выполнить дневной план и минимизировать затраты фирмы.

Построение модели.

1. Введем переменные задачи.

x_1 - число контроллеров 1 разряда.

x_2 - число контроллеров 2 разряда.

2. Ограничения на переменные

Ограничения по условию задачи.

$$\begin{cases} x_1 \leq 8 \\ x_2 \leq 10 \end{cases}.$$

В день необходимо изготовить 1800 изделий (за 8 часов работы).

$$8 \cdot (25 \cdot x_1 + 15 \cdot x_2) \geq 1800,$$

$$200 \cdot x_1 + 120 \cdot x_2 \geq 1800,$$

$$5 \cdot x_1 + 3 \cdot x_2 \geq 45.$$

3. Задание линейной целевой функции

Расходы фирмы имеют две составляющие

- заработная плата контроллеров;
- убытки из-за ошибок контроллеров.

Таким образом, один контроллер соответствующего разряда обходится фирме

I разряд

$$4 + 2 \cdot 0.02 \cdot 25 = 5 \$ / \text{ час}.$$

II разряд

$$3 + 2 \cdot 0.05 \cdot 15 = 4.5 \text{ \$ / час.}$$

Запишем целевую функцию затрат на ОТК за 8 часов.

$$f(\bar{x}) = 8 \cdot (5 \cdot x_1 + 4.5 \cdot x_2) = 40 \cdot x_1 + 36 \cdot x_2$$

Т.о., вся задача технического контроля может быть сформулирована следующим образом.

$$\left. \begin{array}{l} f(\bar{x}) = 40 \cdot x_1 + 36 \cdot x_2 \rightarrow \min \\ \text{при } 5 \cdot x_1 + 3 \cdot x_2 \geq 45 \\ x_1 \leq 8, \quad x_2 \leq 10 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{array} \right\} . \quad (4.2)$$

4.1.2. Транспортная задача

О рациональном перевозе однородных продуктов из пунктов производства в пункты потребления.

В каждом пункте A_i производится a_i количество продукта, $i = \overline{1, m}$.

Пункт B_j потребляет b_j количества продукта, $j = \overline{1, n}$.

Предполагается, что спрос соответствует предложению

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j .$$

Транспортные издержки перевозки продукта из пункта A_i в пункт B_j составляют c_{ij} .

Требуется минимизировать транспортные издержки и удовлетворить запросы всех потребителей за счет производства.

Введем переменные

x_{ij} - количество продукта перевозимого из пункта A_i в B_j .

Математическая постановка задачи имеет вид.

$$\left. \begin{aligned} f(\bar{x}) &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \rightarrow \min \\ \sum_{j=1}^n x_{ij} &= a_i, \quad i = \overline{1, m} \\ \sum_{i=1}^m x_{ij} &= b_j, \quad j = \overline{1, n} \\ x_{ij} &\geq 0 \end{aligned} \right\}$$

4.1.3. Задача о диете

Имеется n различных продуктов. Стоимость каждого продукта составляет c_j .

Ингредиенты продуктов следующие:

- калорийность a_{1j} ($j = \overline{1, n}$);
- жиры a_{2j} ;
- белки a_{3j} ;
- углеводы a_{4j} .

Суточная потребность конкретного человека в энергии, жирах, белках и углеводах составляет b_1, b_2, b_3, b_4 единиц соответственно.

Требуется удовлетворить суточную потребность в энергии, не превышая потребления жиров, белков, углеводов при минимальных затратах.

Пусть x_j - количество потребления j -го продукта.

Математическая постановка задачи имеет вид.

$$\left. \begin{aligned} f(\bar{x}) &= \sum_{j=1}^n c_j \cdot x_j \rightarrow \min \\ \sum_{j=1}^n a_{1j} \cdot x_j &\geq b_1; \quad \sum_{j=1}^n a_{3j} \cdot x_j \geq b_3; \\ \sum_{j=1}^n a_{2j} \cdot x_j &\geq b_2; \quad \sum_{j=1}^n a_{4j} \cdot x_j \geq b_4. \end{aligned} \right\}$$

4.1.4. Задача об использовании сырья

Изготавливаются два продукта P_1 и P_2 из трех видов сырья s_1, s_2, s_3 .

Запасы каждого сырья равны b_1, b_2, b_3 .

На единицу продукции P_1 уходит a_{11} количества сырья s_1

$$a_{21} - s_2;$$

$$a_{31} - s_3.$$

На единицу продукции P_2 уходит $a_{12} - s_1;$

$$a_{22} - s_2;$$

$$a_{32} - s_3.$$

Требуется так запланировать выпуск продуктов P_1 и P_2 , чтобы доход от реализации продукции был максимален при имеющихся запасах сырья. Считается, что сбыт обеспечен.

Исходные данные задачи удобно представить в виде таблицы. В таблице 4.1 представлены данные для общей постановки задачи.

Таблица 4.1-

Исходные данные задачи об использовании сырья

Вид сырья	Запасы сырья	Расход сырья на единицу продукции	
		P_1	P_2
s_1	b_1	a_{11}	a_{12}
s_2	b_2	a_{21}	a_{22}
s_3	b_3	a_{31}	a_{32}
Доход		c_1	c_2

Пусть имеет место задача об использовании сырья некоторого кондитерского предприятия, выпускающего

P_1 - карамель А;

P_2 - карамель Б.

При этом используется сырье

s_1 - сахар;

s_2 - джем;

s_3 - шоколад.

Введем переменные.

x_1 единиц продукции Π_1 выпускает предприятие.

x_2 единиц продукции Π_2 выпускает предприятие.

В таблице 4.2 – данные по изготовлению кондитерских изделий. Отметим, что все данные приведены в условных единицах.

Таблица 4.2-

Исходные данные задачи выпуска кондитерских изделий

Вид сырья	Запасы сырья	Расход сырья на единицу продукции	
		карамель А	карамель Б
сахар	160	5	2
джем	180	3	4
шоколад	196	7	0
Доход		3	2

Тогда задача линейного программирования примет вид.

$$\left. \begin{aligned} f(\bar{x}) &= 3 \cdot x_1 + 2 \cdot x_2 \rightarrow \max \\ 5 \cdot x_1 + 2 \cdot x_2 &\leq 160 \\ 3 \cdot x_1 + 4 \cdot x_2 &\leq 180 \\ 7 \cdot x_1 &\leq 196 \\ x_1, x_2 &\geq 0 \end{aligned} \right\}.$$

Приведем графическое решение этой задачи.

В системе координат $X_1, 0, X_2$ построим ограничения-неравенства и линию уровня $3 \cdot x_1 + 2 \cdot x_2 = const$, например с константой $const=100$, рисунок 4.1. Затем линию уровня параллельно переносим в сторону увеличения целевой функции $f(x_1, x_2)$ до пересечения с крайней вершиной многогранника ограничений. В данном случае - вершина с координатой (20, 30).

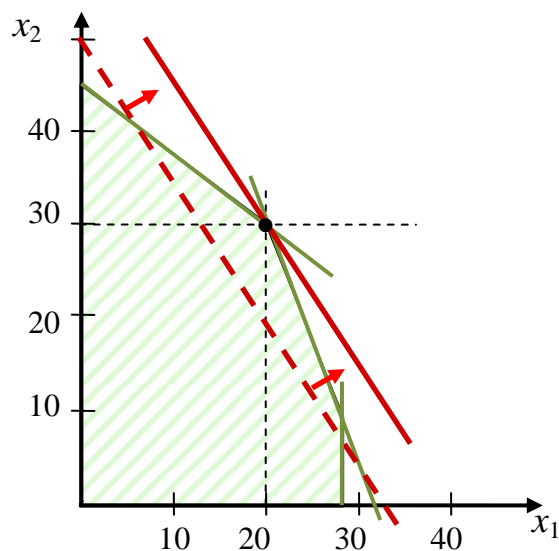


Рисунок 4.1. Графическое решение

4.2 Геометрический метод решения задач линейного программирования

Геометрический метод решения будет подробно рассмотрен на примере задачи технического контроля. На практике графический метод решения используется крайне редко. Однако в примере для двух переменных хорошо иллюстрируется механизм решения задач линейного программирования.

Итак, задача технического контроля (4.2) содержит только две переменные и не содержит ограничений типа равенства, следовательно, такую задачу можно решить графически.

$$\left. \begin{aligned} f(\bar{x}) &= 40 \cdot x_1 + 36 \cdot x_2 \rightarrow \min \\ 5 \cdot x_1 + 3 \cdot x_2 &\geq 45 \\ x_1 &\leq 8, \quad x_2 \leq 10 \\ x_1 &\geq 0, \quad x_2 \geq 0 \end{aligned} \right\}$$

Все ограничения линейные, поэтому любое из них определяет полуплоскость допустимых решений, лежащих по одну сторону прямой $a_{i1} \cdot x_1 + a_{i2} \cdot x_2 = b_i$.

Построив все ограничения-неравенства, получим множество (многогранник) допустимых решений.

Заштрихованный треугольник – это и есть множество допустимых решений, рисунок 4.2.

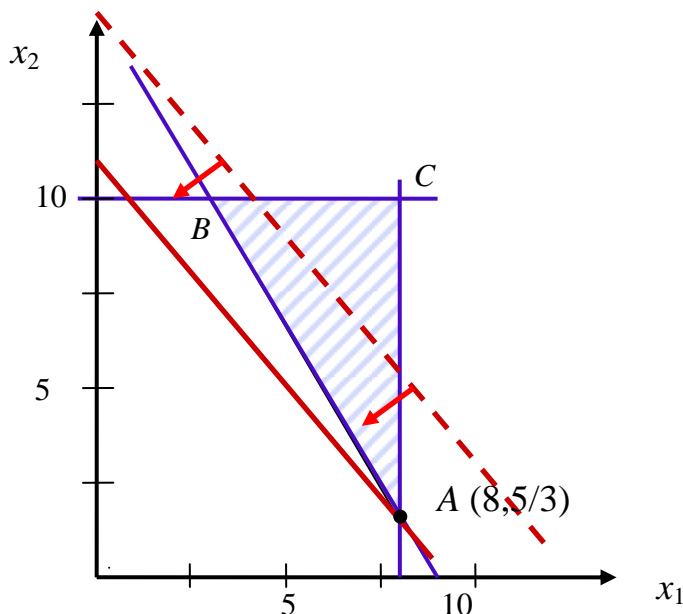


Рисунок 4.2. Геометрическое решение

Для построения целевой функции рассмотрим линии уровня $40 \cdot x_1 + 36 \cdot x_2 = const$. Построим линию уровня для любой константы, например $const = 600$, что соответствует значениям координат $x_1 = 12, x_2 = 15$. Линия уровня попала в область допустимых решений. Теперь если изменять константу, то линии уровня будут перемещаться параллельно. И смещать надо в сторону увеличения константы до тех пор, пока линия уровня не достигнет крайней вершины многогранника допустимых решений. В нашем случае это будет вершина треугольника А с координатами $\left(8, \frac{5}{3}\right)$, $f(\bar{x}) = f\left(8, \frac{5}{3}\right) = 377.6$.

Таким образом, для оптимальной работы ОТК необходимо использовать 8 контроллеров 1 разряда и 1.6 контроллеров 2 разряда. Значение 1.6 означает или неполный рабочий день, или округление до двух контроллеров.

Отметим, что координаты двух других вершин $C(8, 10)$, $f(\bar{x}) = f(8, 10) = 680$; $B(10, 3)$, $f(\bar{x}) = f(10, 3) = 508$. Значения целевой функции в этих вершинах многогранника ограничений больше, чем в вершине А.

В данном примере решение существует и оно единственно.

Некоторые частные случаи геометрических решений рассмотрены ниже в примере I и II.

Пример I.

$$\left. \begin{aligned} f(\bar{x}) &= -x_1 - 2 \cdot x_2 \rightarrow \min \\ x_1 + 2 \cdot x_2 &\leq 7 \\ 2 \cdot x_1 + x_2 &\leq 8 \\ x_2 &\leq 3 \\ x_1, x_2 &\geq 0 \end{aligned} \right\}.$$

На рисунке 4.3-а приведен многогранник ограничений и целевая функция. Целевая функция параллельна стороне CD многогранника ограничений. Убывает целевая функция в направлении стороны CD . Следовательно, при наложении целевой функции на сторону CD и будет найден минимум целевой функции. То есть задача имеет бесконечное множество решений, в число которых входят и вершины C и D

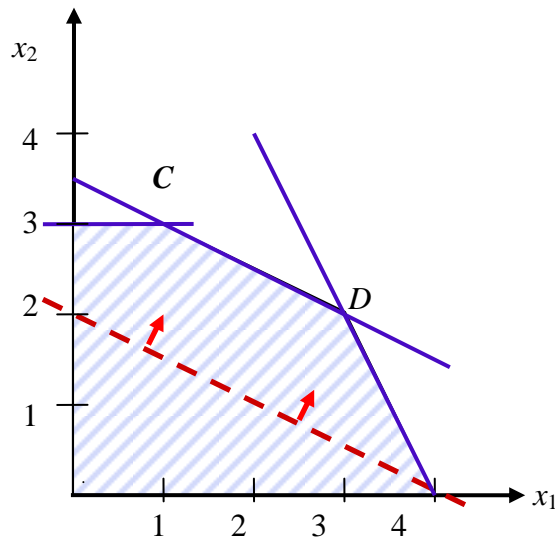
Пример II

$$\left. \begin{aligned} f(\bar{x}) &= -x_1 - 2 \cdot x_2 \rightarrow \min \\ x_1 + x_2 &\geq 1 \\ 2 \cdot x_1 - x_2 &\geq -1 \\ x_1 - 2 \cdot x_2 &\leq 0 \\ x_1, x_2 &\geq 0 \end{aligned} \right\}$$

Геометрическое решение задачи приведено на рисунке 4.3-б.

В направлении уменьшения целевой функции нет ограничений. При параллельном переносе целевой функции она всегда пересекает множество допустимых решений и неограниченно убывает, следовательно, решений задача не имеет.

а



б

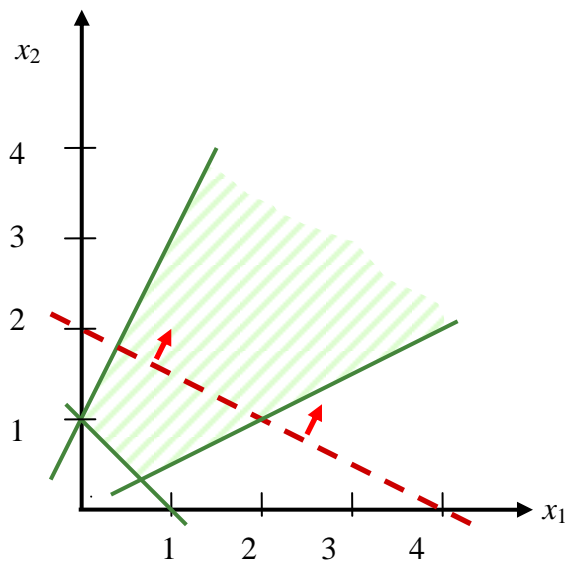


Рисунок 4.3. Частные случаи решения задач ЛП

Ситуация II на практике невозможна. В противоположном случае, можно было бы получить бесконечный доход при конечных ресурсах. Если в практической задаче получено аналогичное решение, следовательно, при построении модели пропущено ограничение.

Примеры на рисунках 4.1 и 4.2 иллюстрируют следующее свойство задач линейного программирования. Если в задаче линейного программирования существует оптимальное решение, то это решение находится в одной из вершин допустимой.

4.3. Задача линейного программирования в стандартной форме

Задача линейного программирования в стандартной форме содержит следующие элементы.

- а) ограничение только в виде равенств;
- б) все переменные x_j , $j = \overline{1, n}$; ограничены;
- с) ресурсы $b_i \geq 0$, $i = \overline{1, m}$;
- д) ЦФ $\rightarrow \min$.

$$\begin{aligned}
 f(\bar{x}) &= \sum_{j=1}^n c_j \cdot x_j \rightarrow \min \\
 \sum_{j=1}^n a_{ij} \cdot x_j &= b_i, \quad i = \overline{1, m}, \\
 x_j &\geq 0, \quad j = \overline{1, n}, \\
 b_i &\geq 0, \quad i = \overline{1, m}.
 \end{aligned} \tag{4.4}$$

Форма записи (4.4) носит также название стандартного вида. Часто используется матрично– векторная запись формулы (4.4).

$$\begin{aligned}
 f(\bar{x}) &= (\bar{c}, \bar{x}) \rightarrow \min \\
 A \cdot \bar{x} &= \bar{b} \\
 \bar{x} &\geq 0, \quad \bar{b} \geq 0,
 \end{aligned} \tag{4.5}$$

где A – матрица коэффициентов;

\bar{x} – вектор переменных;

\bar{b} – вектор ресурсов;

\bar{c} – вектор оценок задачи линейного программирования.

Решение задачи линейного программирования требует приведение ее к стандартному виду. Рассмотрим технологию приведения задачи линейного программирования к стандартному виду.

I. Преобразование неравенств.

Ограничения – неравенства можно преобразовать в равенства при помощи введения, так называемых, остаточных и избыточных переменных x_{n+i} . Остаточные переменные вводятся со знаком «плюс», избыточные со знаком «минус».

$$\sum_{j=1}^n a_{ij} \cdot x_j \leq b_i;$$

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

Здесь мы все неравенства преобразовали в равенства.

Пример I

$$x_1 + 2 \cdot x_2 + 3 \cdot x_3 + 4 \cdot x_4 \leq 25;$$

$$x_1 + 2 \cdot x_2 + 3 \cdot x_3 + 4 \cdot x_4 + x_5 = 25,$$

где $x_5 > 0$ - остаточная переменная.

Пример II

$$2 \cdot x_1 + x_2 - 3 \cdot x_3 \geq 12;$$

$$2 \cdot x_1 + x_2 - 3 \cdot x_3 - x_4 = 12,$$

где $x_4 > 0$ - избыточная переменная.

Преобразуем неравенства из задачи об использовании сырья (4.3), добавив во все три уравнения остаточную переменную.

$$\left. \begin{array}{l} 5 \cdot x_1 + 2 \cdot x_2 \leq 160 \\ 3 \cdot x_1 + 4 \cdot x_2 \leq 180 \\ 7 \cdot x_1 \leq 196 \end{array} \right\}.$$

$$\left. \begin{array}{l} 5 \cdot x_1 + 2 \cdot x_2 + x_3 = 160 \\ 3 \cdot x_1 + 4 \cdot x_2 + x_4 = 180 \\ 7 \cdot x_1 + x_5 = 196 \end{array} \right\}.$$

II. Преобразование неограниченных по знаку переменных

В некоторых задачах ряд переменных могут принимать как отрицательные, так и положительные значения, либо о них ничего неизвестно, то есть переменные не определены.

Однако переменные задачи линейного программирования в стандартной форме предполагаются только неотрицательными. Неограниченные переменные необходимо заменить разностью двух положительных переменных.

$$x_j = x_{n+1} - x_{n+2}.$$

Здесь $x_{n+1}, x_{n+2} > 0$. Переменные x_j может быть положительным и отрицательным в зависимости от соотношения x_{n+1}, x_{n+2} .

III. Устранение отрицательных ресурсов.

Уравнения с отрицательными значениями ресурсов b_i необходимо умножить на (-1).

Ниже рассмотрен пример приведения задач ЛП к стандартному виду.

Пример. Привести к стандартному виду

$$\begin{aligned} f(\bar{x}) &= x_1 - 2 \cdot x_2 + 3 \cdot x_3 \rightarrow \min \\ x_1 + x_2 + x_3 &\leq 7 & 1 \\ x_1 - x_2 + x_3 &\geq 2 & 2 \\ 3 \cdot x_1 - x_2 - 2 \cdot x_3 &= -5 & 3 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Преобразуем задачу линейного программирования к стандартной форме.

1. Переменная x_3 - неограниченна.

$$x_3 = x_4 - x_5, \text{ где } x_4, x_5 \geq 0.$$

2. Умножим на (-1) уравнение (3).

3. Введем дополнительные переменные: остаточную x_6 в уравнение (1) и избыточную переменную x_7 в уравнение (2).

Перепишем задачу ЛП с учетом новых переменных и замен.

$$\left. \begin{aligned} f(\bar{x}) &= x_1 - 2 \cdot x_2 + 3 \cdot x_4 - 3 \cdot x_5 \rightarrow \min \\ x_1 + x_2 + x_4 - x_5 + x_6 &= 7 \\ x_1 - x_2 + x_4 - x_5 - x_7 &= 2 \\ -3 \cdot x_1 + x_2 + 2 \cdot x_4 - 2 \cdot x_5 &= 5 \\ x_1, x_2, x_4, x_5, x_6, x_7 &\geq 0 \end{aligned} \right\}.$$

Получен стандартный вид задачи ЛП.

4.4. Идея и алгебра симплекс – метода

Симплекс – метод является общим методом решения задач линейного программирования. Запишем задачу линейного программирования в стандартной форме в развернутом виде.

$$\left. \begin{aligned} f(\bar{x}) &= c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n \rightarrow \min \\ a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n &= b_1 \\ . &. \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \dots + a_{mn} \cdot x_n &= b_m \\ x_1, \dots, x_n &\geq 0 \end{aligned} \right\}. \quad (4.6)$$

Система ограничений представляет собой систему линейных алгебраических уравнений относительно неизвестных x_i , $i=1, n$. Особенностью задач линейного программирования является то, что число уравнения системы меньше числа неизвестных ($m < n$). То есть СЛАУ представляет собой недоопределенную систему и множество допустимых решений бесконечно. Отсюда следует, что выбор оптимального допустимого решения нетривиален.

Классическим методом решения недоопределенных СЛАУ является метод **Гаусса – Жордана**. Основная идея его состоит в сведении m уравнения с n неизвестными к каноническому или ступенчатому виду, путем линейных преобразования над строками (сложение строк и умножение на скаляр). Это позволяет привести СЛАУ к следующему виду.

$$\left. \begin{aligned} x_1 + \dots + \bar{a}_{1m+1} \cdot x_{m+1} + \dots + \bar{a}_{1n} \cdot x_n &= \bar{b}_1 \\ x_2 + \dots + \bar{a}_{2m+1} \cdot x_{m+1} + \dots + \bar{a}_{2n} \cdot x_n &= \bar{b}_2 \\ \dots &\dots \\ x_m + \dots + \bar{a}_{mm+1} \cdot x_{m+1} + \dots + \bar{a}_{mn} \cdot x_n &= \bar{b}_m \end{aligned} \right\}. \quad (4.7)$$

Переменные x_1, x_2, \dots, x_m называются базисными (зависимыми), которые входят в одно уравнение с коэффициентом 1, а в остальные с коэффициентом 0.

x_{m+1}, \dots, x_n называют свободными (независимыми).

В канонических системах в любом уравнении присутствует только одна базисная переменная.

Выразим из системы (4.7) базисные переменные:

$$\left. \begin{aligned} x_1 &= \bar{b}_1 - \bar{a}_{1m+1} \cdot x_{m+1} - \dots - \bar{a}_{1n} \cdot x_n \\ \dots &\dots \\ x_m &= \bar{b}_m - \bar{a}_{mm+1} \cdot x_{m+1} - \dots - \bar{a}_{mn} \cdot x_n \end{aligned} \right\}.$$

Из этой системы можно получить решение для базисных переменных, присваивая независимым переменным произвольные значения.

Определение. Базисным решением системы канонического вида называют решение, полученное при нулевых значениях независимых переменных. То есть

$$x_1 = \bar{b}_1, \dots, x_m = \bar{b}_m, x_{m+1} = 0, \dots, x_n = 0.$$

Определение. Если $\bar{b}_i > 0, i = \overline{1, m}$, то полученное базисное решение называется допустимым базисным решением или **опорным планом**.

Отметим, что базисными переменными не обязательно являются первые m переменных. Ими могут быть любые m штук переменных системы.

Ниже приведены примеры использования основных модификаций метода Гаусса. Самыми популярными из них являются: метод последовательного исключения Гаусса со схемой единственного деления и с выбором главных элементов.

4.4.1. Метод последовательного исключения переменных (метод Гаусса)

Существует несколько модификаций метода Гаусса, самая известная из них – схема единственного деления.

Метод Гаусса состоит из двух этапов.

I. Прямой ход.

Цель прямого хода - приведение матрицы системы A к верхнетреугольному виду.

Для этого выполняется n преобразований матрицы A . На каждом шаге преобразования выбирается k -я главная или ведущая строка. Диагональные элементы главной строки a_{kk} называются главными (ведущими) элементами.

На прямом ходе выполняется следующие действия:

- для всех строк, кроме главной, находим множитель l_i ;

$$l_i = -\frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}.$$

- к каждой неглавной добавляем главную строку, умноженную на множитель l_i ;

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} + l_{ik} \cdot a_{kj}^{(k-1)};$$

$$b_i^{(k)} = b_i^{(k-1)} + l_{ik} \cdot b_k^{(k-1)};$$

- главную строку делим на главный элемент a_{kk} ;
- главную строку вычеркиваем, размерность системы становится меньше на единицу, то есть $(n-1)$.

На следующем шаге исключения главной строкой вновь становится верхняя строка, находящаяся под вычеркнутой на предыдущем шаге. Все указанные выше преобразования повторяются. Преобразования повторяются столько раз, пока главная строка не становится единственной в системе.

В результате матрица A приведена к верхнетреугольному виду.

II. Обратный ход.

Вектор неизвестных СЛАУ \bar{x} находится в обратном порядке, начиная с последнего. Для этого составляется матрица из вычеркнутых строк, она имеет верхнетреугольный вид. Из последнего уравнения находится неизвестный x_n , затем неизвестные находятся в порядке $x_{n-1}, x_{n-2}, \dots, x_1$.

Схему единственного деления Гаусса удобно реализовать в виде таблицы. В таблице 4.2 приведено решение следующей СЛАУ

$$\left. \begin{aligned} 3 \cdot x_1 - x_2 &= 5 \\ -2 \cdot x_1 + x_2 + x_3 &= 0 \\ 2 \cdot x_1 - x_2 + 4 \cdot x_3 &= 15 \end{aligned} \right\}.$$

Таблица 4.2

Метод Гаусса, схема единственного деления

	k	l_i	Коэффициенты при неизвестных			b	c
			x_1	x_2	x_3		
Прямой ход	0	$\frac{2}{3}$ $-\frac{2}{3}$	3	-1	0	5	7
			-2	1	1	0	0
			2	-1	4	15	20
	1	1	3	-1	0	5	7
			0	$\frac{1}{3}$	1	$\frac{10}{3}$	$\frac{14}{3}$
			0	$-\frac{1}{3}$	4	$\frac{35}{3}$	$\frac{46}{3}$
	2	0	3	-1	0	5	7
			0	$\frac{1}{3}$	1	$\frac{10}{3}$	$\frac{14}{3}$
			0	0	5	$\frac{45}{3}$	$\frac{60}{3}$
Обратный ход			$x_1=1$	$x_2=1$	$x_3=3$		$\bar{x}_3=4$ $\bar{x}_2=2$ $\bar{x}_1=3$

Замечание. Для уменьшения возможности ошибок счета вводятся, так называемые, контрольные суммы (столбец в таблице 4.2), с которым выполняются следующие преобразования.

а) на прямом ходе те же преобразования, что со столбцом свободных членов b . Контроль правильности преобразования строки на очередном шаге проводится суммированием всех коэффициентов в строке и свободного члена. Эти значения должны быть равны.

б) на обратном ходе одновременно с вычислением корней x_i , вычисляются корни \bar{x}_i , которые получены если в выделенном в системе

уравнении вместо свободного члена b_i использовать значение контрольной суммы c_i . Между корнями должно выполняться такое соответствие $\bar{x}_i = x_i + 1$.

При реализации метода Гаусса по схеме единственного деления предполагалось, что диагональные элементы не равны нулю. Это можно гарантировать только на нулевом шаге ($k=0$) в исходной системе. Однако в ходе преобразований вполне возможны ситуации, что главные диагональные элементы станут нулевыми. В этих условиях схема единственного деления становится неработоспособной, так как на главные элементы в ходе преобразований производится деление. Выходом в такой ситуации является использование модификации метода Гаусса – метод главных элементов.

Основная идея метода

1. На каждом шаге выбирают главный элемент матрицы A – максимальный по модулю коэффициент в матрице a_{pq} . Строка p – главная строка.

2. Для всех строк, кроме главной, вычисляются множители

$$m_i = -\frac{a_{iq}}{a_{pq}}.$$

3. К каждой неглавной строке добавляют главную, умноженную на сомножитель m_i . В результате q -й столбец – нулевой.

4. Вычеркиваем p -ю строку и q -й столбец. В результате получаем матрицу A^1 , размерность которой на единицу меньше предыдущей матрицы A .

5. Процедуру повторяем с первого шага $(n-1)$ раз.

6. В результате составляем новую систему уравнений из вычеркнутых строк. Полученная матрица не треугольного вида, как в схеме единственного деления Гаусса. Но каждое уравнение содержит разное количество неизвестных: в последнем уравнении – 1 неизвестное, в $(n-1)$ -м – 2 неизвестных и т.д., в первом уравнении – все n неизвестных. Такой вид преобразованной системы позволяет последовательно находить неизвестные, начиная с последнего уравнения.

Решение той же самой системы методом Гаусса с выбором главного элемента представлено в таблице 4.3. В таблице выделены главные элементы и главные строки.

Таблица 4.3

Метод Гаусса с выбором главного элемента

	k	m_i	Коэффициенты при неизвестных			b	c
			x_1	x_2	x_3		
Прямой ход	0	0	3	-1	0	5	7
		-1/4	-2	1	1	0	0
		-1	2	-1	4	15	20
	1	-1	3	-1	-	5	7
		5/6	-5/2	5/4	-	-15/4	-5
			-	-	-	-	-
	2		-	5/12	-	5/12	5/6
			-	-	-	-	-
			-	-	-	-	-
Обратный ход			$x_1=2$	$x_2=1$	$x_3=3$		$\bar{x}_2=2$
							$\bar{x}_1=3$
							$\bar{x}_3=4$

Симплекс-метод тесно связан с методом Гаусса с выбором главного элемента и основан на ряде утверждений, декларируемых в виде следующих теорем.

Теорема 1. Множество опорных планов – выпукло.

Теорема 2. Если задача линейного программирования разрешима, то экстремум целевой функции достигается на одном из опорных планов (допустимом базисном решении).

То есть оптимальное решение соответствует крайней точке выпуклого множества.

Число опорных планов конечно и определяется числом

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Основная идея симплекс-метода состоит в направленном переборе опорных планов с последовательным уменьшением целевой функции. Принадлежит эта идея двум авторам: американскому математику

Дж.Данцигу (1947 год) и российскому математику Л.В.Канторовичу (1939 год).

Алгоритм симплекс – метода включает три основных этапа.

- I. Выбор начального опорного плана.
- II. Переход от начального опорного плана к другому опорному плану с лучшим значением целевой функции.
- III. Продолжение поиска опорного плана улучшающего целевую функцию до достижения оптимального плана. Приведение системы, для смежного опорного плана к каноническому виду

Определение. Смежным опорным планом называют план, отличающийся от текущего лишь одной переменной.

Таким образом, в получении смежных опорных планов состоит механизм симплекс метода.

Для получения смежного опорного плана одну базисную переменную превращают в свободную, а эту свободную вводят в базисные переменные. Основной здесь вопрос – какую переменную выбрать, чтобы целевая функция уменьшалась. Симплекс-метод позволяет целенаправленно получать смежные планы, избегая простого перебора вершин. Рассмотрим этот механизм.

Составим симплекс – таблицу для задачи использования ресурсов (4.3). Для этого запишем задачу в каноническом виде и заменим цель поиска максимум на минимум.

$$\left. \begin{array}{l} z(\bar{x}) = -3 \cdot x_1 - 2 \cdot x_2 \rightarrow \min \\ 5 \cdot x_1 + 2 \cdot x_2 + x_3 = 160 \\ 3 \cdot x_1 + 4 \cdot x_2 + x_4 = 180 \\ 7 \cdot x_1 + x_5 = 196 \end{array} \right\}.$$

Начальный опорный план и начальная симплекс – таблица приведены в таблице 4.4.

Таблица 4.4

Начальный опорный план

Базис	Свободные переменные					b_i	Q_{ik}	$\Delta_1 = -3 - x_4 \begin{bmatrix} 0 & 5 \\ 0 & 3 \\ 0 & 7 \end{bmatrix} = -3$ $\Delta_2 = -2 - \begin{bmatrix} 0 & 2 \\ 0 & 4 \\ 0 & 0 \end{bmatrix} = -2$
	x_1	x_2	x_3	x_4	x_5			
x_3	5	2	1	0	0	160	32	
x_4	3	4	0	1	0	180	60	
x_5	<u>7</u>	0	0	0	1	196	28	
Δ_k	-3	-2	-	-	-			

I. Итак, это начальный опорный план \bar{x}^{-1} . Переменные x_3, x_4, x_5 - базисные, x_1, x_2 - свободные. Свободные – это значит, что они могут принимать любые значения, но чаще их выбирают нулевыми.

$$x_1 = 0, x_2 = 0 \rightarrow$$

$$x_3 = 160, x_4 = 180, x_5 = 196$$

Сам опорный план

$$\bar{x}^{-1} = (0, 0, 160, 180, 196)$$

Целевая функция

$$z(\bar{x}^{-1}) = 0.$$

II. Нахождение смежного плана.

Его цель: одну переменную из свободных перевести в базис, так, чтобы целевая функция уменьшалась.

Для овладения механизмом получения смежных опорных планов необходимо ввести ряд соотношений. Для этого целесообразно записать задачу ЛП в стандартной векторно-матричной форме. Одновременно введем несколько определений.

$$f(\bar{x}) = \sum_{j=1}^n c_j \cdot x_j \rightarrow \min$$

$$\sum_{j=1}^n a_{ij} \cdot x_j = b_i, i = \overline{1, m}$$

$$x_j \geq 0 \quad j = \overline{1, n}$$

$$b_i \geq 0 \quad i = \overline{1, m}$$

Определение. Приращение целевой функции Δ_k это величина

$$\Delta_k = c_k - \sum_{i \in \text{баз}} c_i \cdot a_{ik}, \quad (4.8)$$

где $k \in$ свободным переменным, в нашем примере $k=1, k=2$.

$\sum_{i \in \text{баз}}$ - означает, что суммирование ведется только по базисным переменным, но для столбца свободных переменных.

Приращения вычисляются только для свободных переменных, то есть индекс $k \in$ свободным.

Δ_k - важнейшая величина, по которой и делают заключение о том, какая из свободных переменных при переводе ее в базисную, дает уменьшение целевой функции.

Это заключение основывается на следующей лемме.

Лемма 1. Если все приращения симплекс-плана $\Delta_k \geq 0$, следовательно, текущий опорный план оптимальный.

Симплекс – метод использует следствие этой леммы.

Улучшение целевой функции приносит та свободная переменная x_k , приращение которой отрицательно $\Delta_k < 0$.

Так как в текущем опорном плане может быть несколько отрицательных Δ_k , то выбирать для замены можно любую переменную x_k , но обычно выбирают переменную x_k с минимальным $\Delta_k < 0$.

В нашем примере такой переменной является x_1 . Строго говоря, это может быть и x_2 , так как $\Delta_2 < 0$.

Другая важнейшая величина симплекс – алгебры – это отношение

$$\theta_i = \frac{b_i}{a_{ik}}. \quad (4.9)$$

Величины θ_i вычисляются для выбранного k – го столбца (с минимальным значением Δ_k). Величина θ_i определяет, какая из базисных переменных уйдет в свободные на место x_k . При этом, отношение вычисляется только для $a_{ik} > 0$. Выбирается та переменная x_1 , для которой отношение θ_i минимально.

Теоретической основой этого шага служит лемма.

Лемма 2. Если хотя бы одно из приращений $\Delta_k < 0$ и при этом среди коэффициентов a_{ik} есть хотя бы один положительный, то существует опорный план улучшающий текущий.

Выбранная для данного примера переменная x_5 меняется местами с x_1 . Элемент, стоящий на пересечении выбранного столбца и выбранной строки называется ведущим, а вся строка – ведущей строкой.

III. Приведение системы к каноническому виду.

Так как переменную x_1 ввели в базис, то она должна в третьем уравнении иметь коэффициент равный единице, а во всех остальных нулю. Записав второй столбец $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, осуществим эквивалентные преобразования с первой строкой системы.

Преобразование основано на методе Гаусса с выбором главного элемента.

К каждой неглавной строке добавляем главную, умноженную на коэффициент $m_i = -\frac{a_{ik}}{a_{pk}}$,

где a_{ik} - главный или ведущий элемент;

k – номер столбца ведущего элемента;

p – строка ведущего элемента.

Таким образом, можно записать следующий смежный план

$$\bar{x}_2 = 28, 0, 20, 96, 0,$$

$$f(\bar{x}_2) = -3 \cdot 28 = -84.$$

Полностью симплекс- таблица для решения задачи приведена в таблице 4.5. На каждом шаге выделен ведущий элемент и выбранное приращение.

Таблица 4.5

Симплекс таблица

m_i		Переменные						b	Q
		Базис	x_1	x_2	x_3	x_4			
I	-5/7	x_3	5	2	1	0	0	160	32
	-3/7	x_4	3	4	0	1	0	180	60
		x_5	7	0	0	0	1	196	28
		Δ_k	-3	-2	-	-	-		
II		x_3	0	2	1	0	-5/7	20	10
	-2	x_4	0	4	0	1	-3/7	96	24
	0	x_1	1	0	0	0	1/7	28	-
		Δ_k	-	-2	-	-	3/7		
III	5/14	x_2	0	1	1/2	0	-5/14	10	-
		x_4	0	0	-2	1	1	56	56
	-1/7	x_1	1	0	0	0	1/7	28	196
		Δ_k	-	-	1	-	-2/7		
IV		x_2	0	1	-3/14	5/14	0	30	
		x_5	0	0	-2	1	1	56	
		x_1	1	0	2/7	-1/7	0	20	
		Δ_k	-	-	6/14	4/14	-		

Запишем последовательности опорных планов.

Шаг 1.

$$\bar{x}_1 = 0, 0, 160, 180, 196, \quad f(\bar{x}_1) = 0.$$

$$\Delta_1 = -3 - x_4 \begin{bmatrix} x_3 & 0 & 5 \\ 0 & 3 \end{bmatrix} = -3, \quad \Delta_2 = -2 - \begin{bmatrix} 0 & 2 \\ 0 & 4 \\ 0 & 0 \end{bmatrix} = -2.$$

Шаг 2.

$$\bar{x}_2 = 28, 0, 20, 96, 0, \quad f(\bar{x}_2) = -3 \cdot 28 = -84.$$

$$\Delta_2 = -2 - x_4 \begin{bmatrix} x_3 & 0 & 2 \\ 0 & 4 \\ x_1 & -3 & 0 \end{bmatrix} = -2, \quad \Delta_5 = 0 - \begin{bmatrix} 0 & -5/7 \\ 0 & -3/7 \\ -3 & 1/7 \end{bmatrix} = -3/7.$$

Шаг 3.

$$\bar{x}_3 = 28, 10, 0, 56, 0, \quad f(\bar{x}_3) = -104.$$

$$\Delta_3 = 0 - x_4 \begin{bmatrix} x_2 & -2 & 1/2 \\ 0 & -2 \\ x_1 & -3 & 0 \end{bmatrix} = 1, \quad \Delta_5 = 0 - \begin{bmatrix} -2 & -5/14 \\ 0 & 1 \\ -3 & 1/7 \end{bmatrix} = -2/7.$$

Шаг 4.

$$\bar{x}_4 = 20, 30, 0, 0, 56, \quad f(\bar{x}_4) = -120 - \text{оптимальный план.}$$

$$\Delta_3 = 0 - x_5 \begin{bmatrix} x_2 & -2 & -3/14 \\ 0 & -2 \\ x_1 & -3 & 2/7 \end{bmatrix} = 6/14, \quad \Delta_4 = 0 - x_5 \begin{bmatrix} x_2 & -2 & 5/14 \\ 0 & 1 \\ x_1 & -3 & -1/7 \end{bmatrix} = 4/14.$$

Так как все приращения положительны $\Delta_3, \Delta_4 > 0$, следовательно, опорный план нельзя улучшить.

Замечание.

Если на шаге I выбрать $\Delta_k = -2$ - разрешающим столбцом, то стратегия поиска такова:

$$\bar{x}_2 = 0, 45, 70, 196, 0 \quad f(\bar{x}_2) = -90;$$

$$\bar{x}_3 = 20, 30, 0, 0, 56 \quad f(\bar{x}_3) = -120.$$

Оптимум достигнут за 2 шага.

4.4.2. Метод искусственного базиса

Необходимым условием применения симплекс – метода является наличие опорного плана, то есть допустимого базисного решения канонической системы уравнений. Для этого должны выполняться следующие условия:

- система должна иметь каноническую (ступенчатую) структуру;
- присутствуют только ограничения - равенства;
- правые части ограничений положительны;
- переменные задачи положительны.

Без этих условий нельзя получить опорный план. Однако в реальных задачах далеко не всегда выполняются перечисленные условия.

Существует специальный метод, называемый искусственным базисом, который позволяет в любой задаче линейного программирования получить начальный опорный план.

Пусть задача линейного программирования приведена к стандартному виду:

$$\left. \begin{aligned} f(\bar{x}) &= \sum_{j=1}^n c_j \cdot x_j \rightarrow \min \\ \sum_{j=1}^n a_{ij} \cdot x_j &= b_i, \quad i = \overline{1, m} \end{aligned} \right\}.$$

Пусть все $b_i \geq 0$, но часть или все базисные переменные отрицательны, $x_j < 0$. Следовательно, опорного плана нет.

Дополним уравнения – ограничения искусственными переменными (предполагаем, что все $x_j < 0 \quad j = \overline{1, n}$).

Введем m переменных (по количеству уравнений) $x_{n+i} < 0 \quad i = \overline{1, m}$, которые в новой системе будут базисными, а отрицательные $x_j < 0$ уйдут в свободные.

В результате получим следующую эквивалентную задачу.

$$\left. \begin{aligned} \hat{f}(\bar{x}) &= \sum_{i=1}^m x_{n+i} \rightarrow \min \\ \sum_{j=1}^n a_{ij} \cdot x_j + x_{n+i} &= b_i, \quad i = \overline{1, m} \\ x_k &> 0, \quad k = \overline{1, n+m} \end{aligned} \right\}. \quad (4.10)$$

Здесь переменные x_{n+i} не имеют никакого отношения к исходной задаче линейного программирования и служат лишь для получения опорного плана и называются искусственными переменными. А новая целевая функция $\hat{f}(\bar{x})$ сформирована для полноты задачи.

В оптимальном опорном плане искусственные переменные должны быть равны нулю. В противном случае нарушится условие первоначальной задачи.

В начальном опорном плане искусственные переменные являются базисными, то есть не равны нулю, а в оптимальном плане искусственные переменные должны быть равны нулю. Значит, искусственные переменные должны стать в оптимальном плане свободными. В этом переводе и состоит основная идея метода: перевод искусственных переменных из базисных переменных в свободные. Рассмотрим механизм такого перевода на примере.

Пример

$$\left. \begin{aligned} f(\bar{x}) &= x_1 + 2 \cdot x_2 \rightarrow \max \\ 2 \cdot x_1 + x_2 &\geq 5 \\ x_1 + 3 \cdot x_2 &\geq 5 \\ -2 \cdot x_1 + 3 \cdot x_2 &\leq 7 \\ x_1 + x_2 &\leq 9 \\ x_1, x_2 &\geq 0 \end{aligned} \right\}.$$

Перепишем ЗЛП в стандартной форме. Для этого введем дополнительные переменные x_3, x_4, x_5, x_6 и запишем задачу в канонической форме.

$$\left. \begin{array}{l} f(\bar{x}) = -x_1 - 2 \cdot x_2 \rightarrow \min \\ 2 \cdot x_1 + x_2 - x_3 = 5 \\ x_1 + 3 \cdot x_2 - x_4 = 5 \\ -2 \cdot x_1 + 3 \cdot x_2 + x_5 = 7 \\ x_1 + x_2 + x_6 = 9 \\ x_i \geq 0, i = \overline{1,6} \end{array} \right\}.$$

Свободные переменные $x_1, x_2 = 0$, при этом базисные переменные примут значения $x_3 = -5, x_4 = -5, x_5 = 7, x_6 = 9$. Так как часть базисных переменных отрицательны, следовательно опорного плана нет. Для получения начального опорного плана введем переменные x_7, x_8 в двух первых уравнениях-ограничениях и сформулируем вспомогательную задачу:

$$\left. \begin{array}{l} \hat{f}(\bar{x}) = x_7 + x_8 \rightarrow \min \\ 2 \cdot x_1 + x_2 - x_3 + x_7 = 5 \\ x_1 + 3 \cdot x_2 - x_4 + x_8 = 5 \\ -2 \cdot x_1 + 3 \cdot x_2 + x_5 = 7 \\ x_1 + x_2 + x_6 = 9 \\ x_i \geq 0, i = \overline{1,8} \end{array} \right\}.$$

Таким образом, начальным базисом является

$$\bar{x}^{-1} = 0, 0, 0, 0, 7, 9, 5, 5.$$

Таблица 4.6

Симплекс таблица с искусственным базисом

m_i	Ба- зис	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	b	θ
I											
$-\frac{1}{3}$	x_7	2	1	-1	0	0	0	1	0	5	5
0	x_8	1	3	0	-1	0	0	0	1	5	5/3
-1	x_5	-2	3	0	0	1	0	0	0	7	7/3
-1/3	x_6	1	1	0	0	0	1	0	0	9	9

	$\hat{\Delta}_k$	-3	<u>-4</u>	1	1	-	-	-	-		
II											
	x_7	5/3	0	-1	1/3	0	0	1	$-\frac{1}{3}$	10/3	2
-1/5	x_2	1/3	1	0	$-\frac{1}{3}$	0	0	0	1/3	5/3	5
-9/5	x_5	-3	0	0	1	1	0	0	-1	2	-
-2/5	x_6	2/3	0	0	1/3	0	1	0	$-\frac{1}{3}$	22/3	11
	$\hat{\Delta}_k$	<u>-5/3</u>	-	1	$-\frac{1}{3}$	-	-	-	4/3		
III											
	x_1	1	0	-3/5	1/5	0	0	3/5	$-\frac{1}{5}$	2	
	x_2	0	1	1/5	$-\frac{2}{5}$	0	0	$-\frac{1}{5}$	$\frac{2}{5}$	1	
	x_3	0	0	-9/5	8/5	1	0	9/5	$-\frac{8}{5}$	8	
	x_6	0	0	2/5	1/5	0	1	$-\frac{2}{5}$	$-\frac{1}{5}$	6	
	$\hat{\Delta}_k$	-	-	0	0	-	-	1	1		
	Δ_k	-	-	-1/5	<u>$-\frac{3}{5}$</u>	-	-				
IV											
	x_1	1	0	-3/8	0	-1/8	0			1	
	x_2	0	1	-1/4	0	1/4	0			3	
	x_4	0	0	-9/8	1	5/8	0			5	
	x_6	0	0	5/8	0	-1/8	1			5	8
	Δ_k	-	-	<u>-7/8</u>	-	3/8	-				
V											
	x_1	1	0	0	0	-1/5	3/5			4	
	x_2	0	1	0	0	1/5	2/5			5	
	x_4	0	0	0	1	2/5	9/5			14	
	x_3	0	0	1	0	-1/5	8/5			8	

	Δ_k	-	-	-	-	1/5	7/5				
--	------------	---	---	---	---	-----	-----	--	--	--	--

Запишем последовательности опорных планов.

Для первых трех шагов приращения $\hat{\Delta}_k$ вычисляются только по искусственным переменным, которые входят в искусственную целевую функцию $\hat{f}(\bar{x}) = x_7 + x_8$ с коэффициентом $c_i = 1$.

На третьем шаге искусственные переменные исключены, так как все $\hat{\Delta}_k$ положительны.

Эквивалентная задача решена. Далее приращения Δ_k вычисляются на основе исходной целевой функции

$$f(\bar{x}) = -x_1 - 2 \cdot x_2.$$

$$\bar{x}^1 = 0, 0, 0, 0, 7, 9, 5, 5, \quad f(\bar{x}^1) = 0, \quad \hat{f}(\bar{x}^1) = 10.$$

$$\bar{x}^2 = \left[0, \frac{5}{3}, 0, 0, 2, \frac{22}{3}, \frac{10}{3}, 0 \right], \quad f(\bar{x}^2) = -\frac{10}{3}, \quad \hat{f}(\bar{x}^2) = \frac{10}{3}.$$

$$\bar{x}^3 = 2, 1, 0, 0, 8, 6, \quad f(\bar{x}^3) = -4, \quad \hat{f}(\bar{x}^3) = 0.$$

$$\bar{x}^4 = [3, 0, 5, 0, 5], \quad f(\bar{x}^4) = -7.$$

$$\bar{x}^5 = [4, 5, 8, 14, 0, 0], \quad f(\bar{x}^5) = -14.$$

Итак, симплекс-метод с введением искусственных переменных включает два этапа.

I этап.

Формирование и решение вспомогательной задачи ЛП с введением искусственных переменных. Искусственные переменные в начальном опорном плане являются базисными. Искусственная целевая функция включает только искусственные переменные. При получении смежных опорных планов искусственные переменные из базисных переводим в свободные. В результате получен оптимальный опорный план для вспомогательной задачи $\hat{f}(\bar{x}) = 0$.

II этап.

Оптимальный опорный план вспомогательной задачи ЛП является начальным опорным планом основной задачи ЛП. Задача решается для исходной целевой функции $f(\bar{x})$ обычным симплекс – методом.

Замечания.

1. Введение искусственных переменных требуется в двух случаях:
 - ряд базисных переменных x_i в канонической форме отрицательны;
 - если трудно свести к канонической форме, то просто в любое уравнение-ограничение добавляем искусственную переменную.

2. Встречающиеся в практике автоматического управления задачи линейного программирования содержат от 500 до 1500 ограничений и более 1000 переменных. Ясно, что задачи такой размерности можно решать лишь с помощью ЭВМ и специального программного обеспечения. Сложность алгоритма заключается в том, что

- ППП требуют канонического вида;
- ППП для задач такой размерности требует использования больших ЭВМ (и параллельных вычислений), т.к. симплекс – метод хранит всю таблицу.

3. Вычислительную эффективность симплекс – метода можно оценить следующими показателями:

- число шагов (смежных опорных планов);
- затраты машинного времени.

Существуют такие теоретические оценки для стандартной задачи линейного программирования с m – ограничениями и n – переменными:

Среднее число шагов $\approx 2 \cdot m$ и лежит в диапазоне $\left[m \div 2 \cdot m + n \right]$.

Расчетное время пропорционально величине m^3 .

4.5. Двойственность задач линейного программирования

Теория двойственности представляет собой важное теоретическое и практическое направления линейного программирования. Основной идеей теории двойственности является постулат: для любой задачи линейного программирования существует некоторая другая задача линейного программирования, решение которой тесно связано с решением исходной.

Таким образом, двойственность состоит в существовании пары задач (4.11) и (4.12).

Прямая задача

$$\left. \begin{aligned} f(\bar{x}) = \sum_{j=1}^n c_j \cdot x_j \rightarrow \max \\ \sum_{j=1}^n a_{ij} \cdot x_j \leq b_i, i = \overline{1, m} \end{aligned} \right\} . \quad (4.11)$$

Двойственная задача

$$\left. \begin{aligned} z(\bar{y}) = \sum_{i=1}^m b_i \cdot y_i \rightarrow \min \\ \sum_{i=1}^m a_{ji} \cdot y_i \geq c_j, j = \overline{1, n} \end{aligned} \right\} . \quad (4.12)$$

В задачах (4.11) и (4.12) a_{ij} , b_i , c_j - заданные числа; x_j , y_i - переменные.

Каждая из задач I и II называется двойственной по отношению к другой.

Общей для двух задач является матрица ограничений a_{ij} .

Важность двойственности задачи линейного программирования состоит в возможности при практической реализации симплекс – метода замены целевой функции $f(\bar{x}) \rightarrow \max$ на $f(\bar{x}) \rightarrow \min$. Эта замена основана на двух теоремах двойственности.

Теорема 1. Если существует оптимальный план одной задачи, то и существует оптимальный план другой, при этом справедливо неравенство

$$\sum_{j=1}^n c_j \cdot x_j \leq \sum_{i=1}^m b_i \cdot y_i . \quad (4.13)$$

Неравенство переходит в равенство если \bar{x} и \bar{y} оптимальны.

Теорема 2. Если \bar{x}^* , \bar{y}^* - оптимальные планы, то компоненты планов связаны соотношениями

$$\begin{aligned} \left(b_i - \sum_{j=1}^n a_{ij} \cdot x_j^* \right) \cdot y_i^* &= 0 \\ \left(c_j - \sum_{i=1}^m a_{ij} \cdot y_i^* \right) \cdot x_j^* &= 0 \end{aligned} \quad (4.14)$$

Сравним двойственные задачи.

1. Матрица ограничений $A = a_{ij}$

В исходной задаче

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

В двойственной

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

То есть матрица A^1 является транспонированной к матрице A .

2. Число ограничений и переменных.

В исходной задаче
 n – переменных,
 m – ограничений.

В двойственной
 m – переменных,
 n – ограничений.

3. Правые части ограничений – это коэффициенты целевой функции двойственной задачи.

4. Знаки ограничений и цель задачи меняются на противоположные.

В исходной задаче

а) знаки ограничений $<$,
б) целевая функция $\rightarrow \max$.

В двойственной

а) знаки ограничений $>$,
б) целевая функция $\rightarrow \min$.

Для практических целей рассмотренные правила позволяют составить схему соответствия двойственных задач, представленных в таблице 4.7.

Таблица 4.7

Соответствие двойственных задач ЛП

Исходная задача	Двойственная задача
Целевая функция $\rightarrow \max$	Целевая функция $\rightarrow \min$
Константы в правых частях ограничений	Коэффициенты целевой функции
Коэффициенты целевой функции	Константы в правых частях ограничений
j-й столбец коэффициентов в ограничениях	j-я строка коэффициентов в ограничениях
j-я строка коэффициентов в ограничениях	j-й столбец коэффициентов в ограничениях
j-я неотрицательная переменная	j-е неравенство вида \geq
j-я переменная, не имеющая ограничений в знаке	j-е соотношение в виде $=$
i-е неравенство вида \leq	i-я неотрицательная переменная
i-е соотношение в виде $=$	i-я переменная не имеющая ограничений в знаке

Практическая ценность проблемы двойственности состоит в следующем.

1. Переход от исходной задачи к двойственной полезен, если число ограничений исходной задачи велико и много больше числа переменных. При этом объем вычислений в основном определяется числом ограничений.

2. Если по какой – то причине решение исходной задачи затруднительно, то пытаются решить двойственную задачу, а затем на основе приведенных выше Теорем 1 и 2, находится решение прямой задачи.

В таблице 4.8 Приведено решение двойственной задачи об использовании сырья. Исходные данные для прямой задачи приведены в таблице 4.2. Отметим, что графическое решение, приведенное выше на рисунке 4.1 привело к оптимальному плану

$$\bar{x} = 20, 30, \quad f(\bar{x}) = 120.$$

Использование решения двойственной задачи должно привести к такому оптимальному решению.

Запишем прямую и двойственную задачу.

Для решения двойственной задачи введен искусственный базис, переменные y_6, y_7 которые исключаются из опорного плана на третьем шаге.

Прямая задача

$$f(\bar{x}) = 3 \cdot x_1 + 2 \cdot x_2 \rightarrow \max$$

$$5 \cdot x_1 + 2 \cdot x_2 \leq 160$$

$$3 \cdot x_1 + 4 \cdot x_2 \leq 180$$

$$7 \cdot x_1 \leq 196$$

Двойственная задача

$$f(\bar{y}) = 160 \cdot y_1 + 180 \cdot y_2 + 196 \cdot y_3 \rightarrow \min$$

$$5 \cdot y_1 + 3 \cdot y_2 + 7 \cdot y_3 - y_4 \quad y_6 = 3$$

$$2 \cdot y_1 + 4 \cdot y_2 \quad - y_5 + y_7 = 2$$

$$F_{иск} = y_6 + y_7$$

$$n = 2, m = 3$$

Таблица 4.8

Симплекс таблица двойственной задачи

	m_i	Базис	y_1	y_2	y_3	y_4	y_5	y_6	y_7	b	θ
I		y_6	5	3	7	-1	0	1	0	3	3/7
	0	y_7	2	4	0	0	-1	0	1	2	-
		$\hat{\Delta}$	-7	-7	-7	1	1	-	-		
II		y_3	5/7	3/7	1	-1	0	1	0	3/7	1
	0	y_7	2	4	0	0	-1	0	1	2	1/2
		$\hat{\Delta}$	-2	-4	-	0	1	1	-		
III		y_3	1/2	0	1	-1	0			3/14	3/7
	-1	y_2	1/2	1	0	0	-1			11/2	1
		$\hat{\Delta}$	0	-	-	0	0				
		Δ	-28	-	-	0	0				
IV		y_1	1	0	2					3/7	
		y_2	0	1	-1					2/7	
		Δ	-	-	56						

Таким образом, получен следующий оптимальный план двойственной задачи:

$$\begin{cases} y^* = \left[\frac{3}{7}, \frac{2}{7}, 0 \right], \\ F(y^*) = 120. \end{cases}$$

Для получения оптимального решения прямой задачи воспользуемся соотношением из Теоремы 2

$$\left(c_j - \sum_{i=1}^m a_{ij} \cdot y_i^* \right) \cdot x_j^* = 0.$$

Подставив оптимальные значения вектора \bar{y}^* , получим оптимальные значения вектора \bar{x}^*

$$\begin{cases} 5 \cdot x_1^* + 2 \cdot x_2^* - 160 \cdot \frac{3}{7} = 0, \\ 3 \cdot x_1^* + 4 \cdot x_2^* - 180 \cdot \frac{2}{7} = 0 \end{cases},$$

$$x_1^* = 20, \quad x_2^* = 30.$$

Получен тот же оптимальный план что и при решении прямой задачи.

4.5.1. Экономическая трактовка двойственности

Если прямую ЗЛП (4.11) рассматривать, как задачу об использовании сырья (ресурсов), то параметры задачи имеют следующий экономический смысл.

В прямой задаче:

x_j - количество единиц j -го продукта;

c_j - стоимость единицы j -го продукта;

b_i - ресурс (запас) i -го сырья.

В обратной задаче:

y_i - стоимость единицы i -го сырья;

Тогда стоимость всех ресурсов

$$\sum_{i=1}^m b_i \cdot y_i \rightarrow \min.$$

А стоимость всех затраченных ресурсов, идущих на выпуск единицы j -ой продукции не меньше окончательной стоимости этого продукта

$$\sum_{i=1}^m a_{ji} \cdot y_i \geq c_j.$$

Если коэффициенты ограничений b_i понимается как ресурсы, то с точки зрения предприятия эти ресурсы имеют определенную ценность. Здесь понимается, что те ресурсы, которые предприятие не может использовать полностью, имеют для него низкую ценность (то есть предприятие не согласно нести расходы на увеличение запасов таких ресурсов). Например, дорогостоящее оборудование, не используемое на предприятии, имеет для него ценность равную нулю. А максимальную ценность имеют те ресурсы, которые используются полностью. На них предприятие согласно нести расходы. Поэтому можно считать, что любой вид ресурса обладает некоторой «теневой ценой», определяющей ценность данного ресурса для получения предприятием дохода, равного разности (прибыль – затраты). А под оптимальным планом понимается отношение $\frac{\text{прибыль}}{\text{затраты}}$.

Определение оптимальных «теневого цен» – основная цель двойственной задачи.

Оптимальное двойственное решение можно интерпретировать, как набор оценок ресурсов, то есть теневые цены. При этом, цель прямой задачи найти сколько и какой продукции x_j необходимо произвести при заданных стоимостях продукции c_j и объемах имеющихся ресурсов b_i .

Подобная интерпретация двойственности производится на основе модели американского экономиста В.Леонтьева «Затраты - выпуск». Эта модель описывает экономическую систему, состоящую из ряда взаимосвязанных отраслей, выпускающих один вид продукции и использующей один вид производства. Производство каждой из отраслей должно удовлетворять спрос других отраслей и внешних источников.

4.7. Динамическое программирование

Во многих задачах ЛП на допустимое множество ограничений накладывается дополнительное условие некоторых переменных x_j . Целочисленными могут быть все переменные (полностью целочисленная задача), или часть переменных. Например, число работников-контроллеров в задаче ОТК не может быть не целым.

На практике задачи целочисленного программирования (ЦП) решаются с помощью симплекс-метода, как для обычных ЗЛП. Если в результате решения все искомые переменные будут целыми, то считается, что задача решена. В противном случае, нужные переменные округляются до ближайшего целого так, чтобы не нарушались ограничения ЗЛП. Обычно этот способ решения дает хорошее целочисленное решение, близкое к оптимуму целевой функции, но только в случае, если переменные задачи велики. Если же значения переменных малы (например, булевы, 0 и 1), то подобное решение даст неприемлемые результаты.

В общем случае, для задач ЦП наиболее известным является метод ветвей и границ, а также метод Гомори. Оба эти метода предназначены как для решения полностью, так и для частичных задач ЦП.

Метод Гомори

Запишем ЗЛП в стандартном (каноническом) виде.

$$\begin{aligned} f(\bar{x}) &= \sum_{j=1}^n c_j \cdot x_j \rightarrow \min \\ \sum_{j=1}^n a_{ij} \cdot x_j &= b_i, \quad i = \overline{1, m}, \\ b_i &\geq 0, \quad i = \overline{1, m}, \\ x_j &\geq 0, \quad j = \overline{1, n}, \\ x_j &\in Z, \end{aligned} \quad (4.15)$$

Где Z – множество целых чисел.

Если полностью целочисленная задача ЦП содержит только 2 переменные, то ее можно решить графически. Для этого сначала строится допустимая область из ограничений, как при обычной ЗЛП. Затем, счи-

тая, что допустимая область целочисленной ЗЛП состоит из точек целочисленной координатной сетки, перемещаем графически ЦФ из оптимальной вершины в ближайшую целочисленную координату. Это и считается оптимумом целочисленной задачи.

На рисунке 4.4 приведен пример графического решения. Решение обычной ЗЛП позволило получить оптимальную вершину D {5.5, 1}. Перемещая ЦФ до ближайшее целой координаты x_1 в допустимой области получаем решение задачи ЦП, точку C {5, 1}.

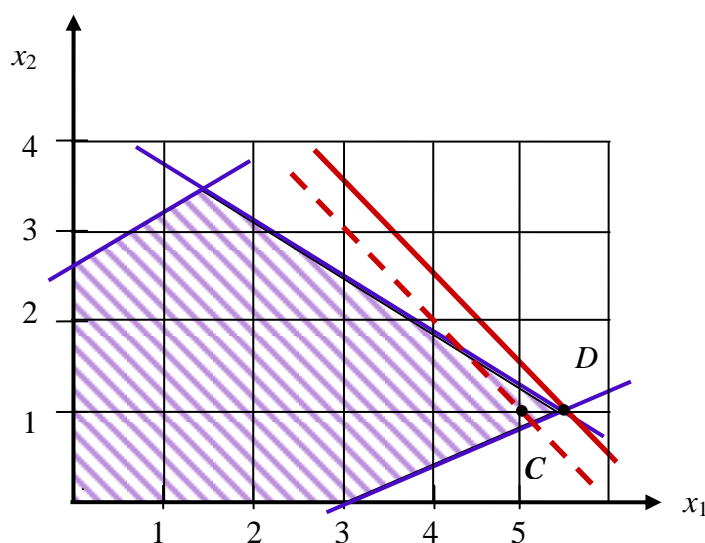


Рисунок 4.4. Графический способ решения задачи ЦП

В случае, если координат задачи ЦП больше двух, то графический способ неприемлем и рекомендуется использовать метод Гомори.

Его основная идея состоит в последовательном отсеке от допустимого множества решений нецелочисленной ЗЛП оптимальных нецелочисленных решений, содержащих точки с целыми координатами. Эти отсеки производятся включением в задачу дополнительных ограничений на переменные x_j .

Описание алгоритма Гомори.

1. С помощью симплекс-метода находим решение x^* ЗЛП без требования целочисленности переменных. Если все переменные x_j - целые, то задача решена. Иначе – шаг 2.

2. Среди нецелых элементов $b_i, i = \overline{1, m}$ выбираем любой b_k , например, с максимальной дробной частью.

По k -ой строке симплекс-таблицы составляется дополнительное ограничение вида

$$-\sum_{j \in \text{своб.}}^n \{a_{kj}\} \cdot x_j \leq -\{b_k\}.$$

В последней формуле m переменных $j \in \text{своб.}$ означает, что в системе m переменных базисные, остальные $j = m+1 \dots n$ – свободные.

Также в этой формуле принято обозначение целой $[a]$ и дробной части $\{a\}$ действительного числа a

$$a = [a] + \{a\}.$$

С помощью вспомогательной $x_{n+1} \geq 0$ переменной это ограничение представляется в виде равенства

$$x_{n+1} - \sum_{j \in \text{своб.}}^n \{a_{kj}\} \cdot x_j \leq -\{b_k\}$$

М вводится в симплекс-таблицу дополнительной $(m+1)$ строкой с базисным элементом x_{n+1}

$$x_{n+1} | a_{n+1,m+1} \dots a_{n+1,n} | b_{n+1}, \quad (4.16)$$

где

$$a_{n+1,j} = \{a_{k,j}\}, \quad b_{n+1} = \{b_k\}, \quad j = \overline{m+1, n}.$$

Так как $b_{n+1} < 0$, то такая симплекс-таблица перестает соответствовать допустимому базисному решению (опорному плану).

3. Для перехода к опорному плану производятся следующие операции:

а) строка с отрицательным свободным членом b_k считается разрешающей (на первом шаге $k = n+1$).

б) если все коэффициенты $a_{k,j} > 0$, то задача не имеет решения. В противном случае номер l разрешающего столбца находится из условия

$$l = \left\{ \min_{l: a_{kl} < 0} \frac{\Delta_l}{|a_{kl}|} \right\}.$$

в) совершаются преобразования симплекс-таблицы с опорным элементом $a_{k,l}$. Если в новой симплекс таблице по-прежнему есть хотя бы один отрицательный свободный член, следовательно, описанная процедура повторяется, начиная с пункта а) необходимое число раз.

Если все $b_i > 0$, следовательно опорный план найден.

Надо отметить, что выбор опорного элемента $a_{k,l}$ гарантирует положительные значения приращений Δ_j новой симплекс-таблицы. Поэтому найденный опорный план и является оптимальным.

4. Если найденное в пункте 3. Решение ЗЛП удовлетворяет условию целочисленности, то вычисления завершаются. Если нет, то переходим к пункту 2.

Пример. Решить методом Гомори следующую задачу целочисленного программирования.

$$\begin{aligned} f(\bar{x}) &= x_1 - 20 \cdot x_2 \rightarrow \min \\ -x_1 + 10 \cdot x_2 &\leq 40 \\ 4 \cdot x_1 + 2 \cdot x_2 &\leq 29 \\ x_1, x_2 &\geq 0, \\ x_1, x_2 &\in Z, \\ n &= 2, m = 2. \end{aligned}$$

Введем переменные x_3, x_4 и запишем ЗЛП в каноническом виде. Так как все коэффициенты ограничений-равенств целочисленные, то целочисленность переменных x_1, x_2 влечет целочисленность переменных x_3, x_4 . Поэтому задача полностью целочисленная и может быть использован метод Гомори, таблица 4.9.

Таблица 4.9

Симплекс таблица целочисленной задачи

	m_i	Базис	x_1	x_2	x_3	x_4	x_5	b	θ
0		x_3	-1	<u>10</u>	1	0		40	4
	-2/10	x_4	4	2	0	1		29	14,5
		$\hat{\Delta}$	1	<u>-20</u>	-	-			
I	1/42	x_2	-1/10	1	1/10	0		4	
		x_4	<u>42/10</u>	0	-2/10	1		21	
		$\hat{\Delta}$	<u>-1</u>	-	2	-			
II		x_2	0	1	4/42	1/42		9/2	

	-1	x_1	1	0	-2/42	10/42		5	
		Δ	-	-	82/42	10/42			
III		x_2	0	1	4/42	1/42	0	9/2	
	-1	x_1	1	0	-2/42	10/42	0	5	
		x_5	0	0	-4/42	<u>-1/42</u>	1	<u>-1/2</u>	
		Δ	-	-	82/42	10/42	-		
IV		x_2	0	1	0	0	1	4	
		x_1	1	0	-1	0	10	0	
		x_4	0	0	4	1	-42	21	
		Δ	-	-		-			

Запишем последовательности опорных планов.

$$\bar{x}^{-0} = 0, 0, 40, 20, \quad f(\bar{x}^{-0}) = 0.$$

$$\bar{x}^{-1} = 0, 4, 0, 21, \quad f(\bar{x}^{-1}) = -80.$$

$$\bar{x}^{-2} = \left[5, \frac{9}{2}, 0, 0 \right], \quad f(\bar{x}^{-2}) = -85.$$

На втором шаге оптимальное решение найдено, но оно не удовлетворяет требованию целочисленности переменных. Поэтому в соответствии с (11) дополним симплекс-таблицу строкой и переменной x_5 . Выполним преобразования пункта 3 алгоритма Гомори.

а) строка с отрицательным свободным членом $b_k (-1/2)$ считается разрешающей.

б) Номер l разрешающего столбца находится из условия

$$\min_{l: a_{kl} < 0} \left\{ \frac{82 \cdot 41}{42 \cdot 4}; \frac{10 \cdot 42}{42 \cdot 1} \right\} = 10.$$

$$l=4.$$

в) преобразование симплекс-таблицы с опорным элементом -1/42 приводит к тому, что все $b_i > 0$, следовательно, опорный план найден.

Полученный опорный план является опорным планом целочисленной задачи.

$$\bar{x}^{-4} = 0, 4, \quad f(\bar{x}^{-4}) = -80.$$

4.7. Динамическое программирование

В задачах линейного и нелинейного программирования оптимизируемый процесс считался статическим, т.е. независимым от времени. Оптимальное решение в таких задачах находится за один этап планирования. Такие задачи называются одноэтапными.

В задачах динамического программирования оптимизационный процесс зависит от времени (от нескольких этапов времени). Поэтому оптимальное решение находится последовательно для каждого этапа, обеспечивая при этом оптимальные решения всего процесса в целом. Задачи динамического программирования называют многоэтапными.

Управлением в динамическом программировании называется совокупность решений, принимаемых на каждом этапе с целью влияния на ход всего процесса. В прикладном плане задачи динамического программирования на 90% - это задачи планирования: объемов производства, поставок сырья, величины финансирования и т.д. Например, совокупность решений, принимаемых в начале календарного года по обеспечению сырьем, замене оборудования, размерам инвестиций - это все этапное планирование, которое должно обеспечить генеральную задачу - максимальный выпуск продукции в конце года. И простыми мерами: использование оборудования на полную мощность, максимально возможные инвестиции - генеральную задачу не решить, т.к. начинают влиять другие факторы, например, износ оборудования. В таком случае необходимо поэтапное планирование, т.е. замена оборудования при его износе на определенных этапах. Таким образом, выпуск продукции - многоэтапная задача, каждый из этапов которой осуществляет влияние на конкретную цель.

На рисунке 4.5, а) приведена область возможных состояний системы для одномерного случая. Рисунки 4.5, б), в) - для двумерного и трехмерного случая соответственно.

Область возможных состояний системы - это совокупность состояний, в которые может переходить система. Например, в случае а) - это ось ОХ или ее участок, а управление - это закон движения точки из состояния $x^0 \rightarrow x^k$.

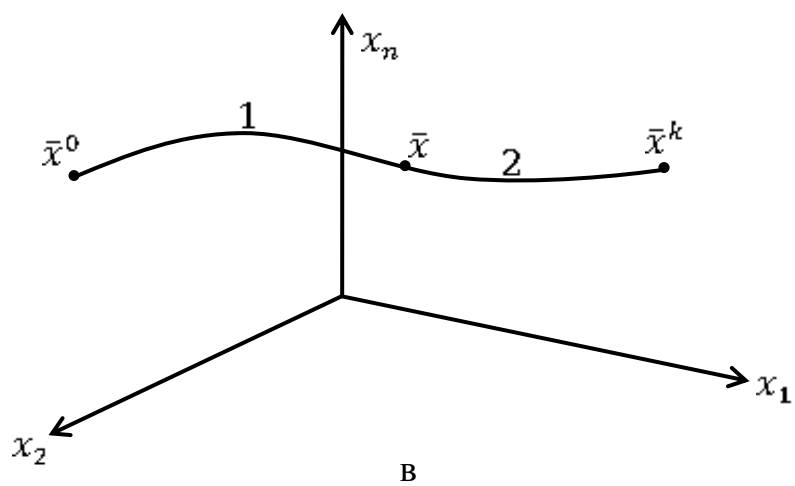
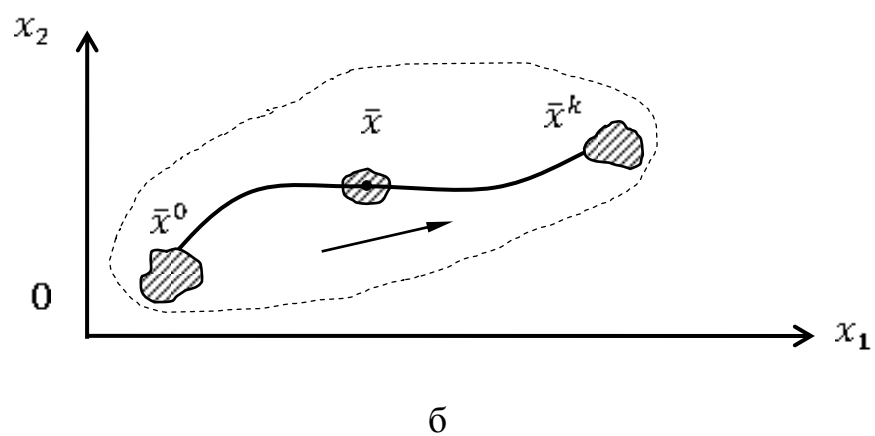


Рисунок 4.5. Схема решения задач динамического программирования

Постановка задачи

Пусть состояние некоторого объекта описывается вектором фазового пространства $\bar{x} = (x_1, x_2, \dots, x_n) \in R_n$. Будем считать, что процесс является N -шаговым, т.е. его эволюция происходит за N шагов по схеме

$$\begin{array}{ccccccc} \bar{x}^0 & \rightarrow & \dots & \rightarrow & \underbrace{\bar{x}^{k-1} \rightarrow \bar{x}^k}_{k\text{-й шаг}} & \rightarrow & \dots \rightarrow \bar{x}^N \\ \text{Начальное} & & & & & & \text{Конечное} \\ \text{состояние} & & & & & & \text{состояние} \end{array}$$

Переход между состояниями на k -м шаге осуществляется в соответствии с уравнением состояния

$$\bar{x}^k = f^k(\bar{x}^{k-1}, \bar{u}^k), \quad (1)$$

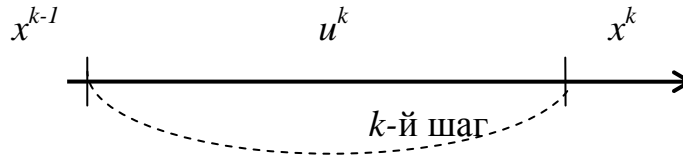
где $\bar{u}^k \in R^m$ - m -мерный вектор управления на k -м шаге;

$f^k(\bar{x}, \bar{u})$ - n -мерная векторная функция аргументов \bar{x} , \bar{u} .

Распишем компоненты вектора

$$x_i^k = f_i(x_1^{k-1}, x_2^{k-1}, \dots, x_n^{k-1}, u_1^k, u_2^k, \dots, u_m^k).$$

Таким образом, предполагается, что в результате k -го шага система переходит в состояние \bar{x}^k , которое определяется только начальным состоянием этого шага \bar{x}^{k-1} и выбранным на нем вектором управления \bar{u}^k и не зависит от предыстории процесса $\bar{x}^0, \bar{x}^{(1)}, \dots, \bar{x}^{(k-2)}, \bar{u}^{(1)}, \dots, \bar{u}^{(k-1)}$.



Показателем эффективности k -го шага является заданная числовая характеристика (целевая функция) этого шага

$$Z_k = Z_k(\bar{x}^{k-1}, \bar{u}^k) \quad k = 1, 2, \dots, N.$$

А эффективность всего процесса складывается из этапных целевых функций

$$Z(\hat{x}, \hat{u}) = \sum_{k=1}^N Z_k(\bar{x}^{k-1}, \bar{u}^k).$$

На фазовые траектории \hat{x} и управления \hat{u} наложены ограничения:

$$\begin{aligned}\bar{x}^k &\in X_k, & k = \overline{1, N-1} \\ \bar{u}^k &\in U_k(\bar{x}^{k-1}), & k = \overline{1, N}\end{aligned}.$$

Множества X_k и U_k заданы в пространствах R^n, R^m .

Кроме того, заданы начальные и конечные условия

$$\begin{aligned}\bar{x}^0 &\in X_0 \\ \bar{x}^N &\in X_N.\end{aligned}$$

Часто множества X_0 и X_N содержат по одной точке, определяющей начало и конец фазовой траектории.

Общую задачу многошаговой оптимизации можно записать следующим образом

$$\begin{aligned}Z(\hat{x}, \hat{u}) &= \sum_{k=1}^N Z_k(\bar{x}^{k-1}, \bar{u}^k) \rightarrow \text{extr} \\ \bar{x}^k &= f^k(\bar{x}^{k-1}, \bar{u}^k), \quad k = \overline{1, N} \\ \bar{x}^k &\in X_k, & k = \overline{1, N-1} \\ \bar{u}^k &\in U_k(\bar{x}^{k-1}), & k = \overline{1, N} \\ \bar{x}^0 &\in X_0, \\ \bar{x}^N &= X_N.\end{aligned} \quad (2)$$

Требуется среди всех управлений $\hat{u} \in U$ выбрать такое $\bar{u}^* = (\bar{u}^{*1}, \dots, \bar{u}^{*N})$, для которого целевая функция (2) принимает экстремальное значение (минимальное или максимальное).

Пример 1. Сформулировать следующую оптимизационную задачу в виде многошаговой задачи оптимизации.

С помощью N-ступенчатой ракеты с заданной стартовой массой М в космос выводится межпланетная станция массой m. За время работы каждая ступень ракета получает добавочную скорость

$$\Delta V = F(y, z),$$

где y - масса, разгоняемая этой ступенью;

z - масса самой ступени.

Найти такое распределение массы ракеты M между ее ступенями, при котором конечная скорость ракеты максимальна.

Пусть $\bar{u}^k, k = \overline{1, N}$ - масса k -й ступени, начиная от межпланетной станции, т.е. на старте работает ступень массой u^N , в конце - u^1 .

Обозначим $x^k, k = \overline{0, N}$ - сумму массы межпланетной станции и массы k ступеней, примыкающих к ней.

Тогда

$$\begin{aligned} x^k &= x^{k-1} + u^k, & k &= \overline{1, N}, \\ x^0 &= m, & x^N &= M + m, \\ x^k &\in [m; M + m], & k &= \overline{1, N-1}, \\ u^k &\in [0, M], & k &= \overline{1, N}, \\ z(\hat{x}, \hat{y}) &= \sum_{k=1}^N F(x^{k-1}, u^k) \rightarrow \max. \end{aligned}$$

Теория многошаговых оптимизационных задач разработана в 50-х годах американским математиком Р. Беллманом. Метод решения такой задачи носит название метода динамического программирования и основан он на принципе оптимальности Беллмана.

Оптимальная траектория обладает тем свойством, что каждая ее завершающая часть, начинающаяся с k -го шага, $k = \overline{1, N-1}$ является оптимальной для остающихся шагов процесса. Другими словами, на каждом этапе решения, учитывая развитие всего процесса, оптимизируют только один шаг. И таким образом, при принятии решения учитывают будущее. Однако в каждом процессе есть последний k -й шаг, который не зависит от будущего. Поэтому на этом шаге управление позволяет получить максимальный эффект. Спланировав k -й шаг, к нему присоединяют $(k-1)$, затем $(k-2)$ и т.д. Процесс динамического программирования как бы разворачивается от конца к началу.

Чтобы спланировать k -й шаг, надо знать состояние системы на $(k-1)$ шаге. Если состояние на $(k-1)$ шаге неизвестно, то делают различные предположения о возможных состояниях системы на этом шаге. Для каждого предположения выбирают оптимальное управление на последнем k -м шаге. Такое оптимальное управление называют условно оптимальным.

Рассмотрим k -шаговый процесс, рисунок 4.6. Сделаем ряд предположений о возможных состояниях процесса на $(k-1)$ шаге. Обо-

значим эти состояния $S_{k-1,1}; S_{k-1,2}; \dots S_{k-1,r}$. На последнем найдем для каждого из этих состояний условно оптимальное управление $u_{k,1}^*(x_{k-1,1}), u_{k,2}^*(x_{k-1,2}), \dots, u_{k,r}^*(x_{k-1,r})$. Таким образом, k -й шаг спланирован. Действительно, какое бы состояние система ни принимала на $(k-1)$ шаге, уже известно как следует поступить на k -м.

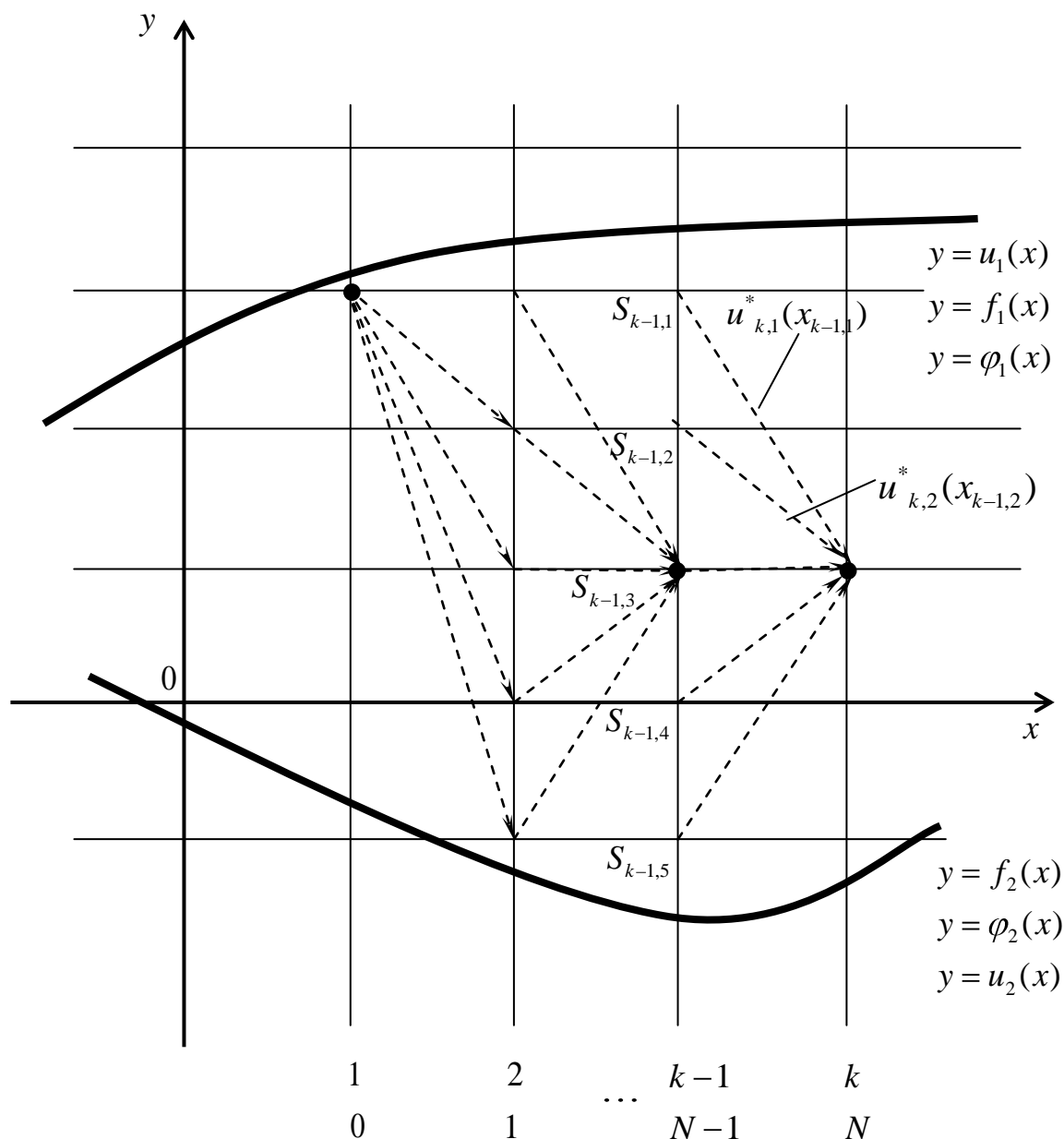


Рисунок 4.6. Схема метода динамического программирования Р.Беллмана

Аналогично поступаем на $(k-1)$ шаге, только условно оптимальные управления необходимо выбирать, учитывая уже выбранные условно

оптимальные на k -м шаге. В итог, выполнив все переходы, приходим к \bar{x}^0 .

Для первого шага предположений не делаем, т.к. \bar{x}^0 - задано, и находим оптимальные управления, учитывая все уже найденные. Проходя от \bar{x}^0 к \bar{x}^k получаем искомое оптимальное управление для всего процесса. Используя принцип оптимальности, найдем необходимые условия, которым должно удовлетворять оптимально управляемая последовательность $\bar{u}_1, \dots, \bar{u}_{N-1}$.

Рассмотрим конечный участок траектории, интервал $[k, N]$. И допустим, что для интервалов $[0, 1, \dots, k-1]$ найдены оптимальные управления и оптимальная траектория, включая \bar{x}_k . Остается найти управление $\bar{u}_k, \dots, \bar{u}_N$ на конечном участке. Из принципа оптимальности следует, что $\{\bar{u}_k, \dots, \bar{u}_N\}$ определяется только состоянием \bar{x}_{k-1} и целью поиска экстремума, которая для $[k, N]$ имеет вид

$$Z_k(\hat{x}, \hat{y}) = \sum_{i=k}^N Z_i(\bar{x}^{i-1}, \bar{u}^i),$$

при $\hat{x} = x^{*k}$, $\hat{u} = u^{*k}$ целевая функция Z_k имеет оптимальное значение (максимум или минимум). Обозначим его

$$B_k(\bar{x}^{k-1}) = Z_k(\bar{x}^{*k}, \bar{u}^{*k}).$$

Аналогично

$$B_N(\bar{x}^{N-1}) = \underset{\bar{u}^N \in U_N(x^{N-1})}{extr} Z_N(\bar{x}^{N-1}, \bar{u}^N). \quad (3)$$

Из принципа оптимальности можно записать следующие рекуррентные выражения:

$$B_k(\bar{x}^{k-1}) = \underset{u^k \in U_k(x^{k-1})}{extr} \{B_{k+1}(\bar{x}^k) + Z_k(\bar{x}^{k-1}, \bar{u}^k)\}, \quad k = 1, N-1 \quad (4)$$

$$(B_k(\bar{x}^{k-1}) = \underset{i=k+1}{extr} \{ \sum_{i=k+1}^N Z_i(\bar{x}^{i-1}, \bar{u}^i) + Z_k(\bar{x}^{k-1}, \bar{u}^k) \})$$

Соотношения (3) - (4) позволяют последовательно найти функции Беллмана.

$B_N(x^{N-1}), B_{N-1}(x^{N-2}), \dots, B_1(x^0)$ - называются уравнениями Беллмана.

Находя $B_k(x^{k-1}), k = N, N-1, \dots, 1$ мы одновременно находим управления $u_*^k(x^{k-1})$, которые называются условно-оптимальными управлениями, а процесс их нахождения - условной оптимизацией.

Управление $\bar{u}_*^k(\bar{x}^{k-1})$, найденное из уравнения (4) удовлетворяет принципу оптимальности: т.е. в зависимости от начального состояния x^{k-1} управления u_*^k учитывает оптимизацию не только k -го шага, но и следующих за ним $(N-k)$ шагов.

Итак, в результате условной оптимизации находят $B_k(x^{k-1})$ и $u_*^k(x^{k-1})$, $k = N, N-1, \dots, 1$. Теперь можно осуществить безусловную оптимизацию задачи (2), т.е. найти искомые оптимальные управления $\hat{u}^* = (u_*^1, u_*^2, \dots, u_*^N)$ и оптимальную фазовую траекторию $\hat{x}^* = (x_*^0, \dots, x_*^N)$.

Т.к. значение $B_1(x^0)$ равно оптимальному значению целевой функции всех N шагов, то

$$B_1(x_*^0) = \underset{x^0 \in x_0}{extr} B_1(x^0), \quad (5)$$

если x_0 – первая точка траектории, то $x_*^0 = x^0$

Далее, используя найденные условно оптимальные управления и уравнения состояния $\bar{x}^k = f^k(\bar{x}^{k-1}, \bar{u}^k)$, последовательно находим:

$$\begin{aligned} & u_*^1, x_*^1, u_*^2, x_*^2, \dots, u_*^N, x_*^N \\ & u_*^1 = u_*^1(x_*^0), \quad x_*^1 = f^1(x_*^0, u_*^1), \\ & u_*^2 = u_*^2(x_*^1), \quad x_*^2 = f^2(x_*^1, u_*^2), \\ & \dots \quad \dots \\ & u_*^N = u_*^N(x_*^{N-1}), \quad x_*^N = f^N(x_*^{N-1}, u_*^N). \end{aligned} \quad (6)$$

Алгоритм метода динамического программирования

Этап I. Условная оптимизация.

Шаг 1. Находим условно оптимальные управления $u_*^N(x^{N-1})$ и функцию Беллмана $B_N(x^{N-1})$ в соответствии с соотношением (3).

Шаг 2. Находим $u_*^{N-1}(x^{N-2})$ и $B_{N-1}(x^{N-2})$.

...

Шаг N . Находим $u_*^1(x^0)$ и $B_1(x^0)$.

Этап II. Безусловная оптимизация.

Шаг 0. Находим x_*^0 в соответствии с (5).

Шаг 1. Находим u_*^1 и x_*^1 по формулам (6).

...

Шаг N . Находим u_*^N и x_*^N .

Итак, метод динамического программирования дает эффективный алгоритм оптимизации решений. Алгоритм основан на решении рекуррентных уравнений Беллмана. При этом однократное решение сложной задачи заменяется многократным решением более простой. Все же процедура динамического программирования слишком громоздка, т.к. на каждом шаге решения необходимо запоминать $B_k(x_{k-1})$, $u_*^k(x_{k-1})$.

И хотя метод динамического программирования существенно упрощает исходные задачи, однако непосредственное его применение, как правило, сопряжено с громоздкими вычислениями. Для преодоления этих трудностей существуют приближённые методы динамического программирования.

Пример 2. Решить задачу из Примера 1 со следующими условиями:

$$N = 3, M = 63, m = 1, F(y, z) = \frac{z}{y}.$$

$$Z(\hat{x}, \hat{u}) = \sum_{k=1}^3 \frac{u^k}{x^{k-1}} \rightarrow \max.$$

$$x^k = x^{k-1} + u^k, \quad k = 1, 2, 3.$$

$$x^k \in [1, 64], \quad k = 1, 2.$$

$$u^k \in [0, 63], \quad k = 1, 2, 3.$$

$$x^0 = 1, \quad x^3 = 64.$$

Этап I. Условная оптимизация.

Найдем уравнения Беллмана по формуле (3)

$$\text{Шаг 1. } B_3(x^2) = \max_{u_1 \in [0; 64-x^2]} \frac{u^3}{x^2}$$

Диапазон u^3 найден из следующих соображений.

$$x^3 = x^2 + u^3,$$

$$u^3 = x^3 - x^2 = 64 - x^2.$$

Т.к. $x^2 \in [1, 64]$. Функция $\frac{u^3}{x^2}$ при всех значениях x^2 является возрастающей аргумента u^3 , поэтому ее максимум

$$u_*^3(x^2) = 64 - x^2,$$

$$B_3^*(x^2) = \frac{64 - x^2}{x^2}.$$

Шаг 2. С учетом формулы (4)

$$B_2(x^1) = \max_{u^2 \in [0; 64 - x^1]} \left(\frac{64 - x^1 - u^2}{\underbrace{x^1 + u^2}_{x^2 = x^1 + u^2}} + \frac{u^2}{x^1} \right).$$

Находим стационарную точку из условия $\frac{\partial B_2(x^1)}{\partial u^2} :$

$$u_*^2(x_1) = 8\sqrt{x_1} - x_1, \quad B_2^*(x^1) = 2\left(\frac{8}{\sqrt{x_1}} - 1\right).$$

Шаг 3.

$$B_1(x^0) = \max_{u_1 \in [0; 64 - x^0]} \left(2\frac{8}{\sqrt{x_0 + u_1}} - 1 + \frac{u_1}{x_0} \right).$$

$$u_*^1(x_0) = 4(x_0)^{\frac{2}{3}} - x_0, \quad B_1^*(x_0) = 12(x_0)^{-\frac{1}{3}} - 3.$$

Этап II. Безусловная оптимизация.

$$\text{Шаг 0. } x^0 = 1 \Rightarrow x_*^0 = 1.$$

$$\text{Шаг 1. } u_*^1 = u_*^1(1) = 3, \quad x_*^1 = x_*^0 + u_*^1 = 4.$$

$$\text{Шаг 2. } u_*^2 = u_*^2(4) = 12, \quad x_*^2 = x_*^1 + u_*^2 = 16.$$

$$\text{Шаг 3. } u_*^3 = u_*^3(16) = 48, \quad x_*^3 = x_*^2 + u_*^3 = 64.$$

$$u = \{3, 12, 48\},$$

$$x = \{1, 4, 16, 64\},$$

$$B_1^*(x_0) = 9,$$

$$B_2^*(x_1) = 7,$$

$$B_3^*(x_2) = 3.$$

5. НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

В наиболее общем виде задача нелинейного программирования (НЛП) имеет вид.

$$\begin{aligned} f(\bar{x}) &\rightarrow \min \\ g_i(\bar{x}) &\geq b_i, \quad i = \overline{1, m}, \\ h_k(\bar{x}) &= 0, \quad k = \overline{1, l}. \end{aligned} \quad (5.1)$$

Где $f(\bar{x}), g_i(\bar{x}), h_k(\bar{x})$ - заданные, в общем случае нелинейные функции векторной переменной $\bar{x} = (x_1, x_2, \dots, x_n)$.

Ограничения, накладываемые на координаты x_j , могут быть равенствами и неравенствами (I и II рода).

В ряде задач НЛП должно выполняться условие $x_j \geq 0$. Это условие можно записать в виде

$$g_j(\bar{x}) = -x_j, \quad b_j = 0, \quad j = m+1, m+2, \dots$$

Известно, что в задачах безусловной оптимизации необходимым условием существования экстремума является наличие стационарной точки. В задачах с ограничениями это условие может нарушаться.

Следующий пример иллюстрирует нарушение необходимых условий.

Пример.

$$\begin{aligned} f(x) &= (x-2)^2 \rightarrow \min, \\ x &\geq 4. \end{aligned}$$

Найдем стационарную точку

$$\begin{aligned} f'(x) &= 0, \\ 2 \cdot (x-2) &= 0, \\ x_{\text{стац}} &= 2. \end{aligned}$$

Полученная стационарная точка не удовлетворяет ограничению $x \geq 4$. Поэтому точка $x^* = 4$ является оптимальным решением задачи НЛП, хотя не является стационарной.

Приведенный тривиальный пример демонстрирует особенности решения задач НЛП.

В течение последних двух десятилетий из нелинейного программирования выделились самостоятельные разделы:

- выпуклое программирование,
- квадратичное программирование,
- целочисленное программирование,
- стохастическое программирование,
- динамическое программирование и др.

Задачи выпуклого программирования – это задачи, в которых определяется минимум выпуклой функции (или максимум вогнутой), заданной на выпуклом замкнутом множестве. Эти задачи среди задач нелинейного программирования наиболее изучены.

Среди задач выпуклого программирования более подробно изучены задачи квадратичного программирования. В этих задачах целевая функция – квадратична, а ограничения – линейны.

В задачах целочисленного программирования неизвестные параметры могут принимать только целочисленные значения.

В задачах стохастического программирования в целевой функции или в функциях ограничений содержатся случайные величины, которые подчиняются законам теории вероятностей.

В задачах динамического программирования ограничения содержат как параметр время и при этом описываются дифференциальными уравнениями. Процесс нахождения решений в задачах динамического программирования является многоэтапным.

Методы решения задач НЛП различаются в зависимости от вида ограничений.

5.1. Задачи нелинейного программирования с ограничениями в виде равенств

Существует два наиболее распространенных метода решения задач НЛП с ограничениями в виде равенств.

Эта задача может быть сведена к задаче безусловной оптимизации путем исключения из целевой функции l независимых переменных. Число исключаемых переменных равно количеству ограничений-равенств.

Пример. Найти решение задачи НЛП.

$$f(\bar{x}) = x_1 \cdot x_2 \cdot x_3 \rightarrow \min,$$
$$h_1(x) = x_1 + x_2 + x_3 - 1 = 0.$$

Из ограничения $h_1(x) = 0$ выразим x_3

$$x_3 = 1 - x_1 - x_2$$

и подставим в целевую функцию

$$f(x) = x_1 \cdot x_2 \cdot (1 - x_1 - x_2) \rightarrow \min.$$

Получена задача безусловной оптимизации, которая может быть решена любым, рассмотренным выше методом.

Замечание 1. Наличие ограничений в виде равенств фактически позволяет уменьшить размерность задачи оптимизации с n до $(n-1)$.

Замечание 2. Метод исключения переменных может быть применен для ограниченного вида уравнений, которые можно разрешить относительно некоторого конкретного набора независимых переменных.

Например, в следующем уравнении нельзя исключить никакую переменную

$$h(x) = x_1^2 \cdot x_3 + x_2 \cdot x_3^2 + \frac{x_1}{x_2} = 0.$$

В таких случаях рекомендуется использовать метод множителей Лагранжа.

5.1.2. Метод множителей Лагранжа

Основная идея данного метода заключается в следующем. Оптимизационная задача с ограничениями преобразуется в задачу безусловной оптимизации, дополненную некоторыми неизвестными параметрами. Эти параметры носят название неопределенных множителей Лагранжа.

Пусть

$$f(\bar{x}) \rightarrow \min,$$
$$h(\bar{x}) = 0, k = \overline{1, l},$$
$$\bar{x} = x_1, x_2, \dots, x_n.$$
(5.2)

Введем переменные $\lambda_1, \lambda_2, \dots, \lambda_l$ и составим функцию Лагранжа

$$L(\bar{x}, \bar{\lambda}) = f(\bar{x}) + \sum_{k=1}^l \lambda_k \cdot h_k.$$
(5.3)

Задача на условный экстремум (5.2) преобразована к задаче безусловной оптимизации. Множители Лагранжа $\lambda_1, \lambda_2, \dots, \lambda_l$ не имеют ограничения на знак.

Неизвестными задачи безусловной оптимизации (5.3) являются $x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_l$. Для нахождения переменных необходимо взять частные производные

$$\begin{cases} \frac{\partial L(\bar{x}, \bar{\lambda})}{\partial x_j} = 0, & j = 1, n, \\ \frac{\partial L(\bar{x}, \bar{\lambda})}{\partial \lambda_k} = 0, & k = 1, l. \end{cases}$$

Рассмотрим полученную систему уравнений

$$\begin{cases} \frac{\partial L(\bar{x}, \bar{\lambda})}{\partial x_j} = \frac{\partial f(\bar{x})}{\partial x_j} + \sum_{k=1}^l \frac{\partial h_k(\bar{x})}{\partial x_j} = 0, & j = 1, n, \\ \frac{\partial L(\bar{x}, \bar{\lambda})}{\partial \lambda_k} = \partial h_k(\bar{x}) = 0, & k = 1, l. \end{cases} \quad (5.4)$$

Решение системы (5.4) определяет точку метрического пространства $\bar{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$, в которой может иметь место экстремум целевой функции задачи (5.2). Полученное решение \bar{x}^* удовлетворяет и целевой функции и ограничениям. Однако это лишь стационарная точка. Для проверки цели оптимизации, так же как в случае безусловной оптимизации (раздел 3), необходима проверка матрицы Гессе. В случае положительной определенности Гессиана, стационарная точка \bar{x}^* определяет минимум задачи (5.2).

Пример. Решить задачу НЛП методом множителей Лагранжа

$$\begin{aligned} f(\bar{x}) &= x_1^2 + x_2^2 \rightarrow \min, \\ x_1 + x_2 &= 5. \end{aligned}$$

Графическое решение представлено на рисунке 5.1. Для этого построим ограничение и целевую функцию $x_1^2 + x_2^2 = \text{const}$. Выберем константу, позволяющую,

Составим функцию Лагранжа.

$$L(\bar{x}, \bar{\lambda}) = x_1^2 + x_2^2 + \lambda(5 - x_1 - x_2).$$

Для нахождения переменных x_1, x_2, λ необходимо взять частные производные

$$\begin{cases} \frac{\partial L(\bar{x}, \lambda)}{\partial x_1} = 2 \cdot x_1 - \lambda = 0, \\ \frac{\partial L(\bar{x}, \lambda)}{\partial x_2} = 2 \cdot x_2 - \lambda = 0, \\ \frac{\partial L(\bar{x}, \lambda)}{\partial \lambda} = x_1 + x_2 - 5 = 0. \end{cases}$$

Вычтем из первого уравнения системы второе, получим.

$$\begin{cases} x_1 - x_2 = 0, \\ x_1 + x_2 = 5. \end{cases}$$

Откуда

$$\begin{cases} x_1 = x_2 = \frac{5}{2}, \\ \lambda = 5. \end{cases}$$

Проверим определенность матрицы Гессе

$$H(\bar{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

Определители всех главных миноров данного Гессиана положительны. Следовательно, по критерию Сильвестра матрица Гессе положительно определена.

Таким образом, стационарная точка $\bar{x}^* = \left(\frac{5}{2}, \frac{5}{2}\right)$, $f(\bar{x}^*) = \frac{25}{2}$ определяет минимум задачи.

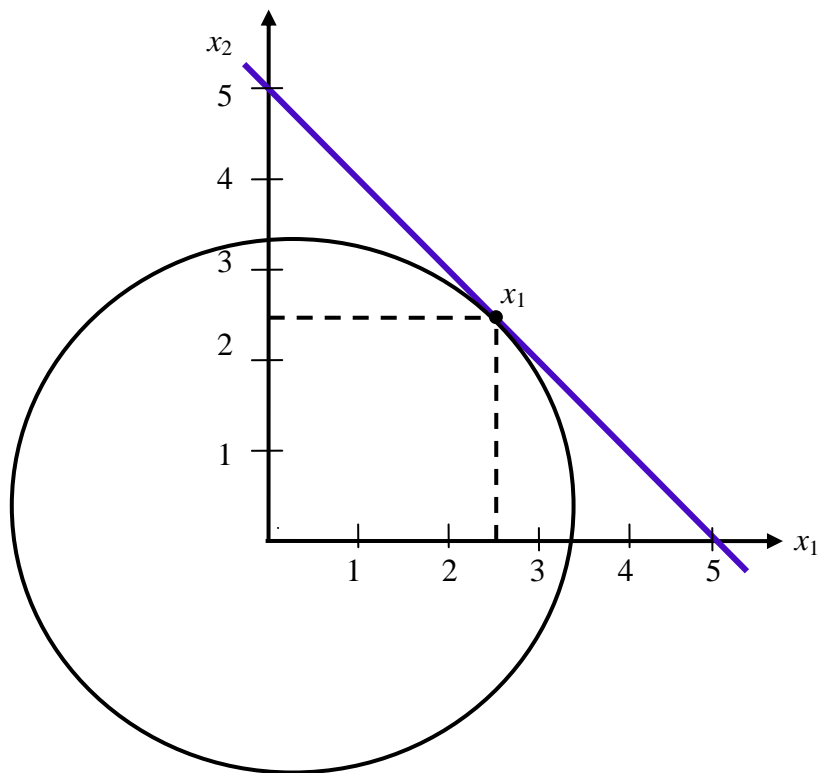


Рисунок 5.1. Графическое решение задачи НЛП

5.2. Задачи нелинейного программирования с ограничениями в виде равенств

Методы решения задач нелинейного программирования разделяются на несколько классов:

- градиентные методы;
- методы линейного программирования;
- методы штрафных функций.

Каждая группа методов используется при определенных условиях, накладываемых на целевую функцию и ограничения. Однако в классе выпуклых функций возможно нахождение оптимального решения задачи НЛП основываясь на отыскании стационарной точки. Такое решение дают условия Куна-Таккера.

5.2.1. Условия Куна-Таккера

Американские ученые Г.Кун и А.Таккер в 1951 году обобщили метод множителей Лагранжа, пригодный для задач НЛП с ограничениями-равенствами, на случай общей задачи НЛП (5.1) с ограничениями в виде равенств и неравенств.

Пусть

$$\begin{aligned} f(\bar{x}) &\rightarrow \min \\ g_i(\bar{x}) &\geq b_i, i = \overline{1, m}, \\ h_k(\bar{x}) &= 0, k = \overline{1, l}. \end{aligned}$$

Кун и Таккер построили необходимые и достаточные условия оптимальности задач НЛП, исходя из предположения о дифференцируемости функций $f(\bar{x})$, $g_i(\bar{x})$, $h_k(\bar{x})$.

Условия оптимальности Куна и Таккера можно сформулировать в виде задачи нахождения решения некоторой системы нелинейных уравнений и неравенств.

Минимум функции $f(\bar{x})$ на допустимом множестве $g_i(\bar{x})$, $h_k(\bar{x})$ может быть найдена, как решение \bar{x}^* , $\bar{\lambda}^*$, $\bar{\gamma}^*$ следующей системы.

$$\begin{aligned} \Delta f(\bar{x}) - \sum_{i=1}^m \gamma_i \cdot \Delta g_i(\bar{x}) - \sum_{k=1}^l \lambda_k \cdot \Delta h_k(\bar{x}), \\ g_i(\bar{x}) \geq 0, i = \overline{1, m}, \\ h_k(\bar{x}) = 0, k = \overline{1, l}, \\ \gamma_i \cdot g_i(\bar{x}) = 0, i = \overline{1, m}, \\ \gamma_i \geq 0, i = \overline{1, m}. \end{aligned} \tag{5.5}$$

Таким образом, условия Куна-Таккера включают обобщенное уравнение Лагранжа, уравнения и неравенства, определяющие ограничения, ограничения на знак множителей γ_i . Предпоследнее уравнение системы (5.5) носит название условий дополнительной нежесткости.

Пример. Найти решение задачи НЛП с использованием

$$f(\bar{x}) = x_1^2 - x_2 \rightarrow \min,$$

$$x_1 + x_2 = 6,$$

$$x_1 - 1 \geq 0,$$

$$x_1^2 + x_2^2 \leq 26.$$

Запишем задачу НЛП в виде стандартной задачи (5.1)

$$f(\bar{x}) = x_1^2 - x_2 \rightarrow \min,$$

$$h_1(\bar{x}) = x_1 + x_2 - 6,$$

$$g_1(\bar{x}) = x_1 - 1,$$

$$g_2(\bar{x}) = 26 - x_1^2 - x_2^2,$$

$$n = 2, \quad m = 2, \quad l = 1.$$

Найдем градиенты функций

$$\Delta f(\bar{x}) = (2 \cdot x_1, -1),$$

$$\Delta h_1(\bar{x}) = 1, 1,$$

$$\Delta g_1(\bar{x}) = 1, 0,$$

$$\Delta g_2(\bar{x}) = (-2 \cdot x_1, -2 \cdot x_2).$$

Обобщенные условия Лагранжа для системы такой размерности имеют вид

$$\frac{\partial f(\bar{x})}{\partial(x_j)} - \gamma_1 \cdot \frac{\partial g_1(\bar{x})}{\partial(x_j)} - \gamma_2 \cdot \frac{\partial g_2(\bar{x})}{\partial(x_j)} - \lambda_1 \cdot \frac{\partial h_1(\bar{x})}{\partial(x_j)} = 0, \quad j = 1, 2.$$

Перепишем их для нашего примера

$$\begin{cases} 2 \cdot x_1 - \gamma_1 \cdot 1 + \gamma_2 \cdot 2 \cdot x_1 - \lambda_1 \cdot 1 = 0, & (j = 1), \\ -1 - \gamma_1 \cdot 0 + \gamma_2 \cdot 2 \cdot x_2 - \lambda_1 \cdot 1 = 0, & (j = 2). \end{cases}$$

Запишем полный вид условий Куна-Таккера

$$\left\{ \begin{array}{l} 2 \cdot x_1 - \gamma_1 + \gamma_2 \cdot 2 \cdot x_1 - \lambda_1 = 0, \\ -1 + \gamma_2 \cdot 2 \cdot x_2 - \lambda_1 = 0, \\ \\ x_1 - 1 \geq 0, \\ 26 - x_1^2 - x_2^2 \geq 0, \\ \\ x_1 + x_2 - 6 = 0, \\ \\ \gamma_1 \cdot x_1 - 1 = 0, \\ \gamma_2 \cdot 26 - x_1^2 - x_2^2 = 0, \\ \\ \gamma_1, \gamma_2 \geq 0. \end{array} \right.$$

Полученная система нелинейных уравнений и неравенств может быть решена известными методами. Отметим, что оптимальное решение получено в пакете программ MathCad

$$\left\{ \begin{array}{l} x_1^* = 1, \quad x_2^* = 5, \\ \gamma_1^* = 3, \quad \gamma_2^* = 0, \\ \lambda_1^* = -1, \\ f(x^*, \gamma^*, \lambda^*) = -4. \end{array} \right.$$

Условия Куна-Таккера могут быть применены для решения относительно небольшого количества задач НЛП и не дают конструктивных методов решения для сложных задач НЛП.

Важный класс градиентных методов применим лишь для выпуклых функций. При этом метод условного градиента применим лишь для задач с линейными ограничениями. Метод Ньютона целесообразно применять в том случае, если целевая функция строго выпуклая и достаточно гладкая в окрестности начальных условий. Метод штрафных функций применим к задачам со сложными ограничениями.

Так же ограниченный класс задач НЛП может быть решен с использованием столь популярных на практике методов линейного программирования. Рассмотрим подробно эту проблему.

5.2.2. Методы линейного программирования

Методы линейного программирования используются для ряда задач НЛП, которые можно свести к задачам линейного программирования.

5.2.2.1 Задача квадратичного программирования

Задача квадратичного программирования представляет собой оптимизационную задачу с линейными ограничениями и квадратичной целевой функцией.

Задача квадратичного программирования в векторно-матричной форме имеет вид:

$$\begin{aligned} f(\bar{x}) &= \bar{c} \cdot \bar{x} + \bar{x}^T \cdot Q \cdot \bar{x} \rightarrow \min \\ A \cdot \bar{x} &= \bar{b}, \\ \bar{x} &\geq 0, \end{aligned} \quad (5.6)$$

где Q – симметричная матрица размера $(n \times n)$;

\bar{c} – вектор-строка размера $(1 \times n)$;

\bar{x} – вектор-столбец размера $(n \times 1)$;

A – матрица ограничений;

\bar{b} – вектор-столбец размера $(m \times 1)$.

Эта задача может быть решена как на основе условий Куна-Таккера, так и методами линейного программирования.

Если матрица Q положительно определена, то целевая функция $f(x)$ является выпуклой функцией и тогда на основании теоремы Куна-Таккера оптимальное решение задачи (5.6) может быть найдено, как решение следующей системы уравнений и неравенств.

$$\begin{aligned} \bar{c} + \bar{x}^T \cdot (Q + Q^T) - \bar{\gamma} - \bar{\lambda} \cdot A &= 0 \\ A \cdot \bar{x} &= \bar{b}, \\ \bar{x} &\geq 0, \\ \bar{\gamma} \cdot \bar{x} &= 0, \\ \bar{\gamma} &\geq 0. \end{aligned} \quad (5.7)$$

Система (5.7) получена на основе условий Куна-Таккера (5.5), полагая

$$h \bar{x} = A \cdot \bar{x} - \bar{b} = 0,$$

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

$$\Delta h \bar{x} = A,$$

$$\Delta g \bar{x} = 1.$$

Решение задачи квадратичного программирования симплекс-методом рассмотрим на конкретном примере. Симплекс-метод может быть использован в случае только ограничений неравенств. Сведение задачи к стандартному виду приведено в разделе (4.3).

Пример.

Решить задачу квадратичного программирования симплекс-методом.

$$\left. \begin{aligned} f(\bar{x}) &= (x_1 - 4)^2 - (x_2 - 5)^2 \rightarrow \min \\ x_1 + x_2 &\leq 6 \\ x_1 - x_2 &\leq 1 \\ x_1, x_2 &\geq 0 \end{aligned} \right\}$$

Построим многогранник ограничений линии уровня целевой функции, рисунок 5.2.

Многогранник ограничений – замкнутый. Экстремум целевой функции может находиться внутри многогранника или на его границах, ребрах или в вершинах.

Найдем стационарную точку целевой функции, определяющую абсолютный экстремум.

$$\left. \begin{aligned} \frac{f(\bar{x})}{\partial x_1} &= 2 \cdot (x_1 - 4) = 0, \\ \frac{f(\bar{x})}{\partial x_2} &= 2 \cdot (x_2 - 5) = 0, \\ x_1^* &= 4, x_2^* = 5. \end{aligned} \right\}$$

На рисунке 5.2 стационарная точка A_0 соответствует центру линий уровня, но она не принадлежит многограннику ограничений.

Если вершин и ребер немного, как в данном примере, то возможен обычный перебор, то есть симплекс.

В таблице 5.1 приведены координаты вершин и значения целевой функции в них.

Таблица 5.1

Вершины многогранника ограничений

Вершина	Координаты	Целевая функция
A	0, 0	41
B	0, 6	17
C	3.5, 2.5	6.5
D	1, 0	34

Для нахождения экстремумов на ребрах многогранника можно использовать метод множителей Лагранжа, так как ограничения неравенства заменены равенством.

Получим четыре уравнения Лагранжа.

Точка A1.

$$\left. \begin{aligned} L_{x_1, x_2, \lambda_1} &= (x_1 - 4)^2 + (x_2 - 5)^2 + \lambda_1 \cdot x_1 + x_2 - 6, \\ x_1^* &= 2.5, \quad x_2^* = 3.5, \\ f_{2.5, 3.5} &= 4.5. \end{aligned} \right\}$$

Точка A2.

$$\left. \begin{aligned} L_{x_1, x_2, \lambda_1} &= (x_1 - 4)^2 + (x_2 - 5)^2 + \lambda_2 \cdot x_1 - x_2 - 1, \\ x_1^* &= 5, \quad x_2^* = 4, \\ f_{5, 4} &= 2. \end{aligned} \right\}$$

Однако точка A2 не принадлежит многограннику ограничений.

Точка A3.

$$\left. \begin{aligned} L_{x_1, x_2, \lambda_1} &= (x_1 - 4)^2 + (x_2 - 5)^2 + \lambda_3 \cdot x_2, \\ x_1^* &= 0, \quad x_2^* = 5, \\ f_{0, 5} &= 16. \end{aligned} \right\}$$

Точка A4.

$$\left. \begin{aligned} L(x_1, x_2, \lambda_1) &= (x_1 - 4)^2 + (x_2 - 5)^2 + \lambda_1 \cdot x_1, \\ x_1^* &= 4, \quad x_2^* = 0, \\ f(4, 0) &= 25. \end{aligned} \right\}$$

Таким образом оптимальное решение соответствует точке A1.

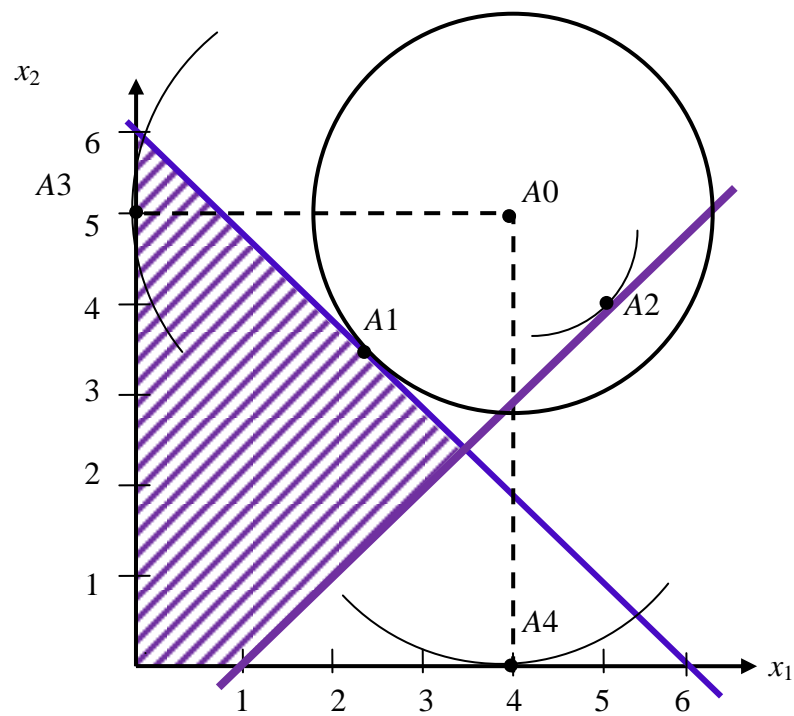


Рисунок 5.2. Графическое решение задачи квадратичного программирования

СПИСОК ЛИТЕРАТУРЫ

1. Акулич И.Л. Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. спец. вузов. - М.: Высш. шк., 1986.-319с.
2. Аоки М. Введение в методы оптимизации. - М.: «Наука», 1977.-334 с.
3. Аттетков А. В., Галкин С. В., Зарубин В. С. Методы оптимизации.-М.: Издательство: МГТУ им. Н. Э. Баумана, 2003.-441 с.
4. Ашманов С.А. Линейное программирование. М.: «Наука», 1981.-304с.
5. Ашманов С.А. Линейное программирование. Учебное пособие. - М.: Наука, 1981. - 340 с.
6. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы. М., Мир, 1982.- 583 с.
7. Банди Б. Методы оптимизации. Вводный курс. - М.: Мир, 1988.-128 с.
8. Банди Б. Основы линейного программирования: Пер. сангл. - М.: Радио и связь, 1989. - 176 с.
9. Батищев Д.И. Методы оптимального проектирования. - М.: Радио и связь, 1984.-64с.
10. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы.-М.: Бином, 2008.-632с.
11. Беллман Р. Динамическое программирование. – Издательство иностранной литературы, 1960.-400с.
12. Березин И.С., Жидков Н.П. Методы вычислений. Том 1 и 2. – М.: «Наука», 1994.-464с.
13. Бирюков С.И. Оптимизация. – М.: МФТИ, 1995.- 248с.
14. Болтянский В.Г. Математические методы оптимального управления. М.: Наука, 1969. - 408 с.
15. Булавский В.А. Численные методы линейного программирования. - М: Наука, 1977. - 367с.
16. Васильев Ф.П. Численные методы решения экстремальных задач. – М.: Наука, 1988.-549с.
17. Васильев Ф.П., Иваницкий А.Ю. Линейное программирование. М.: "Факториал", 1998. - 176 с.
18. Габасов Р., Кириллова Ф. М., Костюкова О. И., Покатаев А. В. Конструктивные методы оптимизации. Ч. 5. Нелинейные задачи.- Минск.: Университетское, 1998. - 390 с.

19. Гилл Ф. Практическая оптимизация. - М: Мир, 1985. - 510с.
20. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. Пер. с англ. - М.: Мир, 1985.-509с.
21. Грешиллов А.А. Прикладные задачи математического программирования: Учебное пособие. - 2-е изд. - М.: Логос, 2006. – 288.
22. Гроот. Оптимальные статистические решения. - М: Мир, 1974. - 496с.
23. Данциг Д. Линейное программирование, его применения и обобщения. - М., Прогресс, 1966. - 600 с.
24. Демьянов В.Ф., Васильев Л.В. Недифференцируемая оптимизация. - М.: Наука, 1981. - 384 с.
25. Ермаков С.М., Жиглявский А.А. Математическая теория оптимального эксперимента: Учеб. пособие. - М.: «Наука», 1987.
26. Жиглявский А.А., Жилинкас А.Г. Методы поиска глобального экстремума. - М.: Наука, Физматлит, 1991.-248с.
27. Жолобов Д.А. Введение в математическое программирование. Учебное пособие – М.: МИФИ, 2008.- 376 с.
28. Зайченко Ю.П. Исследование операций - Киев, "Вища школа", 1979, 391с. (уч.пособие для студентов университетов и тех. вузов).
29. Зангвилл У. Нелинейное программирование. Единый подход. - М.: Сов. радио, 1973.-312с.
30. Зельдович Я.Б. Элементы прикладной математики. - М: Наука, 1972. - 592с.
31. Зуховицкий С.И., Авдеева Л.И. Линейное и выпуклое программирование.– М.: «Наука», 1967.-460с..
32. Интрилигатор М. Математические методы оптимизации и экономическая теория - М., Прогресс, 1975, 605 с.,(пер. с англ.)
33. Калиткин Н.Н. Численные методы. – М.: «Наука», 1978.-512с.
34. Калихман И.Л., Войтенко М.А. Динамическое программирование в примерах и задачах. М., Высшая школа, 1979.-125с.
35. Карманов В.Г. Математическое программирование. - М.: «Наука», 1986.-285с.
36. Карманов В.Г. Математическое программирование. - М.:ФИЗМАТЛИТ, 2004.-264с.
37. Корбут А.А., Финкельштейн Ю.Ю. Дискретное программирование. – М.: Наука, 1969.-368с.
38. Коршунов Ю.М., Коршунов Ю.М. Математические основы кибернетики. - М.: Энергоатомиздат, 1987.-494с.
39. Кузнецов А.В., Сакович В.А., Холод Н.И. Математическое программирование. Минск: Высшая школа. 1994. - 286 с.

40. Кузнецов Ю.Н. и др. Математическое программирование. М., Высшая шк., 1980, 304 с.(уч.пособие для экономистов).
41. Лесин В.В., Лисовец Ю.П. Основы методов оптимизации. – М.: МАИ, 1998.
42. Литвак Б.Л. Теория и численные методы решения задач оптимизации. - М.: Изд-во МИНГП, 1977.
43. Лунгу К. Н. Линейное программирование. Руководство к решению задач. - М.: ФИЗМАТЛИТ, 2005. - 128 с.
44. Максимов Ю.А.,Филлиповская Е.А. Алгоритмы решения задач нелинейного программирования. - М.: МИФИ, 1982.
45. Математические методы и модели в планировании (под ред. А.И.Карасева), 1987, 239с. (уч.пособие для студентов экономических специальностей).
46. Мину М. Математическое программирование. Теория и алгоритмы.-М.: Наука. Гл. ред. физ.-мат. лит., 1990.- 488с.
47. Моисеев Н.Н. Методы оптимизации. - М: Наука, 1978. - 351с.
48. Мэтьюз Д.Г., Финк К.Д. Численные методы. Использование MATLAB.- М.: Вильямс, 2001.-720с.
49. Ногин В.Д. и др. Основы теории оптимизации. М., Высшая школа, 1986.-384с.
50. Основы теории оптимального управления (под ред. В.Ф.Кротова)- М., Высшая школа, 1979, 126 с.(уч.пособие для студентов экономических специальностей).
51. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985.-512с.
52. Поляк Б.Т. Введение в оптимизацию. - М.: «Наука», 1983.-384с.
53. Растрингин Л.А. Статистические методы поиска. - М.: 1968.-376с.
54. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике: в 2-х кн. /Пер. с англ. - М.: Мир, 1986.-672с.
55. Сборник задач по математике для втузов. Ч.4. Методы оптимизации. Уравнения в частотных производных. Интегральные уравнения: Учеб. пособ. /Вуколов Э.А., Ефимов А.В., Земсков В.Н. и др.; Под ред. А.В.Ефимова. - М.: «Наука». Гл.ред.физ.-мат.лит., 1990.
56. Сборник задач по математике для ВТУЗов.(методы оптимизации, уравнения в частных производных, интегральные уравнения). Под ред. А.Ф.Ефимова, М., Наука, 1990.-607с.
57. Свиридов А. Т. Задачи динамического программирования. Учебное пособие. – Калининград: КГТУ, 2006.

58. Свиридов А. Т. Нелинейные задачи математического программирования. Учебное пособие. – Калининград: КГТУ, 2004, 84 с.
59. Сеа Ж. Оптимизация. Теория и алгоритмы. - М.: Мир, 1973.-244с.
60. Справочник по математике для экономистов (под ред. В.И.Ермакова)- М., Высшая школа, 1987, 336 с.(для использования в учебном процессе студентами экономических специальностей).
61. Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. - М.: «Наука», 1986.-326с.
62. Уальд Д. Методы поиска экстремума. - М.: «Наука», 1967.-267с.
63. Уальд Д. Оптимальное проектирование. - М.: Мир, 1981.-272с.
64. Химмельблау Д. Прикладное нелинейное программирование. - М.: Мир, 1975.-534с.
65. Хофер Э., Лундерштедт Р. Численные методы оптимизации. - М.: Машиностроение, 1981.-192с.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Б

базисное решение, 97

В

вычислительный эксперимент, 4

Г

градиент, 42

градиентные методы, 60, 72, 78

Гаусса метод, 98

Гомори метод, 120

Д

дихотомии метод, 27

двойственность ЗЛП, 113

З

задача

 транспортная, 85

 технического контроля, 83

 о диете, 86

 об использовании сырья, 87

 золотого сечения метод, 30

И

Иенсена неравенство, 17

итерация, 22

интервал неопределенности, 25

искусственного базиса метод, 108

К

критерий

 минимаксный, 28

 оптимизации, 9

 останова, 24, 61

Коши метод, 67

Куна-Таккера условия, 140

М

Математическое программирование, 3
Методы оптимизации, 4
многомерная безусловная оптимизация, 15
минимаксная стратегия, 28
матрица Гессе, 43
множителей Лагранжа метод, 137

Н

Ньютона метод, 79
Ньютона метод модифицированный, 82
Ньютона – Рафсона метод, 39

О

оптимизационная задача, 3
одномерная безусловная оптимизация, 15, 25
оврагов метод, 58
опорный план, 97
оптимальная траектория, 129

П

программирование, 3
 выпуклое, 3, 15, 136
 динамическое, 125, 136
 квадратичное, 136, 143
 линейное, 83
 нелинейное, 135
 стохастическое,
 целочисленное, 120, 136
постановка задачи оптимизации, 10
Пауэлла метод, 35
покоординатного спуска метод, 48
прямого поиска методы, 52

Р

равномерного поиска метод, 32

С

стационарная точка, 18
сходимость алгоритма, 21-23
сходимости скорость, 23

Сильвестра Критерий, 43
сопряженность векторов, 72
стандартная форма ЗЛП, 93
симплекс – метод, 96, 101

T

Теория оптимизации, 4,6
Теорема исключения интервалов, 26
точечного оценивания методы, 35

У

устойчивость метода, 21

Ф

функция
униmodalная, 17
выпуклая, 17
сепарабельная, 45

Х

Хука Дживса метод, 53

Ц

целевой функции, 10

Э

экстремум, 19,20

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1. ПОСТАНОВКА И КЛАССИФИКАЦИЯ ЗАДАЧ ОПТИМИЗАЦИИ	4
1.1. История становления и развития теории оптимизации	5
1.2. Содержательная и формализованная постановка задачи	6
1.3. Постановка задачи оптимизации	11
1.4. Классификация задач оптимизации	11
1.5. Анализ экстремальных задач условия существования экстремума	16
1.6. Характеристики алгоритмов оптимизации	20
1.7 Критерии останова итерационных методов	22
2. МОДЕЛИ И МЕТОДЫ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ	24
2.1. Методы исключения интервала неопределенности	24
2.1.1. Метод половинного деления (дихотомии)	26
2.1.2. Минимаксная стратегия поиска оптимума	27
2.1.3. Метод золотого сечения	29
2.1.3. Метод равномерного поиска	31
2.1.4. Сравнительный анализ методов исключения интервалов	32
2.2. Методы точечного оценивания	34
2.3. Методы одномерного поиска с использованием производных	37
2.3.1. Метод Ньютона – Рафсона	38
2.4. Сравнение методов одномерного поиска оптимума	39
3. МЕТОДЫ БЕЗУСЛОВНОЙ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ	40
3.1. Метод покоординатного спуска	45
3.2. Методы поиска безусловного экстремума	50
3.3. Методы прямого поиска	51
3.3.1. Метод Хука Дживса	52
3.3.2. Метод оврагов	58
3.4. Градиентные методы	59
3.4.1. Метод наискорейшего спуска (метод Коши)	66
3.4.2. Метод сопряженных градиентов	72
3.5. Градиентные методы второго порядка	77
3.5.1. Модифицированный метод Ньютона	81
4. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ	83
4.1. Классические задачи линейного программирования	83
4.1.1. Задача технического контроля	83
4.1.2. Транспортная задача	85
4.1.3. Задача о диете	86
4.1.4. Задача об использовании сырья	87
4.2. Геометрический метод решения задач линейного программирования	89
4.3. Задача линейного программирования в стандартной форме	93
4.4. Идея и алгебра симплекс – метода	96
4.4.1. Метод последовательного исключения переменных (метод Гаусса)	98
4.4.2. Метод искусственного базиса	108
4.5. Двойственность задач линейного программирования	113

4.5.1. Экономическая трактовка двойственности	118
4.7. Динамическое программирование	120
4.7. Динамическое программирование	125
5. НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ.....	135
5.1. Задачи нелинейного программирования с ограничениями в виде равенств	136
5.1.2. Метод множителей Лагранжа.....	137
5.2. Задачи нелинейного программирования с ограничениями в виде равенств	140
5.2.1. Условия Куна-Таккера.....	141
5.2.2. Методы линейного программирования	144
5.2.2.1 Задача квадратичного программирования.....	144
СПИСОК ЛИТЕРАТУРЫ	148
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	152
ОГЛАВЛЕНИЕ	155

Учебное издание

КОЧЕГУРОВА Елена Алексеевна

ТЕОРИЯ И МЕТОДЫ ОПТИМИЗАЦИИ

Учебное пособие

Научный редактор *доктор технических наук,*
профессор Г.П. Цапко

Редактор *О.Н. Свинцова*

Компьютерная верстка *Е.А. Кочегурова*

Дизайн обложки *Т.А. Фатеева*

Подписано к печати 05.09.2012. Формат 60х84/16. Бумага «Снегурочка».


Печать XEROX. Усл.печ.л. . Уч.-изд.л.

Заказ . Тираж 100 экз.



Национальный исследовательский Томский политехнический
университет
Система менеджмента качества
Томского политехнического университета сертифицирована
NATIONAL QUALITY ASSURANCE по стандарту ISO 9001:2008



ИЗДАТЕЛЬСТВО  **ТПУ**. 634050, г. Томск, пр. Ленина, 30
Тел./факс: 8(3822)56-35-35, www.tpu.ru