

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

МАТЕМАТИЧЕСКИЕ МЕТОДЫ И МОДЕЛИ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

под редакцией А.Г. Реннера

Учебное пособие

Рекомендовано ученым советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательным программам высшего образования по направлению подготовки 01.03.04 Прикладная математика

Оренбург
2021

УДК 330.45:519.87(075.8)

ББК 65в631я73

М34

Рецензент – кандидат экономических наук Е.М. Крипак

Авторы: О.Н. Яркова, А.В. Раменская, А.Г. Реннер, О.С. Чудинова

М34 **Математические методы и модели исследования операций**: учебное пособие / О.Н. Яркова, А.В. Раменская, А.Г. Реннер, О.С. Чудинова; под ред. А.Г. Реннера; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2021. – 296 с.

В учебном пособии рассмотрены модели, методы и алгоритмы решения задач исследования операций, в том числе задач безусловной оптимизации, нелинейного программирования, многокритериальной оптимизации и сетевого планирования.

Учебное пособие предназначено для студентов направления подготовки 01.03.04 Прикладная математика, а также других направлений подготовки при изучении дисциплин «Математические методы и модели исследования операций», «Методы оптимизации», «Методы оптимальных решений», «Численные методы», «Математическое моделирование». Учебное пособие может быть полезно студентам и магистрантам направлений подготовки Бизнес-информатика и Экономика.

УДК 330.45:519.87(075.8)

ББК 65в631я73

© Яркова О.Н.

Раменская А.В.

Реннер А.Г.

Чудинова О.С., 2021

© ОГУ, 2021

Содержание

| | |
|---|----|
| Введение | 7 |
| 1 Методы безусловной оптимизации | 9 |
| 1.1 Численные методы безусловной минимизации функции одной переменной. Методы прямого линейного поиска | 10 |
| 1.1.1 Квазивыпуклые функции и их свойства..... | 10 |
| 1.1.2 Пассивные и последовательные алгоритмы | 12 |
| 1.1.3 Сходимость итерационных последовательностей | 13 |
| 1.1.4 Метод равномерного поиска..... | 15 |
| 1.1.5 Метод удвоения шага (используется для выделения отрезка неопределенности, содержащего точку минимума)..... | 17 |
| 1.1.6 Метод половинного деления..... | 19 |
| 1.1.7 Метод золотого сечения..... | 21 |
| 1.1.8 Метод полиномиальной (квадратичной) аппроксимации (метод интерполяционных кривых) | 23 |
| 1.1.9 Вопросы для самопроверки | 28 |
| 1.1.10 Задания для практических (лабораторных) работ | 29 |
| 1.1.11 Задания для самостоятельной проработки и подготовке к контрольным работам..... | 31 |
| 1.2 Численные методы безусловной минимизации функции многих переменных | 32 |
| 1.2.1 Релаксационная последовательность. Сходимость. Критерии останова.... | 32 |
| 1.2.2 Методы спуска. Принципы выбора длины шага в методах спуска | 37 |
| 1.2.3 Методы нулевого порядка. Метод циклического покоординатного спуска | 43 |
| 1.2.4 Методы нулевого порядка. Метод Хука-Дживса | 47 |
| 1.2.5 Методы первого порядка. Методы градиентного спуска..... | 52 |
| 1.2.6 Методы второго порядка. Метод Ньютона и его модификации | 57 |

| | |
|--|-----|
| 1.2.7 Квазиньютоновские методы | 60 |
| 1.2.8 Методы сопряженных направлений, сопряженных градиентов | 63 |
| 1.2.9 Вопросы для самопроверки | 71 |
| 1.2.10 Задания для практических (лабораторных) работ | 72 |
| 1.3 Критерии масштабирования и останова | 74 |
| 1.3.1 Масштабирование..... | 75 |
| 1.3.2 Критерии останова | 76 |
| 1.3.3 Задания для практических (лабораторных) работ | 77 |
| 1.3.4 Задания для самостоятельной проработки и подготовке к контрольным работам..... | 78 |
| 2 Нелинейное программирование | 79 |
| 2.1 Постановка задачи нелинейного программирования в общей форме..... | 79 |
| 2.2 Аналитические методы решения задач нелинейного программирования | 80 |
| 2.2.1 Геометрическая интерпретация двумерной задачи нелинейного программирования | 80 |
| 2.2.2 Классическая задача на условный экстремум..... | 82 |
| 2.2.3 Метод множителя Лагранжа | 83 |
| 2.2.4 Конечномерная задача нелинейного программирования с ограничениями в виде равенств и неравенств | 88 |
| 2.2.5 Задача выпуклого нелинейного программирования..... | 94 |
| 2.2.6 Общая теория нелинейного программирования..... | 94 |
| 2.2.7 Условия существования седловой точки | 100 |
| 2.2.8 Теория двойственности нелинейного программирования | 101 |
| 2.2.9 Вопросы для самопроверки | 104 |
| 2.2.10 Задания для практических (лабораторных) работ | 104 |
| 2.2.11 Задания для самостоятельной проработки и подготовке к контрольным работам..... | 108 |
| 2.3 Численные методы решения задач нелинейного программирования | 109 |
| 2.3.1 Метод проекции градиента | 110 |
| 2.3.2 Метод условного градиента | 114 |

| | | |
|-------|--|-----|
| 2.3.3 | Методы штрафных функций, общая идеология | 118 |
| 2.3.4 | Метод внешних штрафных функций..... | 120 |
| 2.3.5 | Метод внутренних (барьерных) штрафных функций..... | 124 |
| 2.3.6 | Комбинированный метод штрафных функций..... | 127 |
| 2.3.7 | Вопросы для самопроверки | 128 |
| 2.3.8 | Задания для практических (лабораторных) работ | 128 |
| 2.3.9 | Задания для самостоятельной проработки и подготовке к контрольным работам..... | 132 |
| 3 | Динамическое программирование | 134 |
| 3.1 | Принцип оптимальности Белмана..... | 134 |
| 3.2 | Моделирование задачи ДП | 137 |
| 3.3 | Этапы решения задачи ДП..... | 139 |
| 3.4 | Оптимальное распределение инвестиций как задача ДП | 140 |
| 3.6 | Задача о рюкзаке | 150 |
| 3.7 | Вопросы для самопроверки | 153 |
| 3.8 | Задания для практических (лабораторных) занятий | 153 |
| 3.9 | Задания для самостоятельной проработки и подготовке к контрольным работам | 156 |
| 4 | Модели и методы многокритериальной оптимизации..... | 158 |
| 4.1 | Математическая модель задачи многокритериальной оптимизации | 158 |
| 4.2 | Принцип доминирования по Парето | 162 |
| 4.3 | Методы решения многокритериальных задач, использующие дополнительную информацию о критериях и свойствах оптимального решения | 167 |
| 4.3.1 | Метод оптимизации основного частного критерия..... | 168 |
| 4.3.2 | Метод свертки критериев (построения обобщенного скалярного критерия) | 170 |
| 4.3.3 | Лексикографическая оптимизация | 173 |
| 4.3.4 | Метод последовательных уступок..... | 175 |
| 4.3.5 | Метод идеальной точки..... | 178 |

| | |
|---|-----|
| 4.4 Интерактивные методы решения задач многокритериальной оптимизации | 182 |
| 4.5 Нормирование критериев | 188 |
| 4.6 Целевое программирование | 190 |
| 4.6.1 Постановка задачи | 190 |
| 4.6.2 Архимедова модель | 194 |
| 4.6.3 Задачи целевого программирования с приоритетами. Метод уступок | 205 |
| 4.6.4 ЦП эффективность и проблема чувствительности | 207 |
| 4.7 Вопросы для самопроверки | 210 |
| 4.8 Задания для практических (лабораторных) занятий | 211 |
| 4.9 Задания для самостоятельной проработки и подготовке к контрольным работам | 219 |
| 5 Модели сетевого планирования | 225 |
| 5.1 Основные понятия и определения | 225 |
| 5.2 Правила построения сетевого графика | 231 |
| 5.3 Параметры сетевого графика | 234 |
| 5.4 Оценка времени выполнения работ в условиях неопределенности | 246 |
| 5.5 Анализ и оптимизация сетевого графика | 250 |
| 5.6 Программные средства для сетевого моделирования | 254 |
| 5.7 Вопросы для самопроверки | 265 |
| 5.8 Задания для практических (лабораторных работ) | 265 |
| 5.9 Задания для самостоятельной проработки и подготовке к контрольным работам | 275 |
| Список использованных источников | 278 |
| Приложение А (справочное) Учет погрешности приближенных вычислений ... | 282 |
| Приложение Б (справочное) Элементы выпуклого анализа | 284 |
| Приложение В (справочное) Квадратичная форма | 293 |
| Приложение Г (справочное) Теоремы отделимости | 295 |

Введение

Методы прикладной математики широко используются в различных областях человеческой деятельности, в частности, в экономике, финансах и управлении. Только в предыдущем столетии за разработку эффективного математического инструментария Нобелевской премии по экономике были удостоены: Р. Солоу, В. Леонтьев, П. Самуэльсон, Г. Марковиц, Д. Тобин, Л. Канторович и др.

Дисциплина «Математические методы и модели исследования операций» посвящена количественному обоснованию принимаемых управленческих решений. Она включает в себя широкий круг задач, решение большинства из которых требует использования численных методов, знания алгоритмов и программных средств их реализации.

Учебное пособие разработано с целью дать читателю представление о численных методах безусловной и условной оптимизации, аналитических методах исследования задач нелинейного программирования, методах динамического программирования, задачах оптимизации с несколькими критериями и моделях сетевого планирования.

Пособие состоит из пяти глав, каждая из которых содержит контрольные вопросы для самопроверки, задания к лабораторным и практическим занятиям.

Первая глава содержит теоретические сведения о методах безусловной оптимизации функции одной и многих переменных.

Во второй главе представлены методы решения задач нелинейного программирования, включая численные алгоритмы решения таких задач.

Третья глава описывает принцип оптимальности Беллмана для решения задач динамического программирования. Приведены примеры задач динамического программирования и вычислительные схемы их решения.

В четвертой главе дана постановка задачи многокритериальной оптимизации и приводится подробный обзор методов для её решения.

Пятая глава посвящена моделям сетевого планирования, в ней описаны задачи оптимизации на сетевых графиках. Для анализа и оптимизации сетевых графиков использовано собственное программное средство.

В конце каждой темы приведены задания для практических занятий, лабораторных работ и самостоятельной проработки.

Разделы учебного пособия могут использоваться также при освоении дисциплин «Методы оптимальных решений», «Методы принятия управленческих решений», «Численные методы», «Методы оптимизации».

1 Методы безусловной оптимизации

Дополнительную информацию по приведенному разделу, включая доказательства приведенных теорем можно посмотреть, обратившись к источникам: 37.

ОПРЕДЕЛЕНИЕ. Будем говорить, что функция $f(x_1, x_2, \dots, x_n) \equiv f(x)$ имеет локальный минимум (максимум) в точке $x^* = (x_1^{(*)}, \dots, x_n^{(*)})$, если существует σ -окрестность $E_\sigma(x^{(*)})$ точки $x^{(*)}$ такая, что $f(x^{(*)}) \leq f(x)$ ($f(x^{(*)}) \geq f(x)$) для всех $x \in E_\sigma(x^{(*)})$. Если эти неравенства справедливы для всех \bar{x} из области определения, то будем говорить, что $x^{(*)}$ является точкой глобального минимума (максимума) в области определения функции.

Поскольку задача определения максимума $f(x)$ эквивалентна задаче нахождения минимума $-f(x)$, то в дальнейшем говорим только о минимизации функции.

Классический подход к задаче нахождения $x^{(*)}$ – точек локального минимума состоит для дважды дифференцируемой функции:

1) в определении таких точек x , в которых все первые частные производные 1-го порядка $\frac{\partial f(x^*)}{\partial x_i} = 0$ обращаются в нуль (необходимое условие);

2) проверка положительной определенности матрицы Гессе $H(x) = \left(\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right)$

($i = 1, \dots, n; j = 1, \dots, n$) в таких точках (достаточное условие).

Существенную трудность при таком подходе составляют как первый, так и второй этап решения задачи, поэтому зачастую более предпочтительными являются численные методы.

1.1 Численные методы безусловной минимизации функции одной переменной. Методы прямого линейного поиска

Материал раздела выполнен в соответствии с источниками: 3, 5, 7, 28, 31.

1.1.1 Квазивыпуклые функции и их свойства

Описываемые ниже методы минимизации требуют, чтобы минимизируемая функция обладала свойством квазивыпуклости.

ОПРЕДЕЛЕНИЕ. Будем говорить, что функция $f(x)$ квазивыпукла (или унимодальна) на отрезке $[a, b]$, если для всех $x_1, x_2 \in [a, b]$ выполняется неравенство $f(\lambda x_1 + (1-\lambda)x_2) \leq \max(f(x_1), f(x_2))$, где $\lambda \in [0, 1]$, если неравенство строгое, то $f(x)$ строго квазивыпукла.

Квазивыпуклость функции – это обобщение понятия выпуклости функции. Произвольная выпуклая функция является квазивыпуклой.

Непрерывная функция $f(x)$ квазивыпукла на отрезке $[a, b]$ если выполняется одно из следующих условий:

1. $f(x)$ – неубывающая;
2. $f(x)$ – невозрастающая;
3. существует такая точка $c \in [a, b]$ что для всех $x \leq c$ функция $f(x)$ – невозрастающая, а для всех $x \geq c$ функция $f(x)$ – неубывающая.

Таким образом, квазивыпуклость функции на отрезке $[a, b]$ гарантирует наличие строго одного минимума. Примеры квазивыпуклой и неквазивыпуклой функций приведены на рисунке 1.1.

Будем искать минимум функции $f(x)$ на некотором промежутке $[a, b]$. В основе предлагаемых ниже численных методов приближенного отыскания

минимума лежит теорема.

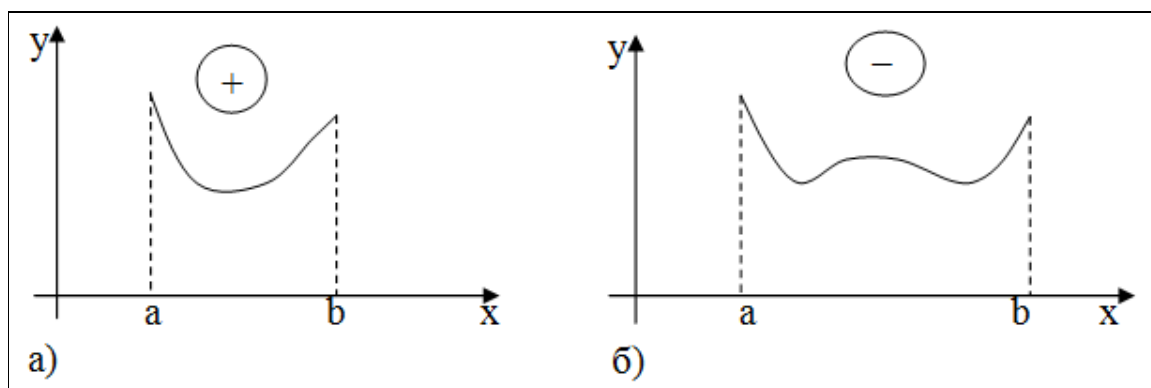


Рисунок 1.1 – Функции: а) – квазивыпуклая; б) – неквазивыпуклая

ТЕОРЕМА 1.1

Пусть $f(x)$ строго квазивыпукла на $[a, b]$. Пусть c, d – любые точки принадлежащие $[a, b]$ такие, что $c < d$. Если $f(c) > f(d)$, то $f(x) \geq f(d)$ для всех $x \in [a, c]$. Если $f(c) \leq f(d)$, то $f(x) \geq f(c)$ для всех $x \in [d, b]$ (рисунок 1.2).

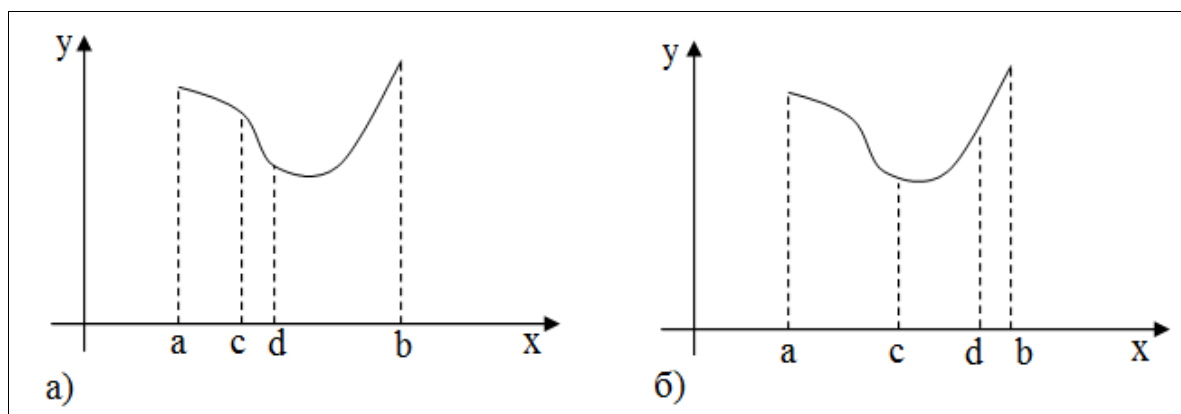


Рисунок 1.2 – Квазивыпуклая функция: а) $f(c) > f(d)$; б) $f(c) \leq f(d)$

СЛЕДСТВИЕ: Пусть $f(x)$ строго квазивыпукла на $[a, b]$. Пусть c, d – любые точки принадлежащие $[a, b]$ такие, что $c < d$, тогда:

1. если $f(c) > f(d)$, то точка минимума $x^* \in [c, b]$;
2. если $f(c) < f(d)$, то точка минимума $x^* \in [a, d]$;
3. если $f(c) = f(d)$, то точка минимума $x^* \in [c, d]$.

Процедуру исключения отрезка, основанную на приведенном следствии используют в методах прямого поиска точки минимума квазивыпуклой функции одной переменной. Сужая, в соответствии со следствиями, отрезок, на котором находится точка минимума, мы сможем сделать длину такого отрезка (интервала неопределенности) меньше требуемой точности ε и взять в качестве приближенного решения любую точку \tilde{x}^* полученного промежутка.

1.1.2 Пассивные и последовательные алгоритмы

Поставим задачу нахождения минимума функции одной переменной

$$\begin{aligned} f(x) &\rightarrow \min, \\ x &\in [a, b] \subset \mathbb{R}. \end{aligned} \tag{1.1}$$

Пусть численный метод решения задачи (1.1) основан на точном или приближенном вычислении характеристик функции. Для каждой конкретной задачи выбор характеристики производится в зависимости от свойств функции.

Алгоритмы, использующие лишь информацию о значениях функции, называют алгоритмами или методами нулевого порядка. Алгоритмы, использующие информацию о значениях 1-ых производных, называют методом 1-го порядка и т.д.

Работа любого алгоритма состоит из следующих этапов:

- 1) на первом этапе вычисляются предусмотренные алгоритмом характеристики задачи;
- 2) на втором этапе строится приближенное решение задачи. Как правило, для задания алгоритма достаточно указать схему его реализации и правило оценки погрешностей.

Если все точки выбираются одновременно до начала вычисления, то алгоритм называется *пассивным*.

Алгоритм, в котором точки строятся поочередно называют *последовательным*, при этом используют значение функции, вычисленное на предыдущих этапах.

Будем считать, что стратегия поиска определена, если определен алгоритм выбор точек $x^{(k)}$ – приближений к точке минимума x^* функции и определено условие прекращения поиска (критерий останова), т.е. условие, при выполнении которого значение $f^*=f(x^*)$ считают найденным с заданной точностью. Для методов пассивного поиска алгоритм выбора точек $x^{(k)}$ – это правило, по которому заранее определяются все точки $x^{(k)}$, $k=1,..N$ в которых затем будет вычислено значение функции $f(x^{(k)})$.

Для метода последовательного поиска количество точек обычно не задают заранее ($k=1,2,..$) и после k вычислений обычно указывают интервал или отрезок длины $l^{(k)}$ (который называют интервалом неопределенности) в котором гарантированно находится x^* . Условие прекращения итераций в случае пассивного и последовательного поиска применяется одинаковое, в простейшем случае, это выполнение условия, $l^{(k)} \leq \varepsilon$, где ε – заданная наибольшая допустимая длина интервала неопределенности.

1.1.3 Сходимость итерационных последовательностей

Пусть некоторый итерационный процесс задается в виде:

$$x^{(k+1)} = \varphi(x^{(k)}). \quad (1.2)$$

Итерационный процесс (1.2) генерирует последовательность $\{x^{(k)}\}_{k=0}^{\infty}$, имеющую $\lim_{k \rightarrow \infty} x^{(k)}$.

ОПРЕДЕЛЕНИЕ. Будем говорить, что метод (1.2) сходится, если

последовательность $x^{(k)}$ сходится к x^* , при $k \rightarrow \infty$ (т.е. $\lim_{k \rightarrow \infty} \{x^{(k)}\} = x^*$) и x^* – решение задачи (1.1), т.е. $x^* = \operatorname{argmin} f(x)$.

Сходимость итерационного процесса означает, что погрешность каждого последующего приближения должна быть меньше погрешности предыдущего приближения, т.е. погрешность приближенных значений с каждым шагом должна уменьшаться $\|x^* - x^{(k+1)}\| < \|x^* - x^{(k)}\|$, где под $\|\bullet\|$ понимается норма вектора [10].

В общем случае это неравенство можно представить в виде $\|x^* - x^{(k+1)}\| < C \|x^* - x^{(k)}\|^p$, где $(C > 0)$ и $p \geq 1$ – некоторые числа, значения которых определяются методом вычисления приближенного решения. От значений C и p зависит насколько с каждым шагом уменьшается погрешность приближенных значений и, соответственно, насколько быстро можно получить приближенное значение с заданной точностью. Главным показателем скорости сходимости метода является значение p , называемое порядком сходимости.

ОПРЕДЕЛЕНИЕ. Сходимость последовательности $x^{(k)} \rightarrow x^*$ называют линейной, а соответствующий итерационный процесс линейно сходящимся, если существует такая постоянная C принадлежащая интервалу $(0,1)$ и такой номер k_0 , что начиная с него выполняется

$$\|x^* - x^{(k+1)}\| \leq C \|x^* - x^{(k)}\|, \forall k \geq k_0, k \in \mathbb{N}. \quad (1.3)$$

ОПРЕДЕЛЕНИЕ. Сходимость последовательности $x^{(k)} \rightarrow x^*$ называют сверхлинейной, если существует такая положительная последовательность $\{C_k\}_{k=0}^{\infty}$, имеющая предел $\lim_{k \rightarrow \infty} C_k = 0$ и выполняется условие:

$$\|x^* - x^{(k+1)}\| \leq C_k \|x^* - x^{(k)}\|, \forall k \geq k_0, k \in \mathbb{N}. \quad (1.4)$$

ОПРЕДЕЛЕНИЕ. Говорят, что последовательность $x^{(k)} \rightarrow x^*$, сходится по меньшей мере с p -ым порядком (соответствующий итерационный процесс имеет порядок p), если найдутся константы $C > 0$ и $p \geq 1$, что выполнится условие

$$\|x^* - x^{(k+1)}\| \leq C \|x^* - x^{(k)}\|^p, \forall k \geq k_0, k \in \mathbb{N}.$$

Если в выражение (1.3) C не удастся найти, но установлено неравенство (1.4) и $C_k \rightarrow C \in [0,1]$, то говорят об асимптотически линейной сходимости. Аналогично определяется асимптотический p -й порядок.

1.1.4 Метод равномерного поиска

Пусть $f(x)$ квазивыпуклая на $[a, b]$ функция.

$$f(x) \rightarrow \min, \\ x \in [a, b] \subset \mathbb{R}.$$

Покроем отрезок $[a, b]$ равномерной сеткой узлов с шагом h :

$$x^i = a + ih, i = 0, 1, \dots, n, h = \frac{(b - a)}{n}.$$

Вычислим значения функции во всех узлах сетки: $f(x^i)$, $i = 0, 1, \dots, n$.

Выберем среди найденных значений минимальное. Допустим, минимальное значение функция принимает в узле x^k .

Тогда, на основании теоремы 1.1, можем утверждать, что точка минимума $f(x)$ принадлежит интервалу (x^{k-1}, x^{k+1}) . Таким образом, интервал

неопределенности сокращен до длины $2h$ (рисунок 1.3).

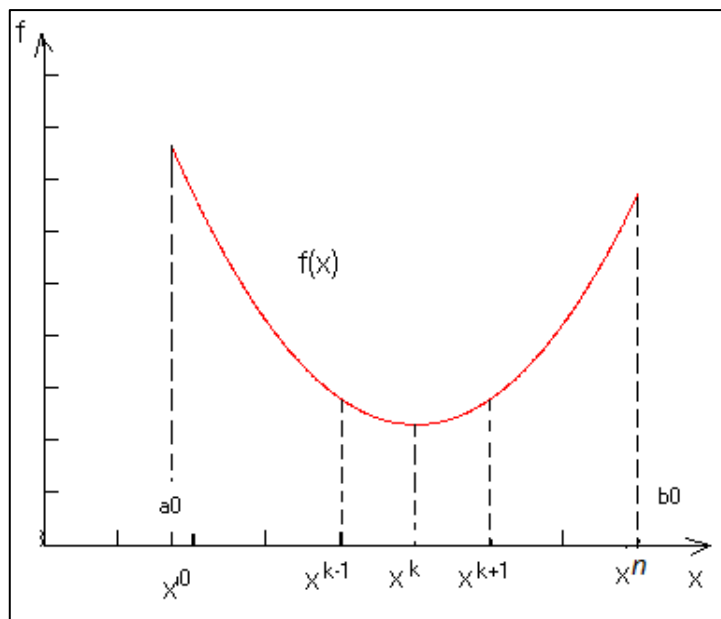


Рисунок 1.3 – Метод равномерного поиска

Для метода равномерного поиска характеристика относительного уменьшения начального интервала неопределенности находится по формуле $R(n) = \frac{2}{n}$.

Далее может быть применена описанная процедура для нового отрезка неопределенности $a=x^{k-1}$, $b=x^{k+1}$ до тех пор, пока его длина не станет меньше заданного значения точности вычислений ε , т.е. $|a - b| < \varepsilon$.

Недостатки метода: большое количество вычислений, медленная сходимость если n велико.

Пример 1.1

Пусть дана квазивыпуклая на отрезке $[0,1]$ функция $f(x) = 3x^2 - 2x + \exp(-x)$ необходимо найти отрезок неопределенности длиной не более 0.2, содержащий точку минимума.

Из соотношения $R(n) = \frac{2}{n} = 0.2$, находим $n = 10$. Вычислим шаг

$h = \frac{b-a}{n} = \frac{1-0}{10} = 0.1$. Далее вычисляем значения функции в точках $x^i = 0 + i * 0.1, i = 0, 1..n$ (значения приведены в таблице 1.1).

Таблица 1.1 – Пример таблицы значений для метода равномерного поиска

| | | | | | | | | | | | |
|----------|---|------|------|------|------|-------|--------|------|------|------|------|
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| x^i | 0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 1 |
| $f(x^i)$ | 1 | 0,71 | 0,44 | 0,22 | 0,06 | -0,02 | -0,003 | 0,13 | 0,39 | 0,79 | 1,37 |

Из полученных результатов можно сделать вывод, что интервал неопределенности (0.4;0.6), т.е. $x^* \in (0.4;0.6)$.

1.1.5 Метод удвоения шага (используется для выделения отрезка неопределенности, содержащего точку минимума)

Метод удвоения шага используется для выделения отрезка неопределенности, содержащего точку минимума.

Пусть $f(x)$ непрерывная функция. Поставим задачу выделения отрезка $[a,b]$ содержащего точку минимума функции $f(x)$.

Алгоритм:

Шаг 0. $k=1, x^0$ – начальная точка;

Шаг 1. Определяем направление убывания функции $f(x)$. Для этого выбирают некоторое число h (шаг) и вычисляют $f(x^0 + h)$;

Шаг 1.1 Если $f(x^0 + h) < f(x^0)$, то полагают $x^1 := x^0 + h$ и переходят к шагу 2;

Шаг 1.2 Если $f(x^0 + h) \geq f(x^0)$, то полагают $h := -h$ и вычисляют $f(x^0 + h)$;

Шаг 1.3 Если $f(x^0 + h) < f(x^0)$, то $x^1 := x^0 + h$ и переход к шагу 2, если $f(x^0 + h) \geq f(x^0)$, то $h := \frac{h}{2}$ и повторяют шаг 1.

В результате шага 1 получают h (где h – направление убывания функции) и точки x^1 и x^0 такие что: $f(x^1) < f(x^0)$;

Шаг 2. Удваиваем $h := 2h$ и $x^{k+1} := x^k + h$;

Шаг 3. Находим значение $f(x^{k+1})$;

Шаг 3.1 Если $f(x^{k+1}) < f(x^k)$, тогда $k=k+1$ и переходим к шагу 2;

иначе Если $f(x^{k+1}) \geq f(x^k)$, то поиск останавливают и в качестве отрезка содержащего точку минимума выбирают отрезок $[x^{k-1}, x^{k+1}]$.

Конец алгоритма.

ЗАМЕЧАНИЕ. Если необходимо определить отрезок неопределенности длиной ε , то процедуру нужно повторять, начиная с пункта 1, где в качестве $x^0 := x^{k-1}$ до тех пор пока длина отрезка неопределенности не станет меньше ε .

Рассмотрим геометрическую интерпретацию алгоритма на примере некоторой функции, график которой представлен на рисунке 1.4.

Пусть выбрана некоторая точка x^0 и шаг $(-h)$. Делаем шаг в направлении $(-h)$ и сравниваем значения функции в точках x^0 и $x^0 - h$. По графику видно, что $f(x^0) < f(x^0 - h)$, следовательно направление выбрано неверно. Проанализируем поведение функции в направлении h . Так как значение функции уменьшилось при движении в этом направлении ($f(x^0) > f(x^0 + h)$), следовательно $x^1 := x^0 + h$, и переходим к шагу 2 алгоритма. Удваиваем шаг и $x^2 := x^1 + 2h$. $f(x^2) < f(x^1)$, следовательно делаем следующий шаг в этом направлении длиной $4h$: $x^3 := x^2 + 4h$. Так как $f(x^3) < f(x^2)$, следовательно делаем следующий шаг в этом направлении длиной $8h$: $x^4 := x^3 + 8h$. Значение функции при этом увеличилось: $f(x^4) > f(x^3)$, следовательно в качестве отрезка содержащего точку минимума берем отрезок

$$[x^2, x^4].$$

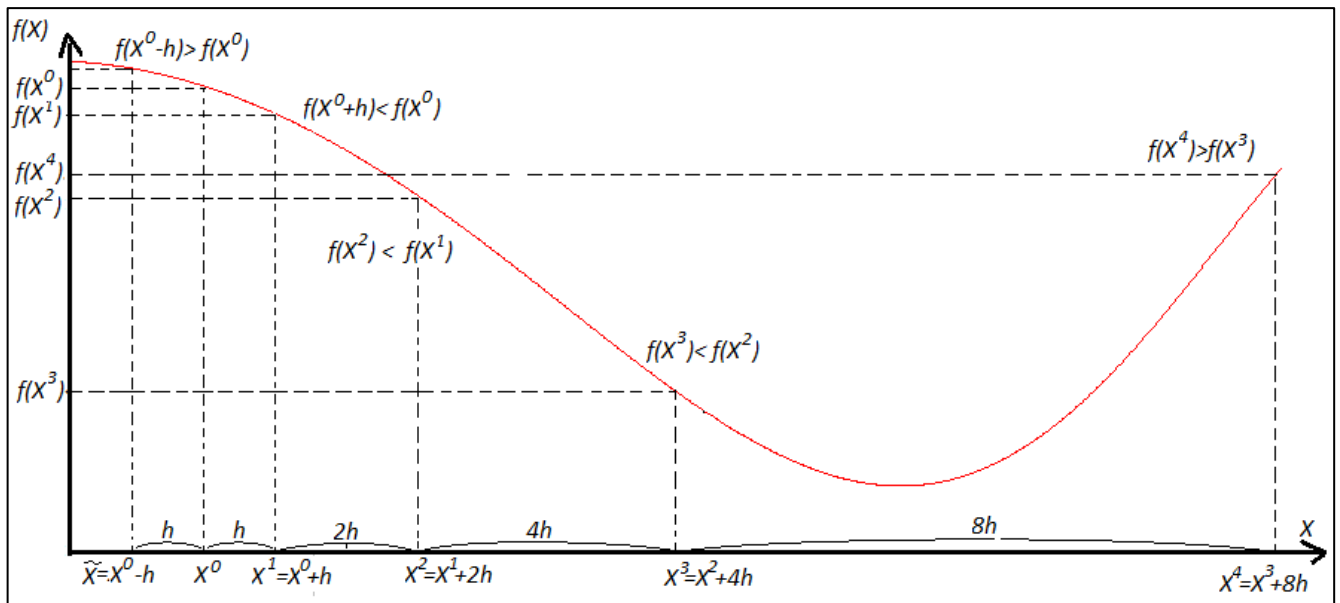


Рисунок 1.4 – Геометрическая интерпретация метода удвоения шага

1.1.6 Метод половинного деления

Пусть $f(x)$ квазивыпуклая на $[a, b]$ функция.

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

$$x \in [a, b] \subset \mathbb{R}.$$

Необходимо определить отрезок неопределенности длиной ε .

Алгоритм:

Шаг 0. Пусть на k -ом шаге исключения отрезок неопределенности имеет вид $[a^k, b^k]$;

Шаг 1. Находим середину отрезка по формуле $c = \frac{a^k + b^k}{2}$. Вычисляем точки

$$x^1 = c - \delta, \quad x^2 = c + \delta;$$

Шаг 2. Вычисляем значение функции в этих точках $f(x^1)$ и $f(x^2)$:

Шаг 2.1 Если $f(x^1) > f(x^2)$, тогда $x^* \in [x^1, b^k]$, т.е. $a^{k+1} = x^1$, $b^{k+1} = b^k$;

иначе Если $f(x^1) < f(x^2)$, тогда $x^* \in [a^k, x^2]$, т.е. $b^{k+1} = x^2$,

$a^{k+1} = a^k$;

иначе Если $f(x^1) = f(x^2)$, тогда $x^* \in [x^1, x^2]$, т.е. $a^{k+1} = x^1$,

$b^{k+1} = x^2$;

Шаг 3. Если $|b^{k+1} - a^{k+1}| < \varepsilon$, тогда $x^* \in [a^{k+1}, b^{k+1}]$;

иначе $k=k+1$, идем на шаг 1.

Конец алгоритма.

Недостаток метода – медленная сходимость.

Геометрическая демонстрация алгоритма приведена на рисунке 1.5.

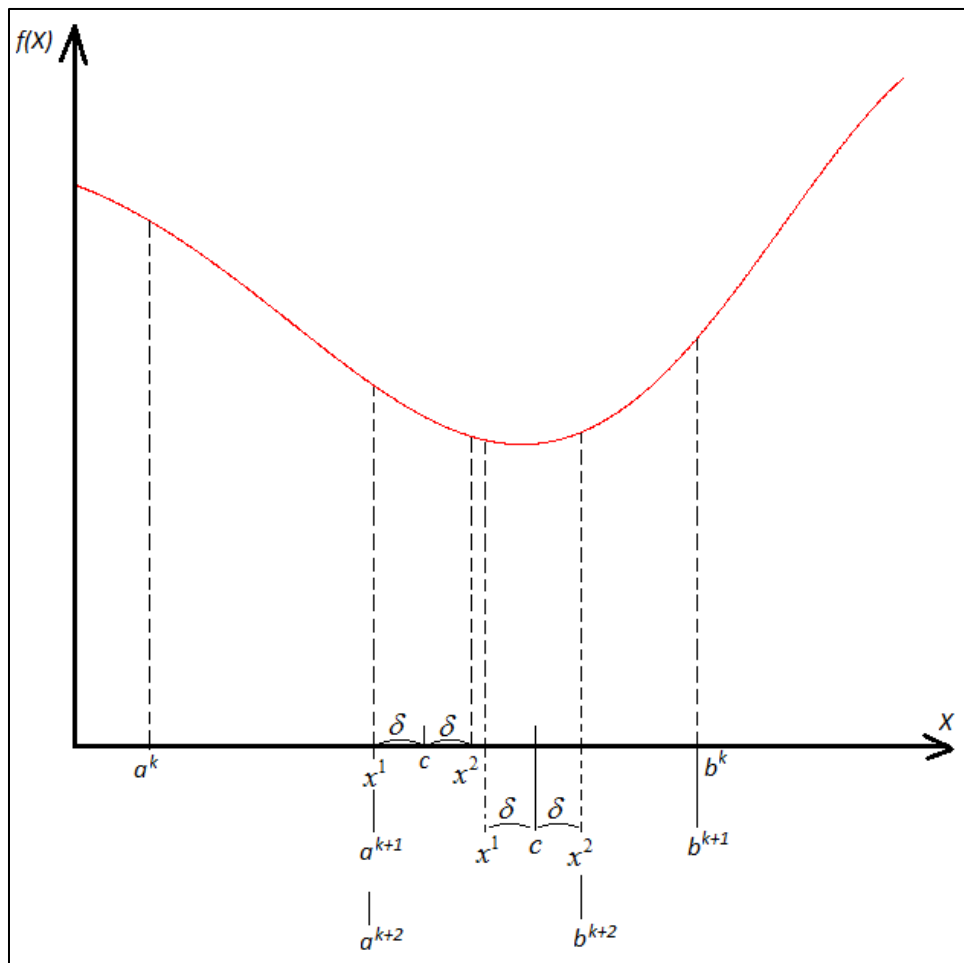


Рисунок 1.5 – Метод половинного деления

1.1.7 Метод золотого сечения

Естественно желание достичь интервала неопределенности требуемой длины при минимуме вычислений. Одним из наиболее эффективных в этом плане методов, является метод золотого сечения.

Золотым сечением отрезка называют такое его деление на две неравные части, при котором отношение длины всего отрезка к длине его большей части равно отношению длины большей части к длине меньшей части.

Таким образом, золотое сечение отрезка $[a, b]$ производит две симметричные точки $x^1 = a + (1 - \tau)(b - a)$, $x^2 = a + \tau(b - a)$, где $\tau = 0,618$.

ЗАМЕЧАНИЕ. При таком делении длина нового интервала неопределенности $b^k - a^k$ не зависит от того, выполняется ли $f(x^1) > f(x^2)$ или $f(x^1) \leq f(x^2)$. При этом x^1 производит сечение отрезка $[a; x^1]$, а x^2 – золотое сечение отрезка $[x^2; b]$. (рисунок 1.6)

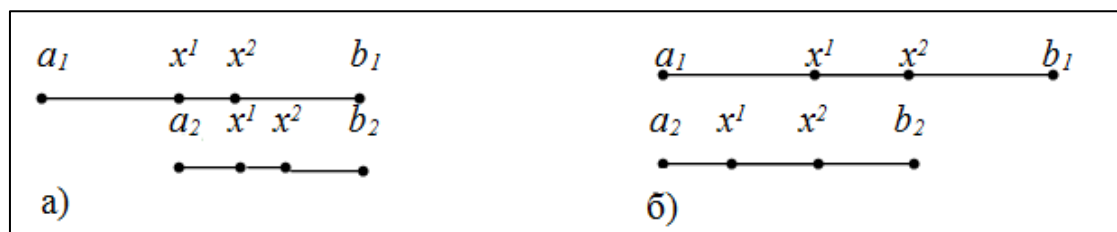


Рисунок 1.6 – Деление отрезка : а) $f(x^1) > f(x^2)$; б) $f(x^1) \leq f(x^2)$

Пусть $f(x)$ квазивыпуклая на $[a, b]$ функция.

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

$$x \in [a, b] \subset \mathbb{R}.$$

Необходимо определить отрезок неопределенности длиной ε .

Алгоритм:

Шаг 0. Пусть заданы ε – точность вычислений, отрезок $[a;b]$, $k=1$;

Шаг 1. $a^k = a$, $b^k = b$;

Шаг 2. Находим $x^1 = a + (1 - \tau)(b - a)$, $x^2 = a + \tau(b - a)$, где $\tau = 0,618$;

Шаг 3. Вычисляем $f(x^1)$ и $f(x^2)$;

Шаг 4. Выполняем процедуру исключения:

Если $f(x^1) \leq f(x^2)$, тогда $(x^* \in [a^k; x^2])$ $a^{k+1} = a^k$, $b^{k+1} = x^2$, $x^2 = x^1$,
 $x^1 = a^{k+1} + (1 - \tau)(b^{k+1} - a^{k+1})$;

иначе Если $f(x^1) > f(x^2)$, тогда $(x^* \in [x^1, b^k])$, т.е. $a^{k+1} = x^1$, $b^{k+1} = b^k$,
 $x^1 = x^2$, $x^2 = a^{k+1} + \tau(b^{k+1} - a^{k+1})$;

Шаг 5. Если $|b^{k+1} - a^{k+1}| < \varepsilon$, тогда конец алгоритма, $x^* \in [a^{k+1}, b^{k+1}]$,

иначе $k=k+1$, переход к шагу 3;

Конец алгоритма.

Геометрическая реализация алгоритма приведена на рисунке 1.7.

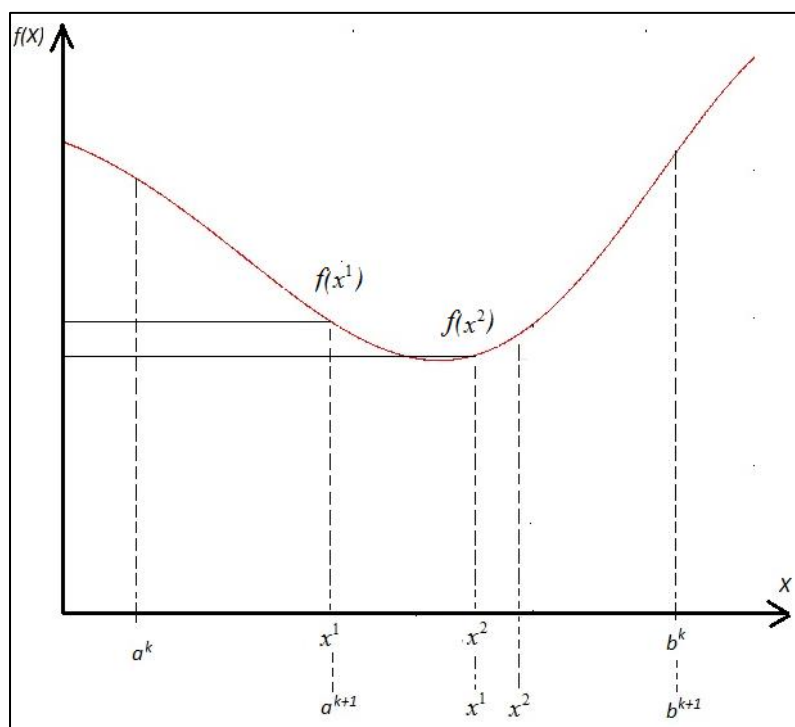


Рисунок 1.7 – Метод золотого сечения

1.1.8 Метод полиномиальной (квадратичной) аппроксимации (метод интерполяционных кривых)

Если функцию в некоторой окрестности точки ее минимума можно достаточно точно заменить (аппроксимировать) многочленом, то для ее минимизации целесообразно использовать так называемые методы полиномиальной аппроксимации. Их общая идея состоит в вычислении коэффициентов многочлена по известным значениям функции в отдельных точках и последующем нахождении минимума этого многочлена с использованием необходимых и достаточных условий экстремума. Рассмотрим метод квадратичной аппроксимации. В нем график исследуемой функции $f(x)$, квазивыпуклой на $[a, b]$, приближенно заменяется параболой, проходящей через три известные точки $(x_i, f(x_i)), i = 1, 2, 3$.

Известно, что по трем различным точкам, не принадлежащим одной прямой, можно единственным образом построить параболу $g(x) = ax^2 + bx + c, a \neq 0$. Коэффициенты можно найти исходя из системы линейных алгебраических уравнений:

$$\begin{cases} ax_1^2 + bx_1 + c = f(x_1) \\ ax_2^2 + bx_2 + c = f(x_2) \\ ax_3^2 + bx_3 + c = f(x_3). \end{cases}$$

Когда координаты $x_i, i = 1, 2, 3$ попарно различны система имеет единственное решение, которое можно представить в виде:

$$\begin{aligned} a &= \frac{f(x_1)}{(x_1 - x_2)(x_1 - x_3)} + \frac{f(x_2)}{(x_2 - x_1)(x_2 - x_3)} + \frac{f(x_3)}{(x_3 - x_1)(x_3 - x_2)}, \\ b &= \frac{-(x_2 + x_3)f(x_1)}{(x_1 - x_2)(x_1 - x_3)} + \frac{-(x_1 + x_3)f(x_2)}{(x_2 - x_1)(x_2 - x_3)} + \frac{-(x_1 + x_2)f(x_3)}{(x_3 - x_1)(x_3 - x_2)}, \\ c &= \frac{x_2 x_3 f(x_1)}{(x_1 - x_2)(x_1 - x_3)} + \frac{x_1 x_3 f(x_2)}{(x_2 - x_1)(x_2 - x_3)} + \frac{x_1 x_2 f(x_3)}{(x_3 - x_1)(x_3 - x_2)}. \end{aligned}$$

Если найденные выражения подставить в формулу вершины параболы, которая при $a > 0$ является точкой минимума параболы (а при $a < 0$ - максимума), получим:

$$\tilde{x}^* = -\frac{b}{2a} = \frac{1}{2} \frac{f(x_1)r_{23} + f(x_2)r_{31} + f(x_3)r_{12}}{f(x_1)s_{23} + f(x_2)s_{31} + f(x_3)s_{12}},$$

где $r_{ij} = x_i^2 - x_j^2$, $s_{ij} = x_i - x_j$, $i, j = 1, 2, 3$.

Если известен отрезок, на котором функция $f(x)$ квазивыпукла, то нет необходимости проверять значение коэффициента a . Достаточно этот отрезок принять в качестве отрезка $[x_1, x_3]$, а точку x_2 выбирать произвольно в интервале (x_1, x_3) . В этом случае выполняются условия $x_1 < x_2 < x_3$, $f(x_1) \geq f(x_2)$ и $f(x_3) \geq f(x_2)$, следовательно $\tilde{x}^* \in [x_1, x_2]$.

В результате указанных действий мы получим четыре точки $x_1, x_2, x_3, \tilde{x}^*$. Далее из них выбирают новую тройку точек $x'_i, i = 1, 2, 3$ по правилу:

а) Если $\tilde{x}^* \in [x_2, x_3]$ и $f(\tilde{x}^*) \leq f(x_2)$, то $x'_1 = x_2$, $x'_3 = x_3$, $x'_2 = \tilde{x}^*$ (рисунок 1.8);

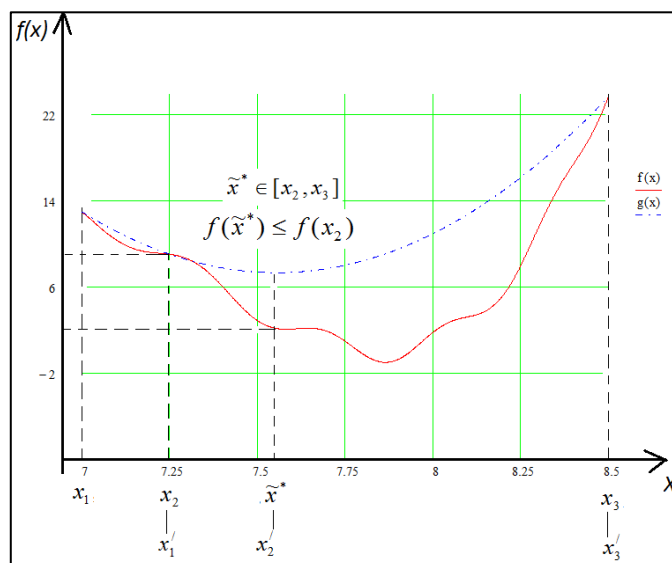


Рисунок 1.8 – Метод квадратичной аппроксимации для случая $\tilde{x}^* \in [x_2, x_3]$ и $f(\tilde{x}^*) \leq f(x_2)$

б) Если $\tilde{x}^* \in [x_2, x_3]$ и $f(\tilde{x}^*) > f(x_2)$, то $x'_1 = x_1$, $x'_2 = x_2$, $x'_3 = \tilde{x}^*$ (рисунок 1.9);

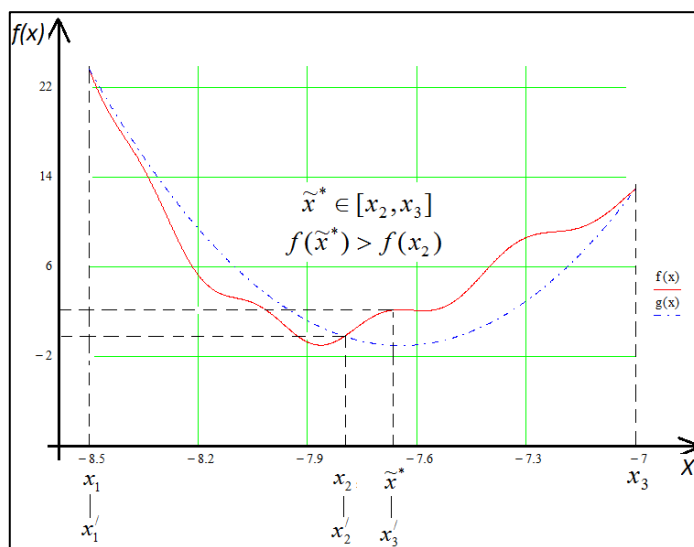


Рисунок 1.9 – Метод квадратичной аппроксимации для случая $\tilde{x}^* \in [x_2, x_3]$ и $f(\tilde{x}^*) > f(x_2)$

в) Если $\tilde{x}^* \in [x_1, x_2]$ и $f(\tilde{x}^*) \leq f(x_2)$, то $x'_1 = x_1$, $x'_3 = x_2$, $x'_2 = \tilde{x}^*$ (рисунок 1.10);

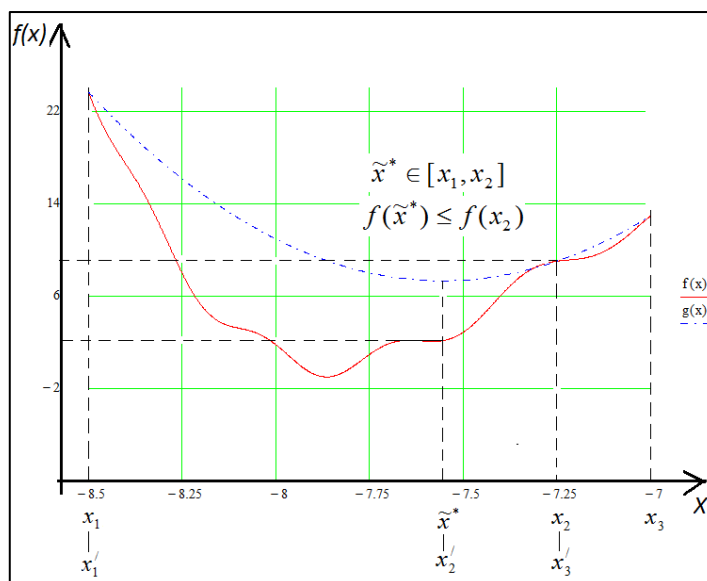


Рисунок 1.10 – Метод квадратичной аппроксимации для случая $\tilde{x}^* \in [x_1, x_2]$ и $f(\tilde{x}^*) \leq f(x_2)$

г) Если $\tilde{x}^* \in [x_1, x_2]$ и $f(\tilde{x}^*) > f(x_2)$, то $x_1' = \tilde{x}^*$, $x_3' = x_3$, $x_2' = x_2$ (рисунок 1.11).

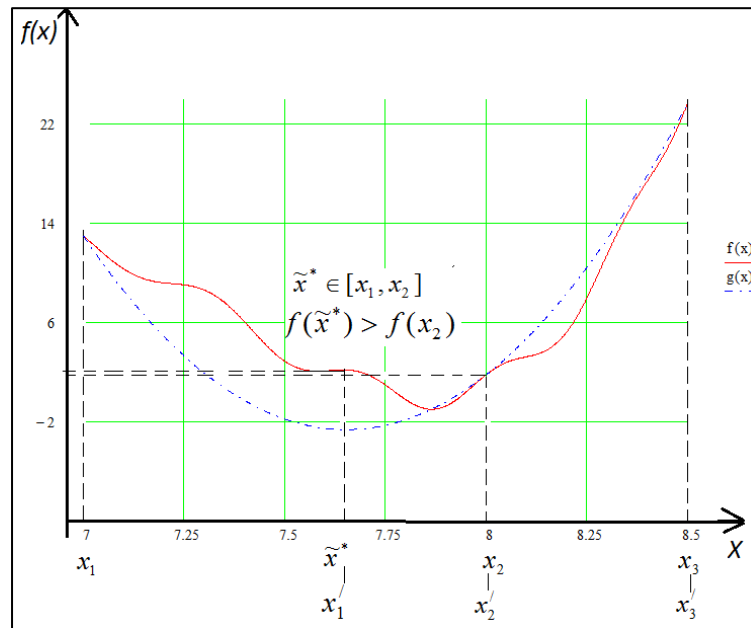


Рисунок 1.11 – Метод квадратичной аппроксимации для случая $\tilde{x}^* \in [x_1, x_2]$ и $f(\tilde{x}^*) > f(x_2)$

Для новой тройки чисел $x_i', i = 1, 2, 3$ процедуру повторяют, и т.д. до тех пор, пока длина интервала неопределенности не станет меньше заданной точности.

Рассмотрим алгоритм метода квадратичной аппроксимации. Пусть $f(x)$ квазивыпуклая на $[a, b]$ функция.

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

$$x \in [a, b] \subset \mathbb{R}.$$

Алгоритм:

Шаг 0. Выбираем точки $x_1^{(0)} = a$, $x_3^{(0)} = b$, $x_2^{(0)} : x_1^{(0)} < x_2^{(0)} < x_3^{(0)}$. Зададим точность ε , $k=1$;

Шаг 1. Вычисляем \tilde{x}^* по формуле:

$$\tilde{x}^{*(0)} = \frac{f(x_1^{(0)}) \cdot r_{23} + f(x_2^{(0)}) \cdot r_{31} + f(x_3^{(0)}) \cdot r_{12}}{2 \cdot (f(x_1^{(0)}) \cdot s_{23} + f(x_2^{(0)}) \cdot s_{31} + f(x_3^{(0)}) \cdot s_{12})},$$

где $r_{ij} = (x_i^{(0)})^2 - (x_j^{(0)})^2$, $s_{ij} = x_i^{(0)} - x_j^{(0)}$, $i, j = 1, 2, 3$.

Находим значение функции $f(\tilde{x}^{*(k)}) = \tilde{f}_*^{(k)}$;

Шаг 2. Выбираем новую тройку точек $x_i^{(k+1)}$ по правилу:

a) Если $\tilde{x}^{*(k)} \in [x_2^{(k)}, x_3^{(k)}]$ и $\tilde{f}_*^{(k)} \leq f(x_2^{(k)})$, тогда $x_1^{(k+1)} = x_2^{(k)}$, $x_3^{(k+1)} = x_3^{(k)}$,
 $x_2^{(k+1)} = \tilde{x}^{*(k)}$;

b) Если $\tilde{x}^{*(k)} \in [x_2^{(k)}, x_3^{(k)}]$ и $\tilde{f}_*^{(k)} > f(x_2^{(k)})$, тогда $x_1^{(k+1)} = x_1^{(k)}$,
 $x_2^{(k+1)} = x_2^{(k)}$, $x_3^{(k+1)} = \tilde{x}^{*(k)}$;

c) Если $\tilde{x}^{*(k)} \in [x_1^{(k)}, x_2^{(k)}]$ и $\tilde{f}_*^{(k)} \leq f(x_2^{(k)})$, тогда $x_1^{(k+1)} = x_1^{(k)}$, $x_3^{(k+1)} = x_2^{(k)}$,
 $x_2^{(k+1)} = \tilde{x}^{*(k)}$;

d) Если $\tilde{x}^{*(k)} \in [x_1^{(k)}, x_2^{(k)}]$ и $\tilde{f}_*^{(k)} > f(x_2^{(k)})$, тогда $x_1^{(k+1)} = \tilde{x}^{*(k)}$, $x_3^{(k+1)} = x_3^{(k)}$,
 $x_2^{(k+1)} = x_2^{(k)}$;

Шаг 3. $k=k+1$;

Шаг 4. Если $|\tilde{x}^{*(k)} - \tilde{x}^{*(k-1)}| < \varepsilon$, тогда $x^* \in [\tilde{x}_*^{(k-1)}; \tilde{x}_*^{(k)}]$, иначе шаг 1;

Конец алгоритма.

Метод квадратичной аппроксимации применяется после локализации точки минимума при достаточно малом отрезке $[a, b]$. Эффективность метода возрастает в случае дифференцируемости функции, т.к. производная такой функции в окрестности точки минимума мала и при разложении функции по формуле Тейлора в этой окрестности наиболее существенный вклад вносит квадратичное слагаемое, содержащее множитель $(x-x_k)^2$. Поэтому многочлен второго порядка в этом случае

достаточно хорошо аппроксимирует функцию в окрестности точки минимума.

Существенным препятствием для применения изложенных выше методов является требование квазивыпуклости минимизируемой функции. Во многих задачах это предположение не выполняется или не может быть легко проверено. Одним из путей преодоления этой трудности, в случае большого интервала неопределенности, является разделение его на маленькие интервалы, нахождение минимума на каждом из подинтервалов и последующий выбор наименьшего значения из минимумов на подинтервалах.

Из других методов, не требующих квазивыпуклости, отметим следующие численные методы решения нелинейных уравнений: методы Ньютона, простых итераций, позволяющие определить стационарные точки, как решение нелинейного уравнения $f'(x)=0$ [10]. При этом требуется дополнительное исследование функции в окрестности найденных корней на предмет наличия минимума (максимума).

1.1.9 Вопросы для самопроверки

1) Какие алгоритмы решения оптимизационных задач называют пассивными и последовательными.

2) Приведите определения понятий сходимость, скорость сходимости, условия останова. Какие бывают виды сходимости?

3) Приведите определение квазивыпуклой функции, какими свойствами обладают квазивыпуклые функции.

4) Опишите метод удвоения шага оптимизации функции одной переменной, приведите алгоритм, охарактеризуйте особенности его реализации в ПО.

5) Опишите метод равномерного поиска оптимизации функции одной переменной, приведите алгоритм, охарактеризуйте особенности его реализации в

ПО.

6) Опишите метод дихотомии оптимизации функции одной переменной, приведите алгоритм, охарактеризуйте особенности его реализации в ПО.

7)

8) Опишите метод золотого сечения оптимизации функции одной переменной, приведите алгоритм, охарактеризуйте особенности его реализации в ПО.

9) Опишите метод квадратичной интерполяции оптимизации функции одной переменной, приведите алгоритм, охарактеризуйте особенности его реализации в ПО.

10) Опишите метод Ньютона оптимизации функции одной переменной, приведите алгоритм, охарактеризуйте особенности его реализации в ПО.

1.1.10 Задания для практических (лабораторных) работ

Для функций, заданных в таблице 1.2, разработать программное средство и решить задачу минимизации указанными методами.

Задание 1.1 Применить метод удвоения шага для отделения отрезка неопределенности, $h=0,1$.

Задание 1.2 Для интервала неопределенности найденного на первом шаге использовать один из следующих методов (в скобках указан вариант) для уточнения отрезка неопределенности с точностью ε :

- 1) дихотомии (четные варианты);
- 2) золотого сечения (нечетные варианты).

Задание 1.3 Для интервала неопределенности найденного на первом шаге уточнить минимум методом квадратичной интерполяции с точностью ε .

По каждой итерации вывести на печать для первого и второго задания концы отрезков неопределенности, для третьего задания значения аргумента и функции.

Перед выполнением задания необходимо протабулировать функцию на заданном отрезке, начертить график функции $f(x)$.

Таблица 1.2 – Варианты заданий

| № | $f(x)$ | a | b | ϵ |
|----|-------------------------------------|-----|-----|------------|
| 1 | $(x^3 + 1,2x^2 - 1,95x)e^x$ | -1 | 1 | 10^{-3} |
| 2 | $-(2x^3 + 5,1x^2 + 3,6x)e^{-x}$ | -2 | 10 | 10^{-2} |
| 3 | $x \sin x + 2 \sin(x - 0,1)$ | 2 | 8 | 10^{-2} |
| 4 | $3 \sin x + 2 \sin(x - 0,5)$ | 2 | 8 | 10^{-2} |
| 5 | $10 \cos x \cdot \ln(x + 1)$ | 7 | 12 | 10^{-3} |
| 6 | $\ln(1 + x) \cdot (x^2 - 4)(x - 5)$ | 1 | 8 | 10^{-2} |
| 7 | $10 \sin x \cdot \ln x$ | 2 | 8 | 10^{-3} |
| 8 | $l^{1,5x} \sin x$ | 2 | 8 | 10^{-2} |
| 9 | $l^{1,5} \cos x$ | 2 | 6 | 10^{-2} |
| 10 | $(2x^2 - x) \sin x$ | 3 | 8 | 10^{-2} |
| 11 | $5(x^2 - x) \sin x$ | 3 | 8 | 10^{-3} |
| 12 | $10(x^2 - 4) \sin x$ | 3 | 8 | 10^{-3} |
| 13 | $(x^2 - 16) \sin x$ | 4 | 8 | 10^{-3} |
| 14 | $10x^2 \cos x$ | 1 | 6,5 | 10^{-3} |
| 15 | $(4x^2 + 8x) \cos x$ | 1 | 6,5 | 10^{-2} |
| 16 | $(8x^2 + 4x) \cos x$ | 1 | 6,5 | 10^{-3} |
| 17 | $(x^2 - 16) \cos x$ | 3 | 6,5 | 10^{-3} |
| 18 | $(-2x^2 + 12,5) \cos x$ | 4 | 8 | 10^{-2} |
| 19 | $(4x^2 - 16x + 10) \sin x$ | 3 | 8 | 10^{-2} |
| 20 | $(4x^2 - 4x - 9) \sin x$ | 3 | 8 | 10^{-2} |
| 21 | $10x(2 - x)(x - 5) \sin 2x$ | 6 | 8 | 10^{-3} |
| 22 | $10 \ln x \sin 2x$ | 4 | 7 | 10^{-3} |
| 23 | $(x - 1,5)(x - 2,5)(x - 4,5) \ln x$ | 0,5 | 5 | 10^{-2} |
| 24 | $(x - 1,5)(x - 3)(x - 4)e^x$ | 2 | 5 | 10^{-3} |
| 25 | $(10 \sin x) / x$ | 1 | 7 | 10^{-3} |

1.1.11 Задания для самостоятельной проработки и подготовке к контрольным работам

1. Задана целевая функция $f(x) = 3(x - 1)^3 - 2(x + 1)^2 + 1 \rightarrow \min$ отделить отрезок неопределенности, используя метод равномерного поиска.

2. Задана целевая функция $f(x) = 3(x - 1)^3 - 2(x + 1)^2 + 1 \rightarrow \min$ продемонстрировать один шаг к точке минимума методом золотого сечения, отрезок неопределенности отделить графически, используя пакет MathCad/MS Excel.

3. Задана целевая функция $f(x) = 3(x - 1)^3 - 2(x + 1)^2 + 1 \rightarrow \min$ отделить отрезок неопределенности, используя метод удвоения шага.

4. Задана целевая функция $f(x) = 3(x - 2)^3 - (4x + 3)^2 \rightarrow \min$ сделать один шаг к точке минимума методом дихотомии, отрезок неопределенности отделить графически, используя пакет MathCad/MS Excel.

5. Задана целевая функция $f(x) = 3(x - 2)^3 - (4x + 3)^2 \rightarrow \min$ сделать один шаг к точке минимума методом квадратичной аппроксимации, отрезок неопределенности отделить графически, используя пакет MathCad/MS Excel.

6. Нарисовать схему алгоритма метода равномерного поиска.

7. Нарисовать схему алгоритма метода половинного деления.

8. Нарисовать схему алгоритма метода золотого сечения.

9. Нарисовать схему алгоритма метода квадратичной аппроксимации.

10. Нарисовать схему алгоритма метода удвоения шага.

1.2 Численные методы безусловной минимизации функции многих переменных

Материал раздела выполнен в соответствии с источниками: 3,5, 6, 7, 11, 28, 31, 32, 37, 38.

1.2.1 Релаксационная последовательность. Сходимость. Критерии останова

Поставим задачу нахождения минимума функции многих переменных:

$$f(x) \rightarrow \min, x \in D \subset \mathbb{R}^n. \quad (1.5)$$

Пусть функция $f(x)$ ограничена снизу. Будем считать, что $x^* = \arg \min f(x)$ существует. В общем случае такая точка может быть не единственная.

Будем говорить, что с точностью до ε точка x может быть взята в качестве приближенного значения точки минимума x^* , если $\|x - x^*\| < \varepsilon$, где

$$\|x - x^*\| = \sqrt{\sum_{i=1}^n (x_i - x_i^*)^2} \text{ или } \|x - x^*\| = \max_{i=1, n} |x_i - x_i^*|.$$

Общей чертой всех методов решения задачи (1.5) является последовательный переход от точки $x^{(k-1)}$ к точке $x^{(k)}$, $k \in \mathbb{N}$, начиная с некоторой начальной точки $x^{(0)} \in \mathbb{R}^n$, причем на каждой итерации с номером k выполняется условие

$$\begin{aligned} f(x^{(k)}) &\leq f(x^{(k-1)}), \quad k \in \mathbb{N}, \\ f(x^{(k)}) &= f^{(k)}. \end{aligned} \quad (1.6)$$

УТВЕРЖДЕНИЕ 1.1. Так как целевая функция ограничена снизу, то в силу признака Вейерштрасса невозрастающая последовательность $\{f^{(k)}\}$ сходится к некоторому пределу.

ЗАМЕЧАНИЕ. Однако из этого утверждения в общем случае еще не следует, что итерационная последовательность $\{x^{(k)}\}$ точек $x^{(k)} \in \mathbb{R}^n$, соответствующих значениям $f^{(k)}$, сходится, а если она и сходится, то ее пределом является точка $x^* \in \mathbb{R}^n$ минимума функции $f(x)$, удовлетворяющая (1.5).

ОПРЕДЕЛЕНИЕ. Последовательность $\{x^{(k)}\}$, удовлетворяющая условию (1.6), называется релаксационной.

Численные методы, применяемые для таких последовательностей, относятся к методам спуска. При анализе сходимости релаксационных последовательностей удобно рассматривать невозрастающую последовательность $\{\varphi_k\}$, где $\varphi_k = f^{(k)} - f^* > 0$, где $f^* = f(x^*)$ (при $\varphi_k = 0$ принимаем $x^* = x^k$). Для оценки сходимости этой последовательности используют следующие утверждения.

ЛЕММА 1.1 [4]

Если для элементов последовательности $\{\varphi_k\}$ выполнены условия:

$$\varphi_{k-1} - \varphi_k \geq \gamma_k \cdot \varphi_{k-1}^2, \quad \varphi_{k-1} > 0, \quad \gamma_k > 0, \quad k \in \mathbb{N}, \quad (1.7)$$

то справедлива оценка:

$$\varphi_m \leq \frac{\varphi_0}{1 + \varphi_0 \cdot \sum_{k=1}^m \gamma_k}. \quad (1.8)$$

Доказательство:

Рассмотрим разность $\varphi_{k-1} - \varphi_k \geq \gamma_k \cdot \varphi_{k-1}^2$. Поделим на $\varphi_{k-1} \cdot \varphi_k$, тогда

получаем $\frac{1}{\varphi_k} - \frac{1}{\varphi_{k-1}} \geq \gamma_k \cdot \frac{\varphi_{k-1}}{\varphi_k}$.

Суммируя это неравенство, получаем
$$\sum_{k=1}^m \left(\frac{1}{\varphi_k} - \frac{1}{\varphi_{k-1}} \right) = \frac{1}{\varphi_m} - \frac{1}{\varphi_0} \geq \sum_{k=1}^m \gamma_k,$$

откуда следует (1.8).

УТВЕРЖДЕНИЕ 1.2 [4]. Пусть функция $f(x)$ является выпуклой, тогда:

$$\forall k \in \mathbb{N}: \quad (1.9)$$

$$0 < \varphi_{k-1} = f^{(k-1)} - f^* \leq \langle \text{grad}f(x^{(k-1)}), x^{(k-1)} - x^* \rangle \leq \|\text{grad}f(x^{(k-1)})\| \cdot \|x^{(k-1)} - x^*\|$$

ТЕОРЕМА 1.2 [4]

Пусть функция $f(x)$ выпукла и дифференцируема на множестве \mathbb{R}^n , а последовательность $\{x^{(k)}\}$ является релаксационной. Если

$$\gamma_k = \frac{f^{(k-1)} - f^{(k)}}{\|\text{grad}f(x^{(k-1)})\|^2 \cdot \|x^{(k-1)} - x^*\|^2}, \quad (1.10)$$

то справедлива оценка (1.8).

Чтобы использовать теорему 1.2, необходимо располагать оценками для значений γ_k , определяемыми неравенствами (1.9). В общем случае получить такие оценки до проведения численного решения задачи достаточно сложно, а в процессе решения разность $(x^{(k-1)} - x^*)$ неизвестна. Однако для некоторых численных методов безусловной минимизации удастся установить, что $\gamma_k \geq \text{const} > 0$ начиная с некоторого номера k . Тогда в соответствии с выражением (1.8) величина φ_m имеет порядок малости $\frac{1}{m}$ при $m \rightarrow \infty$.

Если рассмотреть дифференцируемую функцию, то можно получить оценку вида:

$$\varphi_m = f^{(m)} - f^* \leq \frac{\eta^2}{\sum_{k=1}^m \frac{f^{(k-1)} - f^{(k)}}{\|\text{grad}f(x^{(k-1)})\|^2}}, \quad m \geq m', \quad (1.11)$$

$$\|x^{(k-1)} - x^*\| \leq \eta, \quad \eta = \text{diam}X_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x^{(0)})\}. \quad (1.12)$$

Оценку (1.12) легко вычислять в процессе численного решения задачи минимизации.

ЛЕММА 1.2 [4]

Если для элементов последовательности $\{\varphi_k\}$ выполнено условие

$$\varphi_{k-1} - \varphi_k \geq \tau_k \cdot \varphi_{k-1}^2, \quad \varphi_{k-1} > 0, \quad \tau_k \geq 0, \quad k \in \mathbb{N}, \quad (1.13)$$

то справедлива оценка

$$\varphi_m \leq \varphi_0 \cdot \exp\left(-\sum_{k=1}^m \tau_k\right). \quad (1.14)$$

ТЕОРЕМА 1.3 [4]

Пусть функция $f(x)$ дифференцируема и выпукла, тогда если

$$\tau_k = \frac{f^{(k-1)} - f^{(k)}}{\|\text{grad}f(x^{(k-1)})\|^2}, \quad k \in \mathbb{N}, \quad \text{то справедлива оценка (1.14).}$$

ТЕОРЕМА 1.4 [4]

Для сильно выпуклой, дифференцируемой функции $f(x)$ при

$$\tau_k = \gamma \frac{f^{(k-1)} - f^{(k)}}{\|\text{grad}f(x^{(k-1)})\|^2}, \quad \text{где } \gamma \text{ — параметр сильной выпуклости функции } f(x),$$

справедливы оценки (1.14) и

$$\|x^m - x^*\|^2 \leq \frac{2}{\gamma} \varphi_m. \quad (1.15)$$

На практике чаще всего используют следующие критерии останова:

$$\frac{f^{(k-1)} - f^{(k)}}{\|\text{grad}f(x^{(k-1)})\|^2} < \varepsilon, \quad (1.16)$$

$$\|x^{(k)} - x^{(k-1)}\| < \varepsilon, \quad (1.17)$$

$$\|f(x^{(k)}) - f(x^{(k-1)})\| < \varepsilon. \quad (1.18)$$

Необходимым условием, что x^* – точка минимума, является равенство $\text{grad}f(x^*) = 0$. Поэтому можно воспользоваться еще одним условием прекращения итераций:

$$\|\text{grad}f(x^{(k-1)})\| < \varepsilon. \quad (1.19)$$

Если при проведении итераций, к значению $f(x^*)$ сходится последовательность $\{f^{(k)}\}$, то говорят о слабой сходимости. Если к точке x^* сходится $x^{(k)}$, то говорят о сильной сходимости.

ЗАМЕЧАНИЕ Для геометрической иллюстрации методов будем использовать функцию двух переменных. Напомним, что линиями уровня функции $Z = f(x_1, x_2)$ называют множество точек (x_1, x_2) удовлетворяющих уравнению $f(x_1, x_2) = C = \text{const}$. Меняя C , мы получаем различные линии уровня функции $f(x_1, x_2)$. Геометрически линия уровня – это проекция на плоскость X_1OX_2 линии пересечения $Z = f(x_1, x_2)$ и плоскости $Z = C$ (рисунок 1.12). Имея множество линий уровня, мы получаем представление о поведении функции $f(x_1, x_2)$, говорят о рельефе функции.

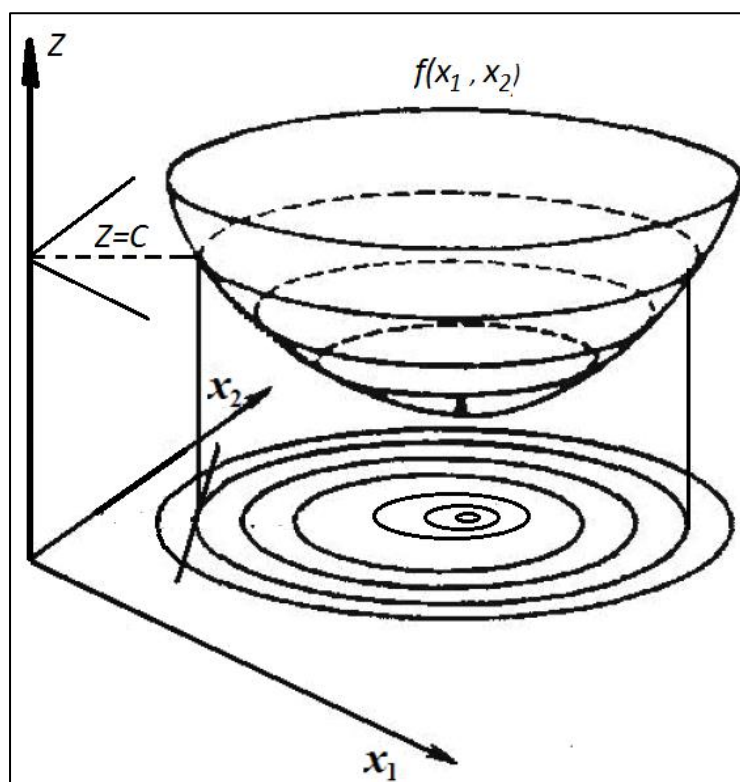


Рисунок 1.12 – Линии уровня целевой функции

1.2.2 Методы спуска. Принципы выбора длины шага в методах спуска

Пусть функция $f(x) \rightarrow \min$, где $x \in D \subset \mathbb{R}^n$. Пусть существует x^* , в которой функция $f(x)$ принимает минимальное значение. Процедуру поиска в этой точке можно описать рекуррентным соотношением:

$$x^{(k)} = x^{(k-1)} + \alpha_k \cdot h^{(k)}, k \in \mathbb{N}, \quad (1.20)$$

где $h^{(k)}$ – вектор, определяющий направление спуска на k -ом шаге поиска;

α_k – длина шага спуска.

Методы спуска различают способами выбора $h^{(k)}$ и α_k . Возникают следующие вопросы.

1. Как выбрать α_k ?

2. Как целесообразнее выбирать $h^{(k)}$?
3. Сходится ли последовательность $x^{(0)}, x^{(1)}, \dots$?
4. Как оценить погрешность?

Ответим на первый вопрос. Пусть на k -ой итерации выбран вектор $h^{(k)}$. Тогда один из способов выбора значения α_k базируется на требовании, чтобы выполнялось неравенство

$$f(x^{(k-1)} + \alpha_k \cdot h^{(k)}) \leq (1 - \lambda_k) \cdot f(x^{(k-1)}) + \lambda_k \cdot \min_{\alpha \in \mathbb{R}} f(x^{(k-1)} + \alpha \cdot h^{(k)}), \quad (1.21)$$

где $\lambda_k \in [0,1]$ – некоторый параметр.

Выбор α_k по формуле (1.21) гарантирует выполнение условия (1.6) ($f_k \leq f_{k-1}$), т.е. последовательность $\{x^{(k)}\}$ в этом случае является релаксационной.

Пусть $\lambda_k = 1$, тогда неравенство (1.21) переходит в равенство: $f(x^{(k-1)} + \alpha_k \cdot h^{(k)}) = \min_{\alpha \in \mathbb{R}} f(x^{(k-1)} + \alpha \cdot h^{(k)})$, а значение α_k соответствует минимальному значению функции одной переменной $f(x^{(k-1)} + \alpha \cdot h^{(k)})$:

$$\alpha_k = \arg \min_{\alpha > 0} f(x^{(k-1)} + \alpha \cdot h^{(k)}), \quad \{x \in \mathbb{R}^n : x = x^{(k-1)} + \alpha \cdot h^{(k)}; \alpha \in \mathbb{R}\}.$$

В этом случае для нахождения α используют методы одномерной оптимизации, представленные ранее, а α может быть не единственно. Процедура поиска минимума функции вдоль заданного направления спуска $h^{(k)}$ может быть представлена следующим образом.

Процедура поиска вдоль заданного направления

Пусть на k -ом шаге поиска точки минимума функции многих переменных требуются решить задачу оптимизации функции вдоль заданного направления $h^{(k)}$:

$$f(x^{(k)}) = \min_{\alpha \in \mathbb{R}} f(x^{(k-1)} + \alpha \cdot h^{(k)}) \quad (1.22)$$

Для локализации отрезка, содержащего $x^{(k)}$ используют следующий алгоритм.

Алгоритм:

Шаг 0: $\alpha^{(0)} = 0$, $h^{(k)}$ – заданное направление спуска, $x^{(k)}$ – очередное приближение к точке минимума, $d \in \mathbb{R}$ – начальный шаг, $r=1$, l – ограничение на требуемую на выходе алгоритма длину отрезка неопределенности;

Шаг 1:

Шаг 1.1: Если $f(x^{(k-1)} + (\alpha^{(0)} + d)h^{(k)}) < f(x^{(k-1)} + \alpha^{(0)} \cdot h^{(k)})$, тогда $\alpha^{(1)} = \alpha^{(0)} + d$, идти к Шагу 2 иначе к шагу 1.2.

Шаг 1.2: Если $f(x^{(k-1)} + (\alpha^{(0)} + d)h^{(k)}) \geq f(x^{(k-1)} + \alpha^{(0)} \cdot h^{(k)})$, тогда

Шаг 1.2.1 $d = -d$;

Шаг 1.2.2 Если $f(x^{(k-1)} + (\alpha^{(0)} + d)h^{(k)}) \geq f(x^{(k-1)} + \alpha^{(0)} \cdot h^{(k)})$, тогда $d = d/2$, идти на шаг 1.1, иначе $\alpha^{(1)} = \alpha^{(0)} + d$, идти к шагу 2;

Шаг 2: $d = 2 \cdot d$, $r = r + 1$;

Шаг 3: Если $f(x^{(k-1)} + (\alpha^{(r)} + d)h^{(k)}) < f(x^{(k-1)} + \alpha^{(r)} \cdot h^{(k)})$, то $\alpha^{(r+1)} = \alpha^{(r)} + d$, идти к шагу 2;

Иначе Если $f(x^{(k-1)} + (\alpha^{(r)} + d)h^{(k)}) \geq f(x^{(k-1)} + \alpha^{(r)} \cdot h^{(k)})$, тогда переход к шагу 4;

Шаг 4: Если длина отрезка неопределенности

$$\| (x^{(k-1)} + \alpha^{(r-1)}h^{(k)}) - (x^{(k-1)} + \alpha^{(r+1)}h^{(k)}) \| = \| (\alpha^{(r-1)} - \alpha^{(r+1)})h^{(k)} \| < l \quad \text{меньше}$$

требуемой l , тогда точка минимума в заданном направлении принадлежит отрезку $x^{(k)} \in [x^{(k-1)} + \alpha^{(r-1)}h^{(k)}, x^{(k-1)} + \alpha^{(r+1)}h^{(k)}]$;

Иначе процедуру повторяют с шага 1, причем в качестве $\alpha^{(0)}$ выбирают $\alpha^{(r-1)}$.

Конец алгоритма.

Для сужения отрезка локализации и определения $x^{(k)}$ с заданной точностью можно воспользоваться, к примеру, методами дихотомии, золотого сечения, квадратичной аппроксимации.

Приведем алгоритм уточнения на примере метода золотого сечения.

Алгоритм:

Шаг 0: Пусть в результате отделения отрезка неопределенности получен отрезок $x^{(k)} \in [x^{(k-1)} + \alpha^{(r-1)}h^{(k)}, x^{(k-1)} + \alpha^{(r+1)}h^{(k)}]$, тогда положим $a = \alpha^{(r-1)}; b = \alpha^{(r+1)}$, ε – необходимая точность определения минимума функции (1.18);

Шаг 1: Находим $z^{(1)} = a + (1 - \tau)(b - a), z^{(2)} = a + \tau(b - a)$, где $\tau = 0,618$;

Шаг 2: Вычисляем $f(x^{(k-1)} + z^{(1)}h^{(k)})$ и $f(x^{(k-1)} + z^{(2)}h^{(k)})$;

Шаг 3: Если $f(x^{(k-1)} + z^{(1)}h^{(k)}) < f(x^{(k-1)} + z^{(2)}h^{(k)})$;

тогда Шаг 3.1 $b = z^{(2)}, z^{(2)} = z^{(1)}, z^{(1)} = a + (1 - \tau)(b - a)$, идти на шаг 4;

иначе Шаг 3.2 Если $f(x^{(k-1)} + z^{(1)}h^{(k)}) \geq f(x^{(k-1)} + z^{(2)}h^{(k)})$

тогда $a = z^{(1)}, z^{(1)} = z^{(2)}, z^{(2)} = a + \tau(b - a)$, идти на шаг 4;

Шаг 4: Если $\|(x^{(k-1)} + b \cdot h^{(k)}) - (x^{(k-1)} + a \cdot h^{(k)})\| = \|(b - a)h^{(k)}\| < \varepsilon$, тогда переход к шагу 5; иначе переход к шагу 2;

Шаг 5: $\alpha_k = \frac{a + b}{2}, x^{(k)} = x^{(k-1)} + \alpha_k \cdot h^{(k)}$.

Конец алгоритма.

1) Пусть $\lambda_k = 0$, тогда получаем, что

$$f(x^{(k-1)} + \alpha_k \cdot h^{(k)}) \leq f(x^{(k-1)}). \quad (1.23)$$

В этом случае отпадает необходимость одномерной оптимизации функции и α_k выбирается эвристически (т.е. подбором), например, одним из следующих методов.

Метод расходящегося ряда

Используется априорный выбор коэффициентов. В этом случае α_k выбирается до начала вычислений, исходя из следующих соображений:

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

Этим условиям удовлетворяет, например:

$$\alpha_k = \frac{c}{k}, c = \text{const}, k = 1, 2, 3, \dots$$

Условие сходимости ряда $\sum_{k=1}^{\infty} \alpha_k^2$ накладывают для того, чтобы добиться достаточно быстрой сходимости последовательности α_k к 0 в окрестности точки минимума.

Метод дробления шага

Алгоритм:

Шаг 0. Выбирают некоторую константу $\beta > 0$ (начальный шаг поиска) и $0 < \lambda < 1$ (коэффициент дробления);

Шаг 1. Пусть $\alpha := \beta$, проверяется условие релаксации $f(x^{(k-1)} + \alpha h^{(k)}) < f(x^{(k)})$;

Шаг 2. Если неравенство $f(x^{(k-1)} + \alpha h^{(k)}) < f(x^{(k)})$ выполняется, тогда конец алгоритма $\alpha_k := \beta$, иначе шаг 3;

Шаг 3. Дробим шаг: $\alpha := \lambda \alpha$;

Шаг 4. Если $\alpha < \varepsilon$ или выполнено $f(x^{(k-1)} + \alpha h^{(k)}) < f(x^{(k)})$, тогда конец алгоритма $\alpha_k := \alpha$ иначе шаг 3.

Конец алгоритма.

ЗАМЕЧАНИЕ. Если $f(x^{(k-1)} + \alpha h^{(k)}) < f(x^{(k)})$ окажется верным, при $\alpha > \beta$, то шаг можно увеличивать до тех пор пока выполняется условие $f(x^{(k-1)} + \alpha h^{(k)}) < f(x^{(k)})$ и принять $\alpha := \mu \cdot \alpha$ ($\mu > 1$). Последний α , при котором $f(x^{(k-1)} + \alpha h^{(k)}) < f(x^{(k)})$ выполняется, берут за окончательный размер шага α_k .

2) Пусть $\lambda_k \in (0,1)$, тогда из неравенства (1.21) получаем следующее выражение

$$f(x^{(k-1)}) - f(x^{(k)}) \geq \lambda_k \cdot (f(x^{(k-1)}) - \min_{\alpha \in \mathbb{R}} f(x^{(k-1)} + \alpha \cdot h^{(k)}))$$

$$\lambda_k \cdot (f(x^{(k-1)}) - \min_{\alpha \in \mathbb{R}} f(x^{(k-1)} + \alpha \cdot h^{(k)})) \geq \lambda_k \cdot (f(x^{(k-1)}) - f(x^{(k-1)} + \alpha \cdot h^{(k)}))$$

и оценка сходимости примет вид:

$$\frac{f(x^{(m)}) - f(x^*)}{f(x^{(0)}) - f(x^*)} \leq \exp \left(\gamma \cdot \sum_{k=1}^m \lambda_k \cdot \frac{f(x^{(k-1)}) - f(x^{(k-1)} + \alpha h^{(k)})}{\|\text{grad}f(x^{(k-1)})\|^2} \right),$$

где γ – параметр сильной выпуклости функции.

Это позволяет установить роль λ_k , в частности пока отношение:

$$\Psi_k = \frac{f(x^{(k-1)}) - f(x^{(k)})}{\|\text{grad}f(x^{(k-1)})\|^2}, \quad (1.24)$$

остается достаточно большим, значение λ_k стоит выбирать так, чтобы $\lambda_k \cdot f^{(k)}$ оставалось на каждой итерации примерно постоянным, т.е. при сравнительно малом λ_k нет необходимости в высокой точности вычисления λ_k и решении задачи одномерной оптимизации $f(x^{(k-1)} + \alpha \cdot h^{(k)})$ по аргументу α и можно выбрать α из

условия релаксации (1.23) эвристически. Однако по мере приближения к точке x^* минимума целевой функции необходимо увеличивать значение λ_k вплоть до единицы и выбирать значение α_k из более точного решения задачи одномерной оптимизации функции $\alpha_k = \arg \min_{\alpha} f(x^{(k-1)} + \alpha \cdot h^{(k)})$ по аргументу α .

1.2.3 Методы нулевого порядка. Метод циклического покоординатного спуска

Опишем алгоритм одного из методов прямого поиска нулевого порядка – метод циклического покоординатного спуска, выбирая в качестве направлений поиска координатные векторы $h_1 = e_1, h_2 = e_2, \dots, h_n = e_n$, т.е. $h_i = e_i$ – вектор, все компоненты которого, за исключением i равны нулю.

Тогда $x^{(0)} + \alpha e_i = (x_1^{(0)}, x_2^{(0)}, \dots, x_{i-1}^{(0)}, x_i^{(0)} + \alpha, x_{i+1}^{(0)}, \dots, x_n^{(0)})$, $f(x^{(0)} + \alpha e_i) = f(x_1^{(0)}, x_2^{(0)}, \dots, x_{i-1}^{(0)}, x_i^{(0)} + \alpha, x_{i+1}^{(0)}, \dots, x_n^{(0)})$ и, следовательно, при минимизации функции $f(x^{(0)} + \alpha e_i)$, речь идет о минимизации функции по одной переменной $x_i = x_i^{(0)} + \alpha_i$, при фиксированных остальных переменных.

Выберем начальную точку $x^{(0)}$, направление $h_1 = e_1$ и, минимизируя функцию $f(x^{(0)} + \alpha e_1)$ по α вдоль заданного направления $h_1 = e_1$ (см. раздел 1.2.2), найдем, что минимум этой функции достигается при α_1 , в точке $x^{(0)} + \alpha_1 e_1 = (x_1^{(0)} + \alpha_1, x_2^{(0)}, \dots, x_n^{(0)})$. Выберем направление e_2 и, минимизируя $f(x^{(0)} + \alpha e_2)$ по α найдем, что минимум этой функции достигается при α_2 в точке $x^{(0)} + \alpha_2 e_2 = (x_1^{(0)}, x_2^{(0)} + \alpha_2, x_3^{(0)}, \dots, x_n^{(0)})$. После n шагов найдем $x^{(0)} + \alpha_n e_n = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)} + \alpha_n)$.

Положим $x^{(1)} = (x_1^{(0)} + \alpha_1, x_2^{(0)} + \alpha_2, \dots, x_n^{(0)} + \alpha_n)$ и тем самым завершим один

цикл покоординатного спуска. Таким образом, на каждом шаге проводят одномерную минимизацию целевой функции последовательно по каждой из координат. Поиск точки x^* можно записать рекуррентным соотношением:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \cdot e^{(k)}, k \in \mathbb{N},$$

где $x^{(0)}$ – начальная точка,

$e^{(k)}$ – единичный вектор, определяющий направление спуска на k -м шаге,

$\alpha^{(k)}$ – вектор, задающий шаг спуска по каждой координате.

Пусть e_i – стандартный базис пространства \mathbb{R}^n . Тогда $x_j^{(k+1)} = x_j^{(k)} + \alpha_j^{(k)}$, где $\alpha_j^{(k)} \in \mathbb{R}$ находят из решения задачи одномерной минимизации $\varphi_j^k(\alpha_j) \rightarrow \min$ и $\varphi_j^k(\alpha_j) = f(x^{(k)} + \alpha_j^{(k)} \cdot e_j)$. Индекс j изменяется циклически, пробегая на каждой шаге покоординатного спуска все значения от 1 до n , причем значения $\alpha_j^{(k)}$ могут быть как положительными, так и отрицательными.

В качестве критерия останова используют

$$\|x^{(k+1)} - x^{(k)}\| < \varepsilon \text{ и } |f(x^{(k)}) - f(x^{(k+1)})| < \varepsilon,$$

где ε – требуемая точность,

$$\|x^{(k+1)} - x^{(k)}\| = \sqrt{\sum_{i=1}^n (x_i^{(k+1)} - x_i^{(k)})^2} \text{ или } \|x^{(k+1)} - x^{(k)}\| = \max |x_i^{(k+1)} - x_i^{(k)}| \quad (i = \overline{1, n}).$$

Алгоритм метода циклического покоординатного спуска:

Шаг 1. $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$, $k = 0, \varepsilon$;

Шаг 2. $\alpha_1^{(k)} = \arg \min_{\alpha} f(x^{(k)} + \alpha \cdot e_1)$, $x_1^{(k+1)} = x_1^{(k)} + \alpha_1^{(k)}$;

$$\alpha_2^{(k)} = \arg \min_{\alpha} f(x^{(k)} + \alpha \cdot e_2), x_2^{(k+1)} = x_2^{(k)} + \alpha_2^{(k)};$$

.....

$$\alpha_n^{(k)} = \arg \min_{\alpha} f(x^{(k)} + \alpha \cdot e_n), x_n^{(k+1)} = x_n^{(k)} + \alpha_n^{(k)};$$

Шаг 3. Если $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$, тогда $x^* = x^{(k+1)}$, иначе $k = k + 1$ и шаг 2;

Конец алгоритма.

Этот метод прост в реализации, но эффективен, когда минимизируемая функция является сепарабельной, т.е. представляет собой сумму функций, каждая из которых зависит лишь от одной координаты:

$$f(x) = \sum_{j=1}^n h_j(x), x \in \mathbb{R}^n.$$

В этом случае решение можно получить за один шаг. Для функции более общего вида может нарушаться даже условие релаксации последовательности $f(x^{(k-1)}) - f(x^{(k)}) > 0$. Поэтому обычно используют модификацию этого метода, где на каждом шаге учитывают результаты метода для определенных ранее координат. В частности, выбирают начальную точку $y_1 = x^{(0)}$, направление $h_1 = e_1$ и минимизируют функцию $f(y_1 + \alpha e_1)$ по α вдоль заданного направления $h_1 = e_1$. Пусть минимум этой функции достигается при α_1 , в точке $y_2 = y_1 + \alpha_1 e_1 = (x_1^{(0)} + \alpha_1, x_2^{(0)}, \dots, x_n^{(0)})$. Полагают $y_2 = y_1 + \alpha_1 e_1$, выбирают направление e_2 , далее минимизируют $f(y_2 + \alpha e_2)$ по α . Пусть минимум этой функции достигается при α_2 в точке $y_3 = y_2 + \alpha_2 e_2 = (x_1^{(0)} + \alpha_1, x_2^{(0)} + \alpha_2, x_3^{(0)}, \dots, x_n^{(0)})$. После n шагов найдено $y_{n+1} = y_n + \alpha_n e_n = (x_1^{(0)} + \alpha_1, x_2^{(0)} + \alpha_2, \dots, x_n^{(0)} + \alpha_n)$. Положим $x^{(1)} = y_{n+1}$ и тем самым завершим один цикл по координатного спуска.

Алгоритм модифицированного метода:

Шаг 1. $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$, $k = 0$, ε , $x^{(k+1)} = x^{(k)}$;

Шаг 2. $\alpha_1^{(k)} = \arg \min_{\alpha} f(x^{(k+1)} + \alpha \cdot e_1)$, $x_1^{(k+1)} = x_1^{(k+1)} + \alpha_1^{(k)}$;

$\alpha_2^{(k)} = \arg \min_{\alpha} f(x^{(k+1)} + \alpha \cdot e_2)$, $x_2^{(k+1)} = x_2^{(k+1)} + \alpha_2^{(k)}$;

.....

$\alpha_n^{(k)} = \arg \min_{\alpha} f(x^{(k+1)} + \alpha \cdot e_n)$, $x_n^{(k+1)} = x_n^{(k+1)} + \alpha_n^{(k)}$;

Шаг 3. Если $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$, тогда $x^* = x^{(k+1)}$, иначе $k = k + 1$, $x^{(k+1)} = x^{(k)}$,

идти на шаг 2.

Конец алгоритма.

Проиллюстрируем метод геометрически на примере функции двух переменных (рисунок 1.13).

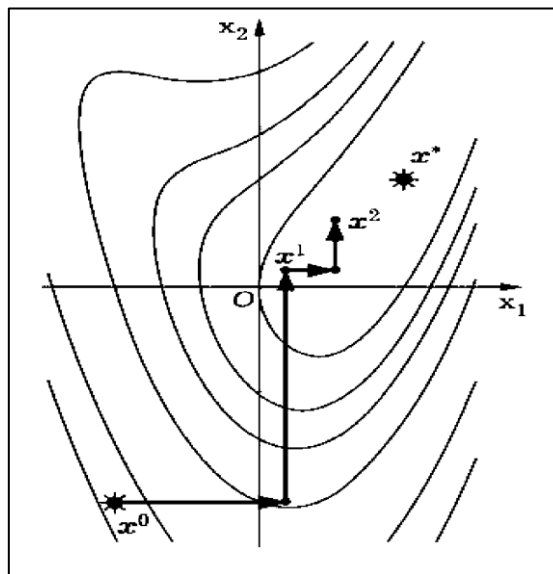


Рисунок 1.13 – Метод циклического покоординатного спуска

Двигаясь от точки $x^{(0)} = y_1$ параллельно оси Ox_1 , мы переходим от линий уровня с большим значением $f(x_1, x_2) = c$ к меньшим. Точка y_2 , в которой мы коснемся некоторой линии уровня, является точкой минимума функции $f(x_1, x_2^{(0)})$ в направлении, параллельном оси Ox_1 . Двигаясь от точки y_2 параллельно оси Ox_2 , мы, переходя от линий уровня с большим "С" к линиям уровня с меньшим "С",

достигнем минимального значения в точке y_3 – точке касания некоторой линии уровня. В точке y_3 завершается цикл по координатного спуска и получаем $x^{(1)} = y_3$.

Отправляясь от $x^{(1)}$, как от начальной точки, можем найти $x^{(2)}$ и т.д. Остается указать условия сходимости последовательности $x^{(0)}, x^{(1)}, \dots$ и указать оценку погрешности.

Для сходимости метода циклического по координатного спуска достаточно следующих требований:

1. Минимум $f(x)$ вдоль любого направления единственен.
2. Последовательность точек $x^{(0)}, x^{(1)}, \dots, x^{(k)}, \dots$ принадлежит некоторому замкнутому ограниченному подмножеству области \bar{D} .

Что касается погрешности, то ее можно оценить по формуле:

$$\|x^* - x^{(k+1)}\| \leq \|x^{(k+1)} - x^{(k)}\|.$$

1.2.4 Методы нулевого порядка. Метод Хука-Дживса

Метод Хука-Дживса так же относят к методам прямого поиска нулевого порядка. Этот метод используется для функции, линия уровня которой сильно вытянута. Эффективность прямого поиска точки минимума ограниченной снизу целевой функции можно повысить, если на каждом k -м шаге поиска соответствующим образом выбирать направление спуска. Для этого на каждом k -м шаге выделяют предварительный этап исследующего поиска. Целью этого этапа является выбор направления спуска путем исследования поведения целевой функции $f(x)$ в окрестности точки $x^{(k-1)}$, найденной на предыдущем шаге. В результате выполнения этого этапа находят точку $\tilde{x}^{(k)}$, такую что $f(\tilde{x}^{(k)}) < f(x^{(k-1)})$. Направление спуска, завершающего k -ый шаг спуска, определяется вектором

$h^k = \tilde{x}^{(k)} - x^{(k-1)}$. Геометрическая интерпретация этапа исследующего поиска приведена на рисунках 1.14, 1.15.

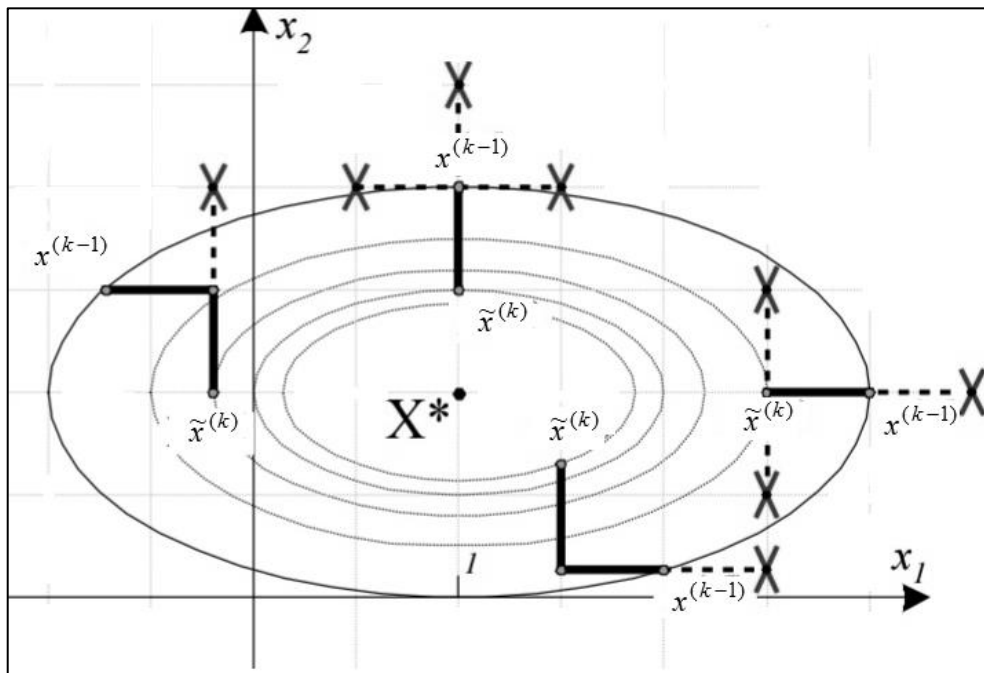


Рисунок 1.14 – Примеры удачного исследующего поиска

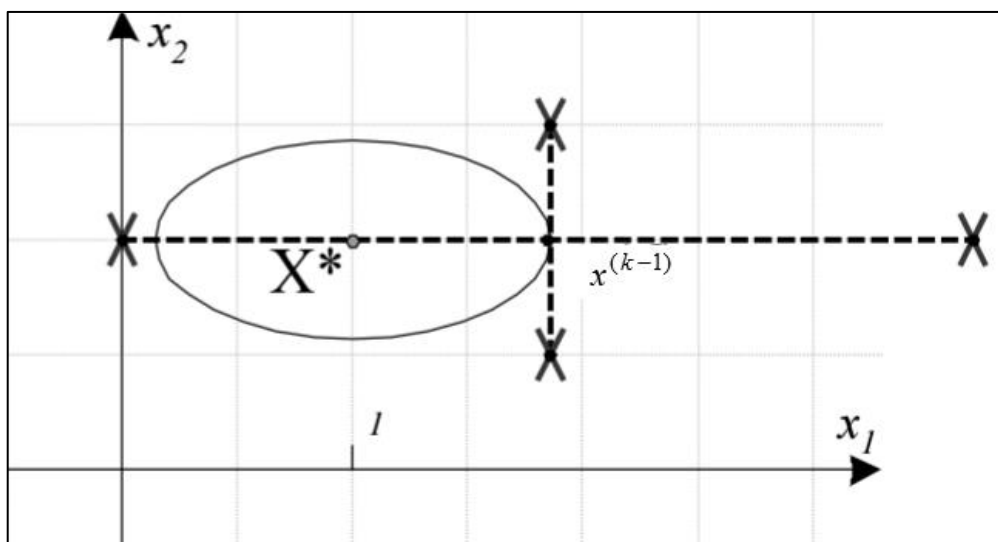


Рисунок 1.15 – Пример неудачного исследующего поиска

Далее следует второй этап – поиск по образцу. На этом этапе по формуле $x^{(k)} = \tilde{x}^{(k)} + \alpha_k h^k$, где $h^k = \tilde{x}^{(k)} - x^{(k-1)}$ подбирают ускоряющий множитель $\alpha_k > 0 : f(x^{(k)}) \leq f(\tilde{x}^{(k+1)})$. Значение α_k можно подобрать из условия минимума

функции $f(x)$ при смещении точки $x^{(k)}$ в направлении этого вектора (см. алгоритм поиска вдоль заданного направления). Геометрическая интерпретация этапа поиска по образцу представлена на рисунках 1.16, 1.17.

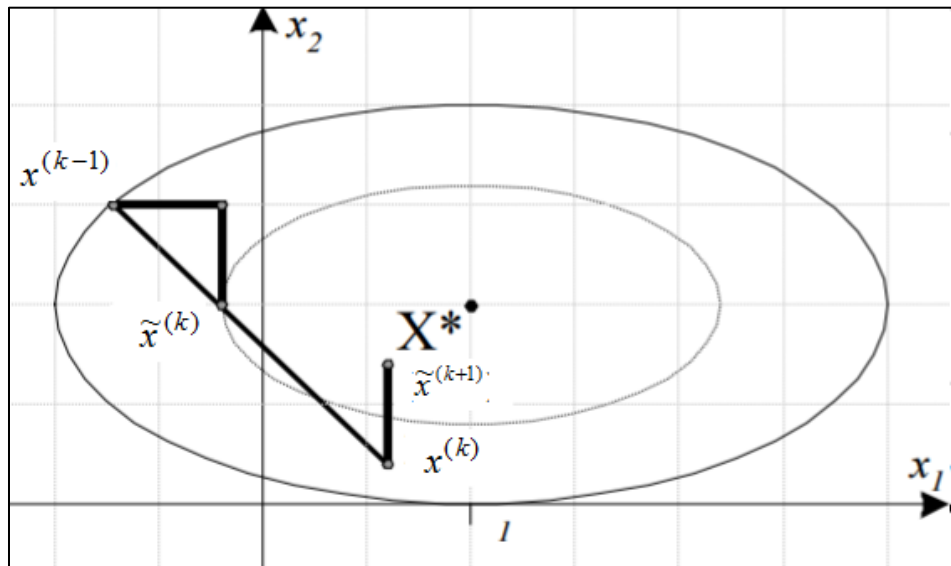


Рисунок 1.16 – Пример удачного поиска по образцу

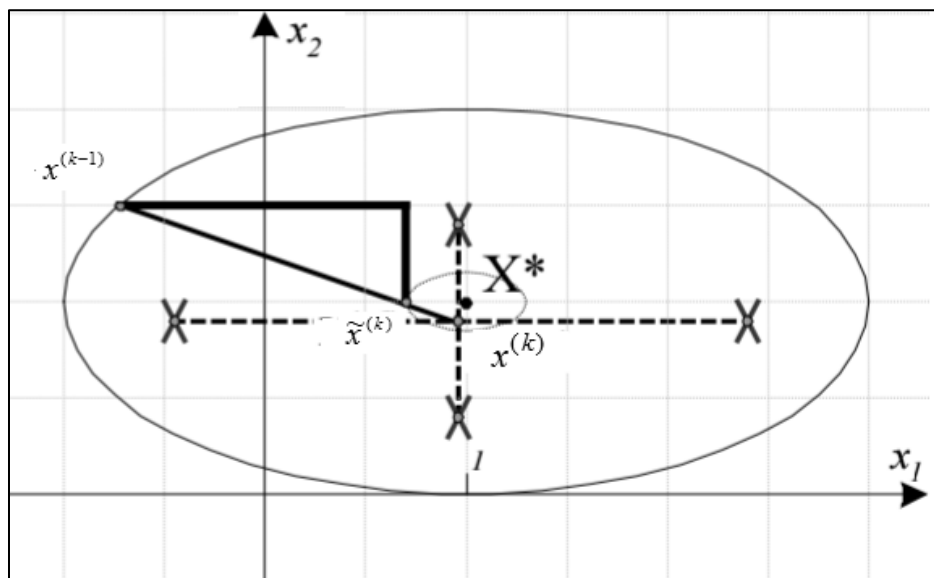


Рисунок 1.17 – Пример неудачного поиска по образцу

Сходимость метода Хука-Дживса обеспечена при тех же условиях, что и метода покоординатного спуска.

На рисунке 1.18 проиллюстрированы несколько итераций метода Хука-

Дживса для функции двух переменных.

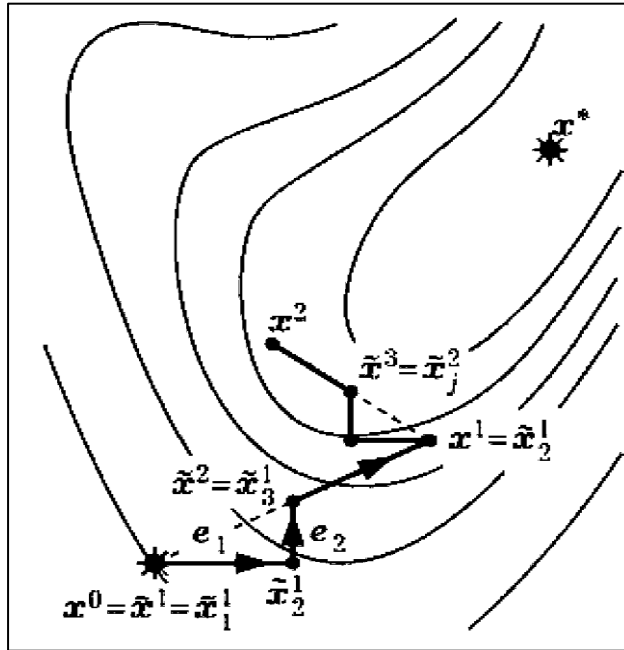


Рисунок 1.18 – Итерации метода Хука-Дживса [31]

Алгоритм:

Этап исследующего поиска

Пусть $x^{(0)}, b = (b_1, b_2, \dots, b_n)^T : \|b\| < \varepsilon$, где $\varepsilon > 0$ – заданный параметр точности исследующего поиска, b – вектор перемещений, координаты b_i определяют приращения координат точки $x^{(k)}$ на этапе исследующего поиска.

Шаг 0. $k = j = 1, \tilde{x}_j^{(k)} = \tilde{x}^{(k)} = x^{(0)}, f_j^{(k)} = f(x^{(0)})$;

Шаг 1. $f_{+j}^{(k)} = f(\tilde{x}_j^{(k)} + b_j \cdot e_j)$ и $f_{-j}^{(k)} = f(\tilde{x}_j^{(k)} - b_j \cdot e_j)$, где e_1, \dots, e_n – стандартный базис

$$x_{j+1}^{(k)} = \begin{cases} \tilde{x}_j^{(k)} + b_j \cdot e_j, & \text{если } f_{+j}^{(k)} < f_j^{(k)}, f_{+j}^{(k)} < f_{-j}^{(k)} \\ \tilde{x}_j^{(k)} - b_j \cdot e_j, & \text{если } f_{-j}^{(k)} < f_j^{(k)}, f_{-j}^{(k)} < f_{+j}^{(k)} \\ \tilde{x}_j^{(k)}, & \text{во всех остальных случаях} \end{cases}$$

получаем $f_{j+1}^{(k)} = f(\tilde{x}_{j+1}^{(k)})$;

Шаг 2. Если $j < n$, то $j = j + 1$ и шаг 1, иначе шаг 3;

Шаг 3. Если $\tilde{x}_{n+1}^{(k)} \neq \tilde{x}^{(k)}$, то шаг 4, иначе уменьшить длину вектора b , т.е. $b = \frac{b}{\gamma}$

, где $\gamma > 1$ – коэффициент дробления шага исследующего поиска.

Полагая $j = 1$, $\tilde{x}_j^{(k)} = \tilde{x}^{(k)}$, $f_j^{(k)} = f(\tilde{x}^{(k)})$ идти на шаг 1;

Шаг 4. Если $\|\tilde{x}_{n+1}^{(k)} - \tilde{x}^{(k)}\| < \varepsilon$, то $x^* = x^{(k-1)}$, $f(x^*) = f(x^{(k-1)})$, конец алгоритма; иначе $\tilde{x}^{(k+1)} = \tilde{x}_{n+1}^{(k)}$ и переходим к этапу спуска в направлении вектора $\tilde{x}^{(k)} - x^{(k-1)}$ при этом $f(\tilde{x}^{(k+1)}) < f(\tilde{x}^{(k)}) < f(x^{(k-1)})$;

Шаг 5. Этап поиска по образцу: На этапе спуска по формуле $x^{(k)} = \tilde{x}^{(k)} + \alpha_k (\tilde{x}^{(k+1)} - \tilde{x}^{(k)})$ подбирают ускоряющий множитель $\alpha_k > 0$, находим такую точку $x^{(k)} : f(x^{(k)}) \leq f(\tilde{x}^{(k+1)})$. Значение α_k можно подобрать из условия минимума функции $f(x)$ при смещении точки $x^{(k)}$ в направлении этого вектора. В простейшем случае $\alpha_k = \text{const}$. Далее $j = 1$, $\tilde{x}_j^{(k+1)} = \tilde{x}^{(k+1)} = x^{(k)}$, $f_j^{(k+1)} = f(x^{(k)})$, $k = k + 1$ и шаг 1.

Конец алгоритма.

ЗАМЕЧАНИЕ. Если $f(x)$ не является дифференцируемой в некоторых точках, либо линии уровня функций сильно вытянуты, изогнуты, либо обладают острыми углами, то методы циклического покоординатного спуска и Хука-Дживса могут оказаться не способными обеспечить продвижение к точке минимума (рисунок 1.18).

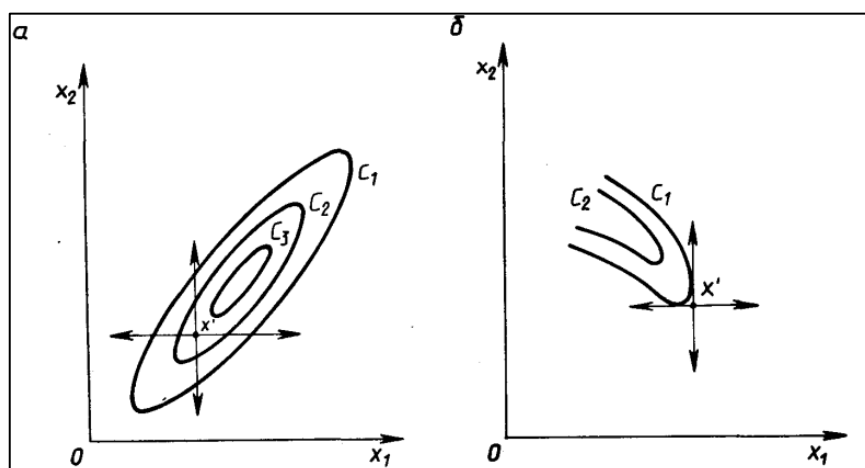


Рисунок 1.19 – Методы прямого поиска для овражных функций

Ясно, что поиск вдоль любой координатной оси в точке x^j не приводит к улучшению целевой функции.

1.2.5 Методы первого порядка. Методы градиентного спуска

Пусть функция $f(x)$ ограничена снизу и непрерывно-дифференцируема на \mathbb{R}^n и пусть существует x^* , в которой функция $f(x)$ принимает минимальное значение.

Известно, что направлением наибольшего возрастания функции f в точке $x^{(0)}$ является направление, задаваемое вектором $\text{grad } f(x^{(0)})$, а $(-\text{grad } f(x^{(0)}))$ задает направление наибольшего убывания функции f в точке $x^{(0)}$. Учитывая это, более естественно осуществлять линейный поиск не в направлении осей координат, а в направлении $-\text{grad } f(x)$.

В методах градиентного спуска элементы релаксационной последовательности удовлетворяют рекуррентному соотношению вида $x^{(k)} = x^{(k-1)} + \alpha_k \cdot h^{(k)}$, $k \in \mathbb{N}$, где $\alpha_k > 0$ шаг спуска, определяемый любым из способов, например, методом расходящегося ряда, дробления шага или путем решения задачи оптимизации $x^{(k)} : f(x^{(k)}) = \min_{\alpha} f(x^{(k-1)} + \alpha_k \cdot h^{(k)})$, а направление $h^{(k)}$ выбирают

$$h^{(k)} = -\text{grad}f(x^{(k-1)}) \text{ либо } h^{(k)} = -\frac{\text{grad}f(x^{(k-1)})}{\|\text{grad}f(x^{(k-1)})\|}.$$

При разложении функции в окрестности точки $x^{(k)}$ в ряд Тейлора, минимум этого разложения достигается, если направление шага совпадает с направлением антиградиента функции в точке $x^{(k)}$.

ЗАМЕЧАНИЕ. Если выбор осуществляется из условия минимизации функции вдоль заданного направления, то метод называется метод наискорейшего спуска. Геометрическая трактовка метода приведена на рисунках 1.20, 1.21.

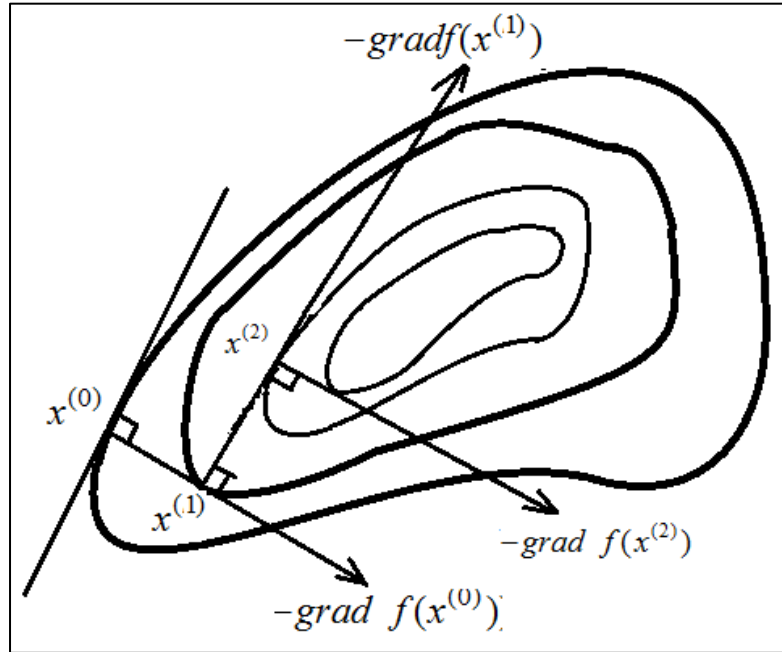


Рисунок 1.20 – Иллюстрация метода наискорейшего спуска (линии уровня)

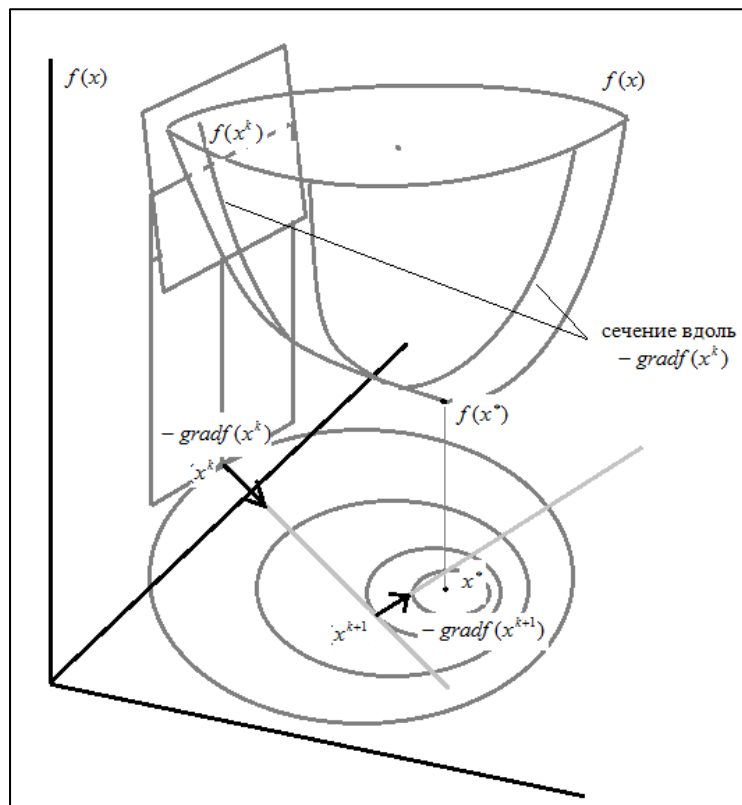


Рисунок 1.21 – Геометрическая интерпретация метода наискорейшего спуска

Алгоритм метода наискорейшего спуска:

Шаг 1. Задаем начальный вектор $x^{(0)}$, точность ε и положим $k=1$;

Шаг 2. Находим направление спуска $h^{(k)} = -\text{grad}f(x^{(k-1)})$;

Шаг 3. Находим $x^{(k)} : f(x^{(k)}) = \min_{\alpha} f(x^{(k-1)} + \alpha_k \cdot h^{(k)})$ (см. алгоритм поиска

вдоль заданного направления) $x^{(k)} = x^{(k-1)} + \alpha_k \cdot h^{(k)}$;

Шаг 4. Если $\|x^{(k)} - x^{(k-1)}\| < \varepsilon$ или $\|\text{grad}f(x^{(k)})\| < \varepsilon$, то $x^* = x^{(k)}$,

иначе $k=k+1$ и переходим к шагу 2.

Конец алгоритма.

ЛЕММА 1.3

Пусть функция $f(x)$ дифференцируема на множестве R^n и ее градиент удовлетворяет условию Липшица: $\forall x, y : \|\text{grad}f(x) - \text{grad}f(y)\| \leq L \cdot \|x - y\|$

следовательно $\forall x^{(k)} \in R^n, \varepsilon \in (0,1), h^{(k)} \in U(h^{(k)}, f)$, где

$U(h^{(k)}, f(x^{(k)} + \alpha h^{(k)}) < f(x^{(k)}))$ – множество возможных значений убывания функции в точке $x^{(k)}$:

$$f(x^{(k)} + \alpha_k h^{(k)}) - f(x^{(k)}) \leq \varepsilon \cdot \alpha_k < \text{grad}f(x^{(k)}), h^{(k)} >, \quad (1.25)$$

при $\alpha_k > 0$ и

$$\alpha_k \leq -(1 - \varepsilon) \cdot \frac{< \text{grad}f(x^{(k)}), h^{(k)} >}{L \cdot \|h^{(k)}\|^2}. \quad (1.26)$$

ТЕОРЕМА 1.5 (О сходимости метода дробления шага).

Пусть функция $f(x)$ дифференцируема на множестве R^n , тогда $\forall x^{(k)} \in R^n, \varepsilon \in (0,1), h^{(k)} \in U(h^{(k)}, f)$ найдется $\alpha_0 > 0$ такая что, для $\forall \alpha > 0$ и $\alpha < \alpha_0$ будет выполняться условие (1.25).

ЗАМЕЧАНИЕ. Для метода градиентного спуска:

$$\begin{aligned}
\mathbf{h}^{(k)} &= -\text{grad}f(\mathbf{x}^{(k)}), \\
f(\mathbf{x}^{(k)} + \alpha_k \mathbf{h}^{(k)}) - f(\mathbf{x}^{(k)}) &\leq -\varepsilon \cdot \alpha_k < \text{grad}f(\mathbf{x}^{(k)}), \text{grad}f(\mathbf{x}^{(k)}) >, \\
f(\mathbf{x}^{(k)} + \alpha_k \mathbf{h}^{(k)}) - f(\mathbf{x}^{(k)}) &\leq -\varepsilon \cdot \alpha_k \cdot \|\text{grad}f(\mathbf{x}^{(k)})\|^2, \\
\alpha_k &\leq (1 - \varepsilon) \cdot \frac{\|\text{grad}f(\mathbf{x}^{(k)})\|^2}{L \cdot \|\text{grad}f(\mathbf{x}^{(k)})\|^2}, \\
\alpha_k &\leq \frac{(1 - \varepsilon)}{L}. \tag{1.27}
\end{aligned}$$

Сходимость в случае невыпуклой функции: пусть функция $f(x)$ дифференцируема и ограничена снизу на множестве \mathbf{R}^n и ее градиент удовлетворяет условию Липшица: $\forall x, y : \|\text{grad}f(x) - \text{grad}f(y)\| \leq L \cdot \|x - y\|$. Тогда при произвольном начальном приближении для метода градиентного спуска при α_k удовлетворяющего условию (1.27) обеспечивается сходимость к стационарной точке последовательности $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \cdot \text{grad}f(\mathbf{x}^{(k)})$.

Сходимость в случае выпуклой функции: пусть функция $f(x)$ дважды дифференцируема и сильно выпукла на множестве \mathbf{R}^n и ее матрица Гессе ограничена сверху. Тогда для любой точки начального приближения последовательность $\{\mathbf{x}^{(k)}\}$ получаемая по методу градиентного спуска сходится к точке минимума со скоростью геометрической прогрессии:

$$\begin{aligned}
|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^*)| &\leq q^k \cdot |f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(0)})|, \\
\|\mathbf{x}^{(k)} - \mathbf{x}^*\| &\leq (\sqrt{q})^k \cdot c, c = \text{const} > 0, q \in (0,1).
\end{aligned}$$

Недостатком метода наискорейшего спуска является зачастую медленная сходимость в окрестности стационарной точки. Это очевидно, например, в тех случаях, когда линии уровня вытянуты в окрестности оптимальной точки. О

функциях, поверхности уровня которых сильно вытянуты, говорят, что она имеет «овражный» характер. Геометрически медленная сходимость объясняется так: спустившись на «дно оврага» мы, двигаясь в направлении $(-\text{grad } f(x))$, будем переходить с одного склона оврага на другой, то есть зигзагообразно продвигаться к точке минимума.

Дважды непрерывно дифференцируемая функция называется «овражной» если существует некоторая область $G \subset R^n$, где собственные числа матрицы Гессе $\left(\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right)$ ($i = \overline{1, n}; j = \overline{1, n}$), упорядоченные в любой точке $x \in G \subset R^n$ по убыванию

модулей $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$, удовлетворяют неравенствам $|\lambda_1| \gg \left| \min_{\lambda_i \neq 0} \lambda_i \right| > 0$ [38, 40].

Степень «овражности» можно охарактеризовать значением: $S = \lambda_1 / \left| \min_{\lambda_i \neq 0} \lambda_i \right|$. Чем оно больше, тем больше «овражность».

Более предпочтительными в таких случаях являются методы Ньютона и сопряженных градиентов. Коротко суть первого из них в том, что $f(x)$ аппроксимируется (с помощью формулы Тейлора) многочленами второй степени, для которых ищутся точки минимума. Последовательность таких точек приводит при определенных условиях к искомой точке минимума $f(x)$. Неудобством метода является необходимость многократного обращения матрицы Гессе. Второй из методов требует для получения очередной точки проведения последовательной минимизации по каждому из n переменных, специальным образом построенных направлений, которые называют сопряженными. Для квадратичной функции минимизация вдоль « n » таких направлений позволяет «точно» достичь точки минимума, а следовательно можно ожидать хороших результатов и для достаточно гладких функций. Рассмотрим подробнее указанные методы.

1.2.6 Методы второго порядка. Метод Ньютона и его модификации

Пусть функция $f(x)$ дважды дифференцируема в области определения $D \subset \mathbb{R}^n$. Стоит задача найти минимум функции $f(x)$, т.е.

$$f(x) \rightarrow \min, x \in D \subset \mathbb{R}^n. \quad (1.28)$$

Пусть $x^{(k-1)}$ – это точка на очередном этапе поиска. Разложим функцию $f(x)$ в ряд Тейлора в окрестности точки $x^{(k-1)}$:

$$f(x) = f(x^{(k-1)}) + \langle \text{grad}f(x^{(k-1)}), x - x^{(k-1)} \rangle + \frac{1}{2} \langle H(x^{(k-1)}) \cdot (x - x^{(k-1)}), x - x^{(k-1)} \rangle + o\left(\|x - x^{(k-1)}\|^2\right),$$

где $H(x)$ – матрица Гессе.

Пренебрегая последним слагаемым в правой части, получаем квадратичную функцию.

$$F_k(x) = f(x^{(k-1)}) + \langle \text{grad}f(x^{(k-1)}), x - x^{(k-1)} \rangle + \frac{1}{2} \langle H(x^{(k-1)}) \cdot (x - x^{(k-1)}), x - x^{(k-1)} \rangle \rightarrow \min.$$

Для решения этой задачи найдем стационарные точки, приравняв к нулю производную

$$F'_k(x) = \text{grad}f(x^{(k-1)}) + H(x^{(k-1)}) \cdot (x - x^{(k-1)}) = 0,$$
$$H(x^{(k-1)}) \cdot (x - x^{(k-1)}) = -\text{grad}f(x^{(k-1)}).$$

Отсюда получаем новое приближение:

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \mathbf{H}^{-1}(\mathbf{x}^{(k-1)}) \cdot \text{grad}f(\mathbf{x}^{(k-1)}). \quad (1.29)$$

На практике вместо задачи (1.29) решают систему линейных алгебраических уравнений вида (1.30), затем находят очередное приближение в соответствии с выражением (1.31):

$$\mathbf{H}(\mathbf{x}^{(k-1)}) \cdot \mathbf{p}^{(k)} = -\text{grad}f(\mathbf{x}^{(k-1)}), \quad (1.30)$$

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{p}^{(k)}. \quad (1.31)$$

Геометрическая интерпретация Ньютоновского направления спуска приведена на рисунке 1.21.

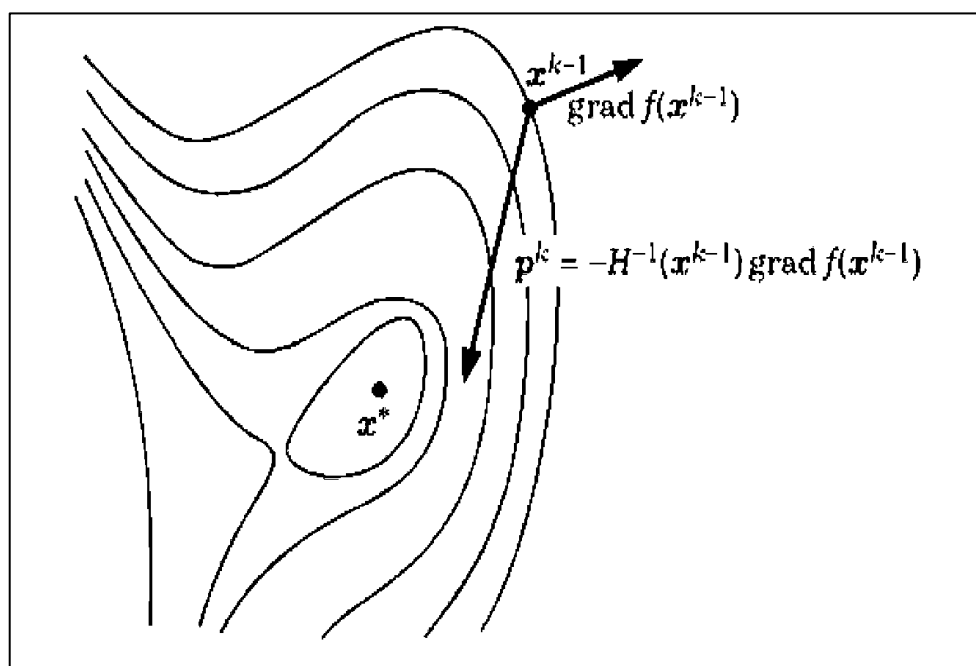


Рисунок 1.21 – Ньютоновское направление спуска [31]

ТЕОРЕМА 1.6 (О сходимости метода Ньютона) [11]

Пусть функция $f(\mathbf{x})$ дважды дифференцируема на \mathbb{R}^n , ее матрица Гессе

положительно определена, т.е. $\langle H(x) \cdot h, h \rangle \geq d \cdot \|h\|^2$, $\forall h \in \mathbb{R}^n$ и выполняется условие Липшица, т.е. $\forall x, y: \|H(x) - H(y)\| \leq L \cdot \|x - y\|$, начальная точка $x^{(0)}$ выбрана достаточно близко к точке минимума, причем $x^{(0)}: \|H(x^{(0)})\| \leq \frac{2 \cdot c \cdot d^2}{L}$, где c – некоторая константа $0 < c < 1$, тогда последовательность $\{x^{(k)}\}$, сформированная по методу Ньютона сходится к точке минимума с квадратичной скоростью сходимости и справедлива оценка:

$$\|x^{(k)} - x^*\| \leq \frac{2 \cdot c^{2^k} \cdot d}{L}, \quad k = 0, 1, \dots$$

ЗАМЕЧАНИЕ. Для квадратичной функции метод Ньютона сходится за одну итерацию.

Достоинство метода: быстрая скорость сходимости.

Недостатки метода:

1) на каждой итерации необходимо решать СЛАУ (1.30);

2) требуется положительная определенность матрицы Гессе на каждом шаге

$$|H(x^{(k)})| > 0;$$

3) в случае, когда целевая функция не является сильно выпуклой или же начальное приближение находится далеко от искомой точки x^* , метод Ньютона может расходиться, по этой причине рекомендуется на первом этапе для нахождения начального приближения использовать градиентные методы или методы нулевого порядка, а уже после применять метод Ньютона для уточнения точки минимума.

С целью ускорения сходимости и ослабления указанных недостатков предложены модификации метода Ньютона, один из таких методов - метод Ньютона с регуляризацией шага. В методе с регуляризацией шага рекуррентная последовательность задается следующим образом:

$$x^{(k)} = x^{(k-1)} - \alpha_k \cdot H^{-1}(x^{(k-1)}) \cdot \text{grad}f(x^{(k-1)}).$$

Выбор коэффициентов α_k осуществляется из условия $x^{(k)} : f(x^{(k)}) = \min_{\alpha} f(x^{(k-1)} - \alpha_k \cdot H^{-1}(x^{(k-1)}) \cdot \text{grad}f(x^{(k-1)}))$ или методом дробления шага. С целью уменьшения трудоемкости метода можно вычислить $H^{-1}(x^{(k-1)})$ не на каждом шаге, а через S шагов, где S выбирается эмпирически.

1.2.7 Квазиньютоновские методы

Квазиньютоновские методы объединяют достоинства метода Ньютона и наискорейшего спуска. Основная идея состоит в аппроксимации матрицы $H^{-1}(x^{(k)})$, на основе информации о значениях в предыдущих точках. Итерационную последовательность строят следующим образом:

$$x^{(k+1)} = x^{(k)} + \alpha_k \cdot h^{(k)}, \quad (1.32)$$

$$h^{(k)} = -A^{(k)} \text{grad}(x^{(k)}), \quad (1.33)$$

$$\alpha_k = \arg \min_{\alpha} f(x^{(k)} + \alpha_k \cdot h^{(k)}). \quad (1.34)$$

Матрица $A^{(k)}$ выбирается так, чтобы она аппроксимировала $H^{-1}(x^{(k)})$. A^k – положительно определенная матрица.

Последовательность $\{A^{(k)}\}$ определяют таким образом, чтобы $\lim_{k \rightarrow \infty} A^{(k)} = H^{-1}(x^*)$, где x^* – минимум $f(x)$.

Направление $h^{(k)}$, задаваемое на k -ой итерации, является направлением спуска:

$$\langle \text{grad}f(x^{(k)}), h^{(k)} \rangle = - \langle \text{grad}f(x^{(k)}), A^{(k)} \cdot \text{grad}f(x^{(k)}) \rangle < 0.$$

Разложим функцию $\text{grad}f(x^{(k)})$ по формуле Тейлора в окрестности точки $x^{(k+1)}$:

$$\text{grad}f(x^{(k)}) = \text{grad}f(x^{(k+1)}) + H(x^{(k+1)}) \cdot (x^{(k)} - x^{(k+1)}) + o(\|x^{(k)} - x^{(k+1)}\|).$$

Отбрасывая последнее слагаемое, получаем:

$$\underbrace{x^{(k)} - x^{(k+1)}}_{\Delta x^{(k)}} = H^{-1}(x^{(k+1)}) \cdot \underbrace{(\text{grad}f(x^{(k)}) - \text{grad}f(x^{(k+1)}))}_{\Delta y^{(k)}}, \quad (1.35)$$

$$\Delta x^{(k)} \approx H^{-1}(x^{(k+1)}) \cdot \Delta y^{(k)}, \quad (1.36)$$

$$\Delta x^{(k)} = A^{(k+1)} \cdot \Delta y^{(k)}. \quad (1.37)$$

Выражение (1.37) называют «квазиньютоновское условие».

$$A^{(k+1)} = A^{(k)} + \Delta A^{(k)}, \quad A^{(0)} = E,$$

где $\Delta A^{(k)}$ – поправочная матрица.

Таким образом:

$$\begin{aligned} \Delta x^{(k)} &= (A^{(k)} + \Delta A^{(k)}) \cdot \Delta y^{(k)} = A^{(k)} \cdot \Delta y^{(k)} + \Delta A^{(k)} \cdot \Delta y^{(k)}, \\ \Delta A^{(k)} \cdot \Delta y^{(k)} &= \Delta x^{(k)} - A^{(k)} \cdot \Delta y^{(k)} \end{aligned}$$

Выберем $\Delta A^{(k)}$ из условия: $\lim_{k \rightarrow \infty} \|A^{(k)} - H^{-1}(x^*)\| = 0$. Этому условию

удовлетворяет матрица вида:

$$\Delta A^{(k)} = -\frac{\Delta x^{(k)} (\Delta x^{(k)})^T}{\langle \Delta y^{(k)}, \Delta x^{(k)} \rangle} - \frac{A^{(k)} \cdot \Delta y^{(k)} \cdot (z^{(k)})^T}{\langle \Delta y^{(k)}, z^{(k)} \rangle},$$

где $d, z \in R^n$ произвольные вектора.

Наибольшее распространение получили следующие способы задания матрицы $\Delta A^{(k)}$, различающиеся, как правило, выбором векторов $d, z \in R^n$.

1. Метод Девидона-Флетчера-Пауэлла.

Пусть $d^{(k)} = \Delta x^{(k)}$, $z^{(k)} = A^{(k)} \Delta y^{(k)}$, тогда:

$$\Delta A^{(k)} = -\frac{\Delta x^{(k)} (\Delta x^{(k)})^T}{\langle \Delta y^{(k)}, \Delta x^{(k)} \rangle} - \frac{A^{(k)} \Delta y^{(k)} (A^{(k)} \Delta y^{(k)})^T}{\langle A^{(k)} \Delta y^{(k)}, \Delta y^{(k)} \rangle}. \quad (1.38)$$

Метод эффективен на практике, если ошибка вычисления градиента невелика и матрица $A^{(k)}$ не становится плохо обусловленной.

2. Метод Пауэлла.

Пусть $d^{(k)} = z^{(k)} = \Delta x^{(k)} - A^{(k)} \cdot \Delta y^{(k)}$, тогда:

$$\Delta A^{(k)} = \frac{(\Delta x^{(k)} + A^{(k)} \cdot \Delta y^{(k)}) \cdot (\Delta x^{(k)} + A^{(k)} \cdot \Delta y^{(k)})^T}{\langle \Delta y^{(k)}, \Delta x^{(k)} + A^{(k)} \cdot \Delta y^{(k)} \rangle}. \quad (1.39)$$

3. Метод Бroyдена-Флетчера-Шенно:

$$\Delta A^{(k)} = -\frac{\Delta x^{(k)} (\Delta x^{(k)})^T}{\langle \Delta y^{(k)}, \Delta x^{(k)} \rangle} - \frac{A^{(k)} \cdot \Delta y^{(k)} \cdot (A^{(k)} \cdot \Delta y^{(k)})^T}{\rho^{(k)}} + \rho^{(k)} \cdot \tau^{(k)} \cdot (\tau^{(k)})^T, \quad (1.40)$$

$$\tau^{(k)} = \frac{A^{(k)} \cdot \Delta y^{(k)}}{\rho^{(k)}} - \frac{\Delta x^{(k)}}{\langle \Delta x^{(k)}, \Delta y^{(k)} \rangle}; \rho^{(k)} = \langle A^{(k)} \cdot \Delta y^{(k)}, \Delta y^{(k)} \rangle. \quad (1.41)$$

БФШ-метод обладает слабой чувствительностью к погрешности вычислений, не требует обновления матрицы $A(k)$.

4. Метод Маккормика:

$$\Delta A^{(k)} = - \frac{(\Delta x^{(k)} + A^{(k)} \cdot \Delta y^{(k)}) \cdot (\Delta x^{(k)})^T}{\langle \Delta y^{(k)}, \Delta x^{(k)} \rangle}. \quad (1.42)$$

В качестве $A^{(0)}$ берут единичную матрицу.

Можно показать, что при минимизации квадратичной функции с положительно определенной матрицей квадратичной формы, алгоритм квазиньютоновского метода позволяет получить минимум не более чем за n итераций.

Если минимизируемая целевая функция не является квадратичной, то квазиньютоновские методы не позволяют найти минимум за конечное число итерации. В этом случае процедуру поиска осуществляют до достижения заданной степени точности вычислений, в соответствии с выбранным критерием останова (1.16)-(1.19), и при этом через каждые n итерации применяют процедуру «обновления» алгоритма, т.е. принимают $A^{(k)}$, где $k=n, 2n, \dots$ равной единичной матрице.

1.2.8 Методы сопряженных направлений, сопряженных градиентов

Пусть $Q(x)$ – произвольная квадратичная функция:

$$Q(x) = \frac{1}{2} x^T A x + b^T x + c, \quad (1.43)$$

где $A = \{a_{ij}\}_{n \times n}$ – симметричная положительно определенная матрица;

$b = \{b_i\}_{n \times 1}$ – вектор;

c – скалярная величина;

$x = \{x_i\}_{n \times 1}$ – вектор переменных.

Поставим задачу минимизации квадратичной функции:

$$f(x) = Q(x) = \frac{1}{2} x^T A x + b^T x + c \rightarrow \min. \quad (1.44)$$

Рассмотрим некоторые свойства квадратичной функции. Так как $A = \{a_{ij}\}_{n \times n}$ симметричная положительно определенная матрица, следовательно $x^T A x > 0, \forall x \neq 0$. Градиент функции

$$g(x) = \text{grad} f(x) = A x + b, \quad (1.45)$$

а матрица Гессе $H(x) = A$. Следовательно, для приращения аргумента функции вида (1.43) справедливо: $g(x + \Delta x) = A(x + \Delta x) + b$. Тогда приращение градиентов $\Delta g = g(x + \Delta x) - g(x) = A(x + \Delta x) - A x = A \cdot \Delta x$:

$$\Delta g = A \cdot \Delta x, \quad (1.46)$$

Пусть x^* – точка минимума функции (1.43), тогда $g(x) - g(x^*) = A(x - x^*)$, а согласно необходимому условию минимума $g(x^*) = \text{grad} Q(x^*) = 0$, следовательно:

$$g(x) = A(x - x^*).$$

Поскольку квадратичная функция является выпуклой $g(x^*) = Ax^* + b = 0$, следовательно:

$$x^* = A^{-1}b, \quad (1.47)$$

при этом:

$$f^* = f(x^*) = \frac{1}{2} x^{*T} Ax^* + b^T x^* + c = \frac{1}{2} b^T A^{-1} A A^{-1} b - b^T A^{-1} b + c = c - \frac{1}{2} b^T A^{-1} b.$$

Пусть рекуррентное соотношение, как и ранее, имеет вид:

$$x^{(k+1)} = x^{(k)} + \alpha_k \cdot h^{(k)}, \quad (1.48)$$

где α_k находим из условий минимизации функции

$\alpha_k : \varphi(\alpha) = f(x^{(k)} + \alpha_k \cdot h^{(k)}) \rightarrow \min_{\alpha}$ вдоль заданного направления $h^{(k)}$ из точки

$x^{(k)}$. Так как $\varphi'(\alpha_k) = g(x^{(k)} + \alpha_k \cdot h^{(k)})^T h^{(k)} = 0$, тогда условие поиска α_k

сводится к следующему условию ортогональности:

$$g(x^{(k+1)})^T h^{(k)} = 0, \quad (1.49)$$

учитывая (1.46) и (1.48) получим:

$$g(x^{(k+1)}) - g(x^{(k)}) = A(x^{(k+1)} - x^{(k)}) = A\alpha_k h^{(k)},$$

$$g(x^{(k+1)}) = g(x^{(k)}) + A\alpha_k h^{(k)}.$$

Подставим в (1.49):

$$(\mathbf{g}(\mathbf{x}^{(k)}) + \mathbf{A}\alpha_k \mathbf{h}^{(k)})^T \mathbf{h}^{(k)} = 0.$$

Так как матрица \mathbf{A} положительно определена и выполняется свойство $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0, \forall \mathbf{x} \neq \mathbf{0}$, то:

$$\alpha_k = -\frac{\mathbf{g}(\mathbf{x}^{(k)})^T \mathbf{h}^{(k)}}{\mathbf{h}^{(k)T} \mathbf{A} \mathbf{h}^{(k)}}. \quad (1.50)$$

Условие (1.50) задает условие точного одномерного поиска α_k . Приведенные свойства используются при построении эффективных методов оптимизации.

Пусть выбрана произвольная точка $\mathbf{x}^{(0)}$, в которой для функции $f(\mathbf{x})$ выполняется:

$$\begin{aligned} -\text{grad}f(\mathbf{x}^{(0)}) &= -\mathbf{A}\mathbf{x}^{(0)}, \\ k=1: \mathbf{x}^{(1)} &= \mathbf{x}^{(0)} + \alpha_0 \mathbf{h}^{(0)}, \end{aligned}$$

$\mathbf{x}^* = \min f(\mathbf{x})$ можно достичь из произвольной точки \mathbf{x}^0 за одну итерацию, т.е. $\mathbf{x}^1 = \mathbf{x}^*$, если выбрать $\alpha_1 \cdot \mathbf{h}^1 = -\mathbf{x}^0 = \mathbf{A}^{-1} \cdot \text{grad}f(\mathbf{x}^0)$.

Такое направление спуска называется исчерпывающим направлением спуска. При этом направление спуска не совпадает с направлением антиградиента. Последовательность исчерпывающих направлений спуска образует зигзагообразную траекторию в направлении антиградиента приближений к точке минимума. Причем все соседние звенья ломанной ортогональны, т.е. $-\text{grad}f(\mathbf{x}^k)$ ортогонален \mathbf{h}^k :

$$\langle -\text{grad}f(\mathbf{x}^k), \mathbf{h}^k \rangle = 0. \quad (1.51)$$

ОПРЕДЕЛЕНИЕ. Ненулевые вектора \mathbf{h}^1 и \mathbf{h}^2 называются сопряженными

относительно матрицы A , если $\langle A \cdot h^1, h^2 \rangle = 0$. При этом говорят что h^1 и h^2 A -ортогональны.

Если матрица A равна единичной матрице E , то условие сопряженности примет вид $\langle h^1, h^2 \rangle = 0$ и условие сопряженности эквивалентно условию ортогональности.

ОПРЕДЕЛЕНИЕ. Ненулевые вектора h^0, h^1, \dots, h^{n-1} пространства R^n называются A -сопряженными, если $\langle A \cdot h^i, h^j \rangle = 0, \forall i, j: i \neq j$.

ТЕОРЕМА 1.7

Пусть вектора h^0, h^1, \dots, h^{n-1} A -сопряженные, тогда они линейно-независимые.

СЛЕДСТВИЕ Сопряженные вектора h^0, h^1, \dots, h^{n-1} образуют базис в пространстве R^n .

ТЕОРЕМА 1.8 (Свойство параллельного подпространства)

Пусть h – некоторое направление поиска в пространстве R^n , а u и v – две различные точки минимума квадратичной функции $f(x)$, полученные из двух точек u_0 и v_0 в направлении h . Тогда направление $s = u - v$ сопряжено направлению h .

По теореме 1.8 выполнение двух одномерных поисков из разных точек в направлении h позволяет получить сопряженное к h направление s (рисунок 1.22). Точка минимума квадратичной функции $f(x)$ при $n=2$ может быть найдена одномерным поиском из точек u или v в направлении s .

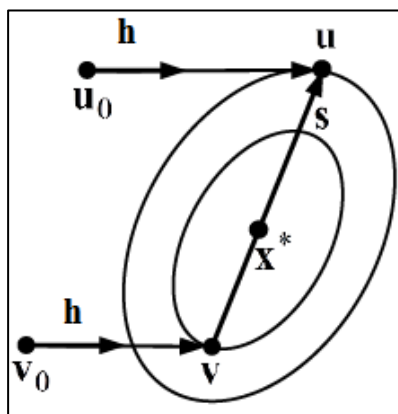


Рисунок 1.22 – Сопряженные направления

Гладкие овражные функции вблизи дна оврага можно аппроксимировать квадратичными функциями. Поэтому свойства сопряженных направлений позволяют находить эффективные направления поиска для любых гладких функций.

Методами сопряженных направлений для функции n переменных называются итерационные методы с последовательным одномерным поиском в n сопряженных направлениях.

ТЕОРЕМА 1.9

Пусть вектора h^k в методе (1.48) взаимно сопряженные для $k = 0, \dots, n-1$, тогда для (1.44) значение минимума достигается за n этап (т.е. за n итераций $k = 0, \dots, n-1$), причем в конечной точке любой итерации градиент ортогонален всем предыдущим направлениям поиска.

ТЕОРЕМА 1.10

Точку минимума функции $f(x)$ с положительно определенной матрицей A можно найти не более, чем за n итерации вида (1.48), где $h^k \in \mathbb{R}^n$ сопряжены относительно матрицы A , если на k -ой итерации

$$\alpha^k = \frac{g(x^{(k)})^T h^{(k)}}{h^{(k)T} A h^{(k)}} = -\frac{\langle Ax^0, h^k \rangle}{\langle Ah^k, h^k \rangle}, k = 1, n.$$

ЗАМЕЧАНИЕ. Градиентные методы являются медленно сходящимися и их используют для нахождения начального приближения точки минимума. Для уточнения начального приближения используются быстросходящиеся вблизи точки минимума методы, такие как метод Ньютона, квазиньютоновские методы, метод сопряженный градиентов.

Опишем метод сопряженных градиентов для квадратичной функции (1.44).

В методе сопряженных градиентов итерационная последовательность строится в соответствии с рекуррентными выражениями:

$$x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}, k = 1, n,$$

$$\alpha_k = \arg \min_{\alpha} f(x^{(k)} + \alpha h^{(k)}),$$

$$h^{(0)} = -\text{grad}f(x^{(0)}), \quad (1.52)$$

$$h^{(k)} = -\text{grad}f(x^{(k)}) + \beta_{k-1}h^{(k-1)}, \quad k > 1,$$

$$\beta_{k-1} = \frac{\langle \text{grad}f(x^{(k)}), Ah^{(k-1)} \rangle}{\langle Ah^{(k-1)}, h^{(k-1)} \rangle}.$$

Метод вида (1.52) носит название Флетчера – Ривса. Геометрическая интерпретация метода сопряженных градиентов приведена на рисунке 1.23, 1.24.

УТВЕРЖДЕНИЕ 1.3

Пусть точка $x^{(0)} \in \mathbb{R}^n$ – произвольная точка, а $x^{(1)}, x^{(2)} \dots x^{(n-1)}$ получены по методу (1.52) и $\text{grad}f(x^{(i)}) \neq 0, i = 1, 2, \dots, n-1$. Тогда $h^{(0)}, h^{(1)}, \dots, h^{(n-1)}$ взаимно сопряжены, а градиенты взаимно ортогональны. Метод (1.52) обеспечивает отыскание минимума функции $f(x)$ не более чем за n итераций.

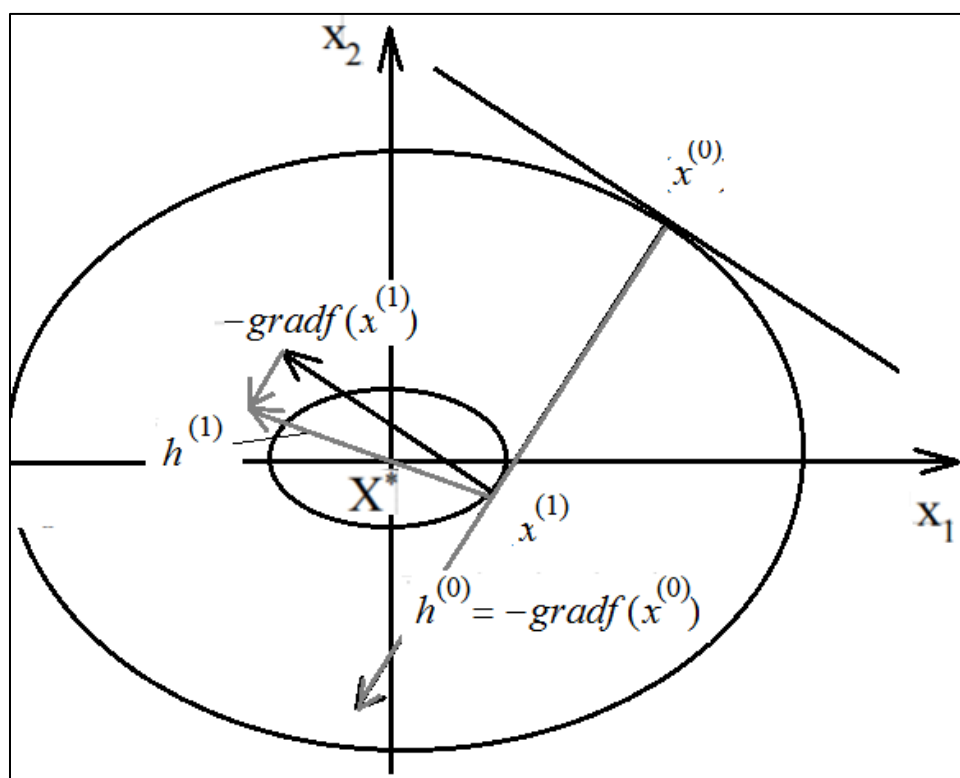


Рисунок 1.23 – Геометрическая интерпретация метода сопряженных градиентов

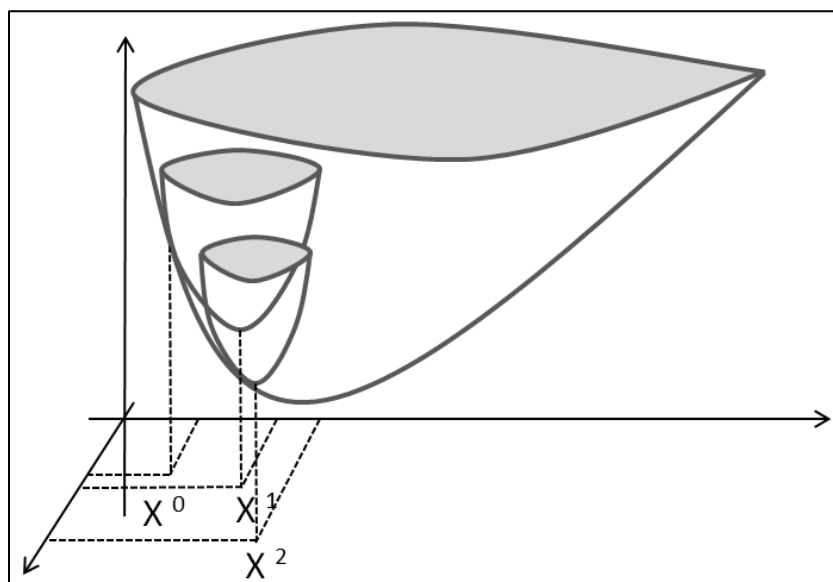


Рисунок 1.24 – Геометрическая интерпретация метода сопряженных градиентов

Опишем метод сопряженных градиентов для функции общего вида:

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

В этом случае метод сопряженных градиентов не позволяет получить минимум функции за n итераций, поэтому поиск точки минимума ведут до выполнения выбранного критерия окончания итераций (1.16)-(1.19), и каждые n итераций производят обновление параметров алгоритма: обнуляют параметр β_{k-1} .

Итерационная последовательность строится в соответствии с рекуррентным выражением:

$$x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)},$$

где $\alpha_k = \arg \min_{\alpha} f(x^{(k)} + \alpha h^{(k)}),$

$$h^{(0)} = -\text{grad}f(x^{(0)}),$$

$$h^{(k)} = -\text{grad}f(x^{(k)}) + \beta_{k-1} h^{(k-1)} \quad \text{при } k \geq 1, \quad \beta_{k-1} \quad \text{определяются следующим}$$

образом:

$$\beta_{k-1} = \begin{cases} \frac{\langle \text{grad}f(x^{(k)}), \text{grad}f(x^{(k)}) - \text{grad}f(x^{(k-1)}) \rangle}{\|\text{grad}f(x^{(k-1)})\|^2}, & \text{при } k \neq n, 2n, 3n\dots \\ 0, & \text{при } k = n, 2n, 3n\dots \end{cases} \quad (1.53)$$

или

$$\beta_{k-1} = \begin{cases} \frac{\|\text{grad}f(x^{(k)})\|^2}{\|\text{grad}f(x^{(k-1)})\|^2}, & \text{при } k \neq n, 2n, 3n\dots \\ 0, & \text{при } k = n, 2n, 3n\dots \end{cases} \quad (1.54)$$

Вычисления продолжают до тех, пока не выполнится выбранный критерий останова (1.16)-(1.19), к примеру $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$.

1.2.9 Вопросы для самопроверки

1. Охарактеризуйте понятие релаксационной последовательности, приведите свойства релаксационных последовательностей.
2. Опишите принципы выбора длины шага в методах спуска (из условия минимизации функции вдоль заданного направления, априорный выбор коэффициентов, метод дробления шага).
3. Опишите метод циклического покоординатного спуска, модифицированный метод циклического покоординатного спуска, приведите алгоритмы.
4. Опишите метод Хука-Дживса, приведите алгоритм.
5. Приведите алгоритм поиска вдоль заданного направления.
6. Опишите градиентный метод расходящегося ряда, приведите алгоритм.
7. Опишите метод наискорейшего спуска, приведите алгоритм.
8. Опишите градиентный метод с дроблением шага, охарактеризуйте

сходимость метода, приведите алгоритм.

9. Опишите метод Ньютона, охарактеризуйте сходимость, приведите алгоритм, отметьте достоинства и недостатки. Приведите модификации метода.

10. Опишите квазиньютоновские методы, приведите алгоритм метода для квадратичной функции.

11. Опишите квазиньютоновские методы, приведите алгоритм метода для неквадратичной функции.

12. Опишите идеологию методов сопряженных направлений, приведите обоснование.

13. Приведите алгоритм метода сопряженных градиентов для квадратичных функций.

14. Приведите алгоритм метода сопряженных градиентов для неквадратичных функций.

1.2.10 Задания для практических (лабораторных) работ

Рекомендации к выполнению и оформлению практических (лабораторных) работ по разделу приведены в источнике 17.

Дана целевая функция многих переменных $f(x) = f(x_1, \dots, x_n)$ определенная на множестве $D \subset \mathbb{R}^n$.

Задание 1.4 Разработать программу для нахождения безусловного минимума целевой функции методом прямого поиска первого порядка. Варианты задания приведены в таблице 1.3.

Задание 1.5 Разработать программу для нахождения начального приближения к точке минимума целевой функции градиентным методом. Варианты задания приведены в таблице 1.3.

Таблица 1.3 – Варианты заданий

| № варианта | Функция | Методы | | |
|------------|---|-------------------------------------|----------------------|---|
| | | задание 1.4 | задание 1.5 | задание 1.6 |
| 1 | 2 | 3 | 4 | 5 |
| 1 | $2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_2 - 4x_1$ | Хука-Дживса | расходящегося ряда | ДФП |
| 2 | $2x_1^2 - 4x_1 + x_2^2 - 2x_2 + x^2 y$ | циклического покоординатного спуска | дробления шага | Квазиньтоновский Пауэлла |
| 3 | $10x_1^2 - 4x_1x_2 + 7x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | Хука-Дживса | наискорейшего спуска | БФШ |
| 4 | $(x_1 - 5)^2 + (x_2 - 3)^2 + (x_3 - 7)^2$ | циклического покоординатного спуска | расходящегося ряда | сопряженных градиентов (Флетчера-Ривса) |
| 5 | $\frac{1}{3}(x_1 + 1)^2 + (x_2 - 2)^2 + x_1x_2$ | Хука-Дживса | наискорейшего спуска | БФШ |
| 6 | $(x_1 - 5)^2 + (x_2 - 3)^2$ | циклического покоординатного спуска | Дробления шага | Квазиньтоновский Пауэлла |
| 7 | $(x_1 - 4)^2 + (x_2 + 1)^2 + x_1x_2$ | Хука-Дживса | расходящегося ряда | ДФП |
| 8 | $x_1x_2x_3^2$ | Хука-Дживса | Дробления шага | БФШ |
| 9 | $\frac{1}{3}(x_1 - 1)^2 + (x_2 - 2)^4 + x_1x_2$ | циклического покоординатного спуска | наискорейшего спуска | сопряженных градиентов (Флетчера-Ривса) |
| 10 | $2(x_1 + 1)^2 + 3x_2^2 + x_2 + (x_3 - 3)^2$ | циклического покоординатного спуска | расходящегося ряда | Квазиньтоновский Пауэлла |
| 11 | $(x_1 - 15)^2 + (x_2 - 3)^2 + x_1x_3^2$ | Хука-Дживса | наискорейшего спуска | ДФП |
| 12 | $\frac{5}{2x_1} + \frac{9}{x_2} + 2x_1 + x_2$ | циклического покоординатного спуска | Дробления шага | сопряженных градиентов (Флетчера-Ривса) |
| 13 | $(x_1 - 2)^2 - 3x_1 + (y - 1)^2$ | Хука-Дживса | расходящегося ряда | БФШ |
| 14 | $12x_1^2 - 2x_1x_2 + 6x_2^2 - 4\sqrt{5}(3x_1 - x_2) - 10$ | циклического покоординатного спуска | наискорейшего спуска | Квазиньтоновский Пауэлла |
| 15 | $x_1^2 + x_2^2$ | Хука-Дживса | Дробления шага | ДФП |
| 16 | $(x_1 - 2)^2 - 3x_1 + (y - 1)^2 + (z + 3)^2$ | циклического покоординатного спуска | расходящегося ряда | сопряженных градиентов (Флетчера-Ривса) |

Продолжение таблицы 1.3

| 1 | 2 | 3 | 4 | 5 |
|----|--|-------------------------------------|----------------------|---|
| 17 | $(x_1 - 4)^2 + (x_2 + 1)^2 + x_1 x_2$ | Хука-Дживса | наискорейшего спуска | БФШ |
| 18 | $10x_1^2 - 4x_1 x_2 + 7x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | циклического покоординатного спуска | Дробления шага | Квазиньтоновский Пауэлла |
| 19 | $(x_1 - 1)^2 + (x_2 + 3)^2 + 6x_1 x_2$ | Хука-Дживса | расходящегося ряда | ДФП |
| 20 | $(x_1 - 4)^2 + (x_2 + 1)^2 + x_1 x_2$ | циклического покоординатного спуска | наискорейшего спуска | сопряженных градиентов (Флетчера-Ривса) |
| 21 | $2x_1^2 + 2x_1 + 4x_2^2 + x_3^2 - 3x_3$ | Хука-Дживса | Дробления шага | БФШ |
| 22 | $10x_1^2 - 4x_1 x_2 + 7x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | циклического покоординатного спуска | расходящегося ряда | Квазиньтоновский Пауэлла |
| 23 | $(x_1 - 5)^2 + (x_2 - 4)^2 + x_1 x_3^2$ | Хука-Дживса | наискорейшего спуска | ДФП |
| 24 | $2x_1^2 + 2x_1 + 4x_2^2 + x_3^2 - 3x_3$ | циклического покоординатного спуска | Дробления шага | сопряженных градиентов (Флетчера-Ривса) |
| 25 | $(x_1 - 3)^2 + (x_2 - 2)^2 + x_1 x_3^2$ | Хука-Дживса | расходящегося ряда | БФШ |

Задание 1.6 Разработать программу для уточнения минимума целевой функции методами второго порядка. Варианты задания приведены в таблице 1.3.

1.3 Критерии масштабирования и останова

Материал раздела выполнен в соответствии с источниками: 18, 38.

Рассмотрим важные, для решения реальных оптимизационных задач на ЭВМ, вопросы:

1. Как решать оптимизационную задачу, в которой переменные различаются по величине на несколько порядков?

2. Каковы критерии на завершение итерационного процесса в реальных вычислительных процессах?

1.3.1 Масштабирование

Достаточно типичная ситуация при решении реальных задач, когда переменные сильно различаются по величине. К примеру, одна переменная x_1 , x_2 принимают значения из промежутков $[10^6 \div 10^7]$ рублей и $[10^{-3} \div 10^{-2}]$ тонн соответственно. Эти области характеризуют масштабы переменных. Если нам при реализации алгоритма требуется вычислить, допустим, евклидову норму разности $\|x^{(1)} - x^{(0)}\|$ двух векторов $x^{(1)} = (x_1^{(1)}, x_2^{(1)})$ и $x^{(0)} = (x_1^{(0)}, x_2^{(0)})$, то такая норма не будет чувствительной к достаточно большим вариациям переменной x_2 . Исправить ситуацию можно, изменив единицы измерения (осуществить масштабирование). В нашем примере переменную x_1 измерять в млн. рублей, а x_2 в кг. Таким образом, обе переменные будут измеряться в промежутке $[1, 10]$. Очевидно, масштабирование эквивалентно замене переменных

$$\tilde{x} = (\tilde{x}_1, \tilde{x}_2) = D_x x,$$

где D_x – диагональная матрица масштабирования [18], которая для нашего примера, имеет вид:

$$D_x = \begin{pmatrix} 10^{-6} & 0 \\ 0 & 10^3 \end{pmatrix}.$$

В общем случае преобразование переменных может осуществляться с помощью линейного преобразования:

$$\tilde{x} = T x,$$

где T – невырожденная матрица.

Подбор матрицы T должен осуществляться так, чтобы все компоненты

вектора x после преобразования принимали сопоставимые по величине значения, на практике все же предпочитают использовать матрицу D_x для построения которой необходимо указать характерные значения каждой из переменных x_i , тогда на диагонали матрицы D_x стоят $(x_i)^{-1}$. Не исключена ситуация, в которой по мере продвижения к точке экстремума придется проводить перемасштабирование (менять матрицу D) [16].

1.3.2 Критерии останова

Приведенные выше критерии завершения итерационной процедуры минимизации функции многих переменных, основанный на условии $\text{grad}f(x) \approx 0$. Хотя $\text{grad}f(x) = 0$ может встретиться в точке максимума или седловой точке, стратегия внесения возмущений в гессиан модели и стратегия локализации точки минимума делают фактически невозможной сходимость к точкам максимума или седловым точкам [18].

Поэтому, равенство $\text{grad}f(x) = 0$ будем рассматривать как необходимое и достаточное условие достижения в точке x локального минимума f .

Но на практике условие $\text{grad}f(x) \approx 0$ или эквивалентное соотношение $\|\text{grad}f(x)\| < \varepsilon$ часто бывает непригодным, поскольку зависит от масштаба «измерения» и x , и $f(x)$. Например, если область значений $f(x)$ много меньше (на порядки) по сравнению с ε , то вполне возможно, что любое x удовлетворяет приведенному выше неравенству (функция плавно меняется в малом диапазоне). Возможна и диаметрально противоположная ситуация, функция $f(x)$ принимает значения в большом диапазоне в окрестности точки минимума, тогда приведенный критерий завершения итерационного процесса будет практически нереализуем. Исходя из сказанного, рекомендуется использовать относительный градиент функции f в точке x , i -я компонента которого представляет собой отношение

относительной скорости изменения функции f к относительной скорости изменения компоненты x_i [18]:

$$(\text{relgrad}(x))_i = \lim_{\delta \rightarrow 0} \frac{\frac{f(x + \delta e_i) - f(x)}{f(x)}}{\frac{\delta}{x_i}} = \frac{(\text{grad}f(x))_i x_i}{f(x)},$$

и проверять неравенство:

$$\left| (\text{relgrad}(x^{(k)}))_i \right| \leq \varepsilon, \quad \forall i = \overline{1, n}.$$

В случае, когда x_i или f близки к нулю, рекомендуется использовать критерий, использующий типичные значения f и x :

$$\max_{i=1, \dots, n} \left| \frac{(\text{grad}f(x^{(k)}))_i \cdot \max\{|x^{(k)}_i|, \text{тип } x_i\}}{\max\{|f(x^{(k)})|, \text{тип } f\}} \right| \leq \varepsilon.$$

Подробнее о вопросах масштабирования, критериях останова для плохо масштабированных функций и тестировании алгоритмов приведено в источниках [18, 38].

1.3.3 Задания для практических (лабораторных) работ

Задание 1.7 Разработать программу для нахождения начального приближения к точке минимума целевой функции овражного типа градиентным методом. Варианты задания приведены в таблице 1.2.

Задание 1.8 Разработать программу для уточнения минимума целевой функции овражного типа методами второго порядка. Варианты задания приведены в таблице 1.2.

1.3.4 Задания для самостоятельной проработки и подготовке к контрольным работам

1. Составить алгоритм метода наискорейшего спуска для плохо масштабированной функции.

2. Составить алгоритм градиентного метода дробления шага для плохо масштабированной функции.

3. Составить алгоритм градиентного метода расходящегося ряда для плохо масштабированной функции.

4. Составить алгоритм квазиньютоновского метода для плохо масштабированной функции.

5. Составить алгоритм метода сопряженных градиентов для плохо масштабированной функции.

2 Нелинейное программирование

2.1 Постановка задачи нелинейного программирования в общей форме

Пусть требуется найти такой вектор $x = (x_1, \dots, x_n)$, который минимизирует функцию:

$$f(x) \rightarrow \min(\max), \quad (2.1)$$

при наличии ограничений:

$$D = \begin{cases} g_i(x) \leq 0 & i = 1, 2, \dots, m, \\ x \in X \subset \mathbb{R}^n, \end{cases} \quad (2.2)$$

$$(2.3)$$

где хотя бы одна из функций f и g_i , $i=1 \dots m$, нелинейна.

В область X (2.3) включаются ограничения специального вида, например, связанные с областью определения функций.

В соответствии с теоремой Вейерштрасса (достаточное условие существования экстремума) если $f(x)$ непрерывна на непустом замкнутом ограниченном множестве (компакте), тогда функция $f(x)$ достигает минимального (максимального) значения, по крайней мере, в одной из точек этого множества.

Решение задачи обозначим: $x^* = \arg \min_{x \in D} f(x)$ или для задачи на максимум:

$$x^* = \arg \max_{x \in D} f(x).$$

2.2 Аналитические методы решения задач нелинейного программирования

Материал раздела выполнен в соответствии с источниками: 4, 5, 6, 7, 13, 24.

2.2.1 Геометрическая интерпретация двумерной задачи нелинейного программирования

Для решения задачи нелинейного программирования геометрически:

1. Строят область допустимых решений задачи D . Если она пуста, то задача не имеет решения.

2. Строят гиперповерхность уровня целевой функции $f(x_1, \dots, x_n) = h = \text{const}$ и определяют направление убывания (или возрастания для задачи на максимум) целевой функции.

3. Находят точку области допустимых решений, через которую проходят линии наименьшего (или наибольшего для задачи на максимум) уровня, определяют в них значение функции (2.1).

Пример 2.1

$$f(x_1, x_2) = (x_1 - 4)^2 + (x_2 - 3)^2 \rightarrow \min(\max),$$

$$\begin{cases} 2x_1 + 3x_2 \geq 6 \\ 3x_1 - 2x_2 \leq 18 \\ -x_1 + 2x_2 \leq 8. \end{cases}$$

Построим область допустимых решений задачи. Для этого определены точки, через которые проходят прямые, задающие ограничения задачи.

Прямая I: $2x_1 + 3x_2 = 6$. Координаты точек пересечения с осями координат:

| | | |
|-------|---|---|
| x_1 | 0 | 3 |
| x_2 | 2 | 0 |

Прямая II: $3x_1 - 2x_2 = 18$. Координаты точек пересечения с осями координат:

| | | |
|-------|----|---|
| x_1 | 0 | 6 |
| x_2 | -9 | 0 |

Прямая III: $-x_1 + 2x_2 = 8$

| | | |
|-------|---|----|
| x_1 | 0 | -8 |
| x_2 | 4 | 0 |

Линии уровня целевой функции $f(x_1, x_2) = (x_1 - 4)^2 + (x_2 - 3)^2 = C$, где $C = \text{const}$ представляют собой окружности с центром в точке $(4, 3)$. Геометрическое решение приведено на рисунке 2.1.

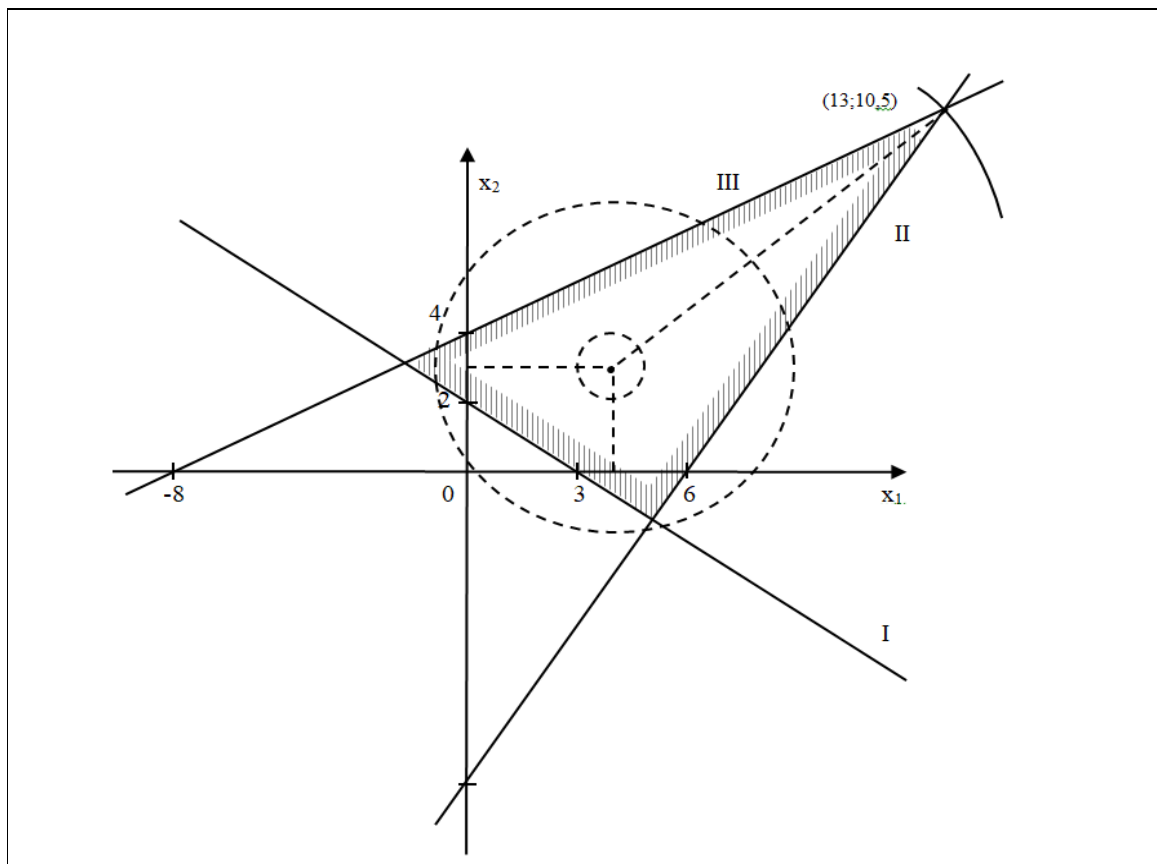


Рисунок 2.1 – Геометрическая интерпретация решения задачи нелинейного программирования

Для рассматриваемого примера точка минимума достигается в центре окружности, задающей линию уровня целевой функции $f(x_1, x_2) = f(4, 3) = 0$, т.е. $f_{\min} = 0$, $x_{\min} = (4, 3)$. Максимум достигается в точке пересечения прямых II и III, для нахождения указанной точки нужно решить систему уравнений вида:

$$\begin{cases} 3x_1 - 2x_2 \leq 18 \\ -x_1 + 2x_2 \leq 8 \end{cases} \Leftrightarrow \begin{cases} x_1 = 13, \\ x_2 = 10,5. \end{cases}$$

$$f_{\max} = f(13; 10,5) = 137,25; \quad x_{\max} = (13; 10,5).$$

Анализируя решение примера 2.1, можно сделать вывод, что решение задачи нелинейного программирования может находиться как внутри допустимой области, так и на ее границе, поэтому выполнять перебор всех точек допустимой области сложно.

Решение задачи нелинейного программирования затруднено наличием локальных и глобальных экстремумов целевой функции и сложностью области допустимых решений, поэтому в нелинейном программировании выделяют классы задач, для которых разрабатывают специфические способы решения.

2.2.2 Классическая задача на условный экстремум

Рассмотрим частный случай общей задачи нелинейного программирования (2.1)-(2.3), предполагая, что система ограничений содержит только уравнения, отсутствует условие неотрицательности переменных, и f , g_i – функции непрерывные вместе со своими производными:

$$f(x) \rightarrow \min(\max), \tag{2.4}$$

$$g_i(x) = 0, \quad i = 1, \dots, m, \tag{2.5}$$

где $x = (x_1, \dots, x_n)$.

Такая задача называется классической задачей условной нелинейной оптимизации.

2.2.3 Метод множителя Лагранжа

Чтобы найти решение задачи (2.4)-(2.5) вводят набор переменных: y_1, \dots, y_m , называемых множителями Лагранжа и составляют функцию Лагранжа:

$$L(x, y) = y_0 f(x) + \sum_{i=1}^m y_i g_i(x), \quad x \in \mathbb{R}^n, y \in \mathbb{R}^m,$$

при этом y_0 – константа, которая принимает значение $y_0 \geq 0$ для задачи на минимум, $y_0 \leq 0$ для задачи на максимум, $y_0 = 0$ для проверки стационарной точки.

ТЕОРЕМА 2.1 Необходимое условие оптимальности.

Пусть функции f и g_i непрерывны и дифференцируемы в некоторой окрестности $x^* \in \mathbb{R}^n$. Если точка x^* – решение задачи (2.4)-(2.5), то существует такая точка y_0 и вектор $y^* = (y_1^*, \dots, y_m^*)$ одновременно не равные нулю, т.е.

$\sum_{i=0}^m (y_i^*)^2 \neq 0$, что выполняются следующие условия:

1) условия стационарности обобщенной функции Лагранжа по x :

$$\frac{\partial L}{\partial x_j}(x^*, y^*) = 0, \quad j = 1, \dots, n, \quad (2.6)$$

$$\frac{\partial L}{\partial x_j}(x^*, y^*) = y_0 \frac{\partial f(x^*)}{\partial x_j} + \sum_{i=1}^m y_i \frac{\partial g_i(x^*)}{\partial x_j} = 0, \quad j = 1, \dots, n,$$

$$y_0 \frac{\partial f(x^*)}{\partial x_j} = - \sum_{i=1}^m y_i \frac{\partial g_i(x^*)}{\partial x_j}, \quad j = 1, \dots, n.$$

Т.е. частные производные $\frac{\partial f(x^*)}{\partial x_j}$ и $\frac{\partial g_i(x^*)}{\partial x_j}$, $i = 1, \dots, m$, линейно зависимы;

2) условия стационарности обобщенной функции Лагранжа по y приводят к соотношениям вида

$$\frac{\partial L}{\partial y_i}(x^*, y^*) = g_i(x) = 0, \quad i = 1, \dots, m,$$

и носят название условия допустимости решения:

$$g_i(x^*) = 0, \quad i = 1, \dots, m. \quad (2.7)$$

Если при этом градиенты $g'_i(x^*)$, $i = 1, \dots, m$ линейно независимы (это условие называют условием регулярности), то $y_0^* \neq 0$ и, следовательно, в этом случае множители Лагранжа можно задавать следующим образом:

$$y_i^* = \frac{y_i^*}{y_0^*},$$

следовательно, можно принять $y_0 = 1$ если задача на минимум и $y_0 = -1$ если задача на максимум. Такую точку x^* называют регулярной.

Далее для определенности будем рассматривать задачу на минимум:

$$f(x) \rightarrow \min, \quad (2.8)$$

$$D = \begin{cases} g_i(x) \leq 0, & i = 1, \dots, m, \\ x \in X \in \mathbb{R}^n. \end{cases} \quad (2.9)$$

ТЕОРЕМА 2.2 Достаточное условие локального минимума.

Пусть функции f и g_i дважды дифференцируемы в допустимой точке x^* . Если существует набор множителей Лагранжа $y_0 = 1, y = (y_1, \dots, y_m)$, такой что выполнены условия стационарности (2.6) и условия дополняющей нежесткости (2.7) и гессиан функции Лагранжа в точке x^* положительно определен

$$L''_{xx}(x, y^*) \Big|_{x=x^*} > 0, \quad (2.10)$$

тогда x^* – локальный минимум задачи (2.8)-(2.9).

ТЕОРЕМА 2.3 Необходимые условия 2-го порядка.

Пусть функции f и g_i дважды непрерывно дифференцируемы в точке x^* , причем выполнены условия регулярности (производные f и g_i линейно независимы). Если x^* – локальный минимум задачи (2.8)-(2.9), то существует набор множителей $y_0 = 1, y = (y_1, \dots, y_m)$, таких что

$$\frac{\partial L(x, y)}{\partial x_i} \Big|_{\substack{x=x^* \\ y=y^*}} = 0 \text{ и } L''_{xx}(x, y^*) \Big|_{x=x^*} \geq 0. \quad (2.11)$$

Пример 2.2

Дана задача:

$$f(x_1, x_2) = x_1^2 + x_2^2 \rightarrow \min, \\ x_1 + x_2 = 5.$$

Геометрическое решение приведено на рисунке 2.2.

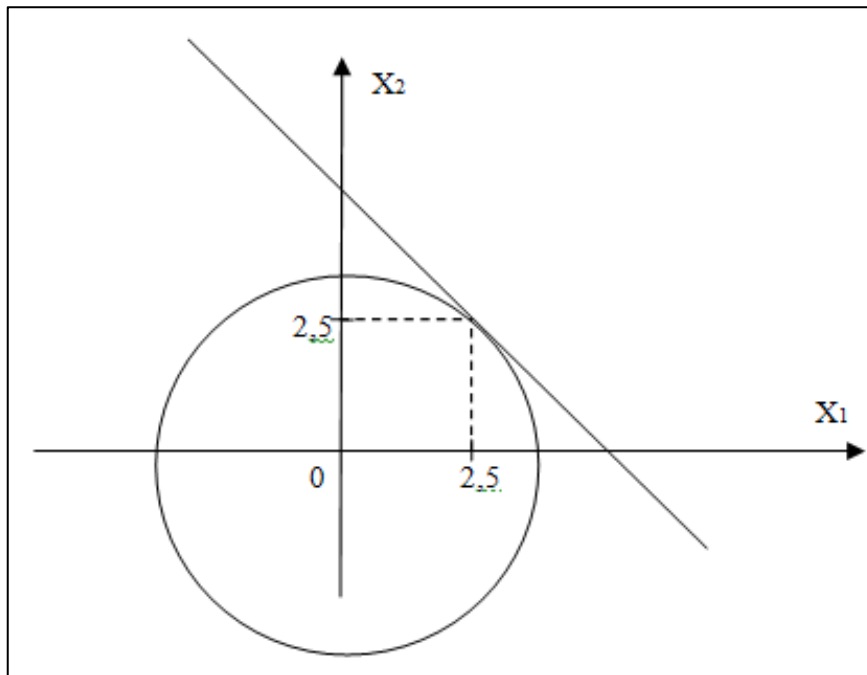


Рисунок 2.2 – Геометрическая интерпретация решения задачи нелинейного программирования для примера 2.2

Функция Лагранжа имеет вид $L(x_1, x_2, y_1) = y_0(x_1^2 + x_2^2) + y_1(x_1 + x_2 - 5)$. Так как область допустимых решений задает одно уравнение, следовательно, условие регулярности выполняется и в качестве y_0 можно принять $y_0 = 1$.

$$L(x_1, x_2, y_1) = (x_1^2 + x_2^2) + y_1(x_1 + x_2 - 5).$$

Запишем условия стационарности и дополняющей нежесткости:

$$\begin{cases} \frac{\partial L(x, y_1)}{\partial x_1} = 2x_1 + y_1 = 0 \\ \frac{\partial L(x, y_1)}{\partial x_2} = 2x_2 + y_1 = 0 \\ \frac{\partial L(x, y_1)}{\partial y_1} = x_1 + x_2 - 5 = 0 \end{cases} \Leftrightarrow \begin{cases} x_1 = 2,5 \\ x_2 = 2,5 \\ y_1 = -5 \end{cases}$$

Проверим достаточное условие

$$\begin{pmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 \partial x_2} \\ \frac{\partial^2 L}{\partial x_1 \partial x_2} & \frac{\partial^2 L}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

Гессеан в точке (x_1, x_2) положительно определен, следовательно, точка $(2,5;2,5)$ – точка минимума исходной задачи.

Алгоритм решения классической задачи нелинейного программирования методом множителей Лагранжа:

1. Составить функцию Лагранжа $L(x, y) = y_0 f(x) + \sum_{i=1}^m y_i g_i(x)$

2. Найти стационарные точки при $y_0 = 0$ и при $y_0 \neq 0$:

$$\begin{cases} \left. \frac{\partial L(x, y)}{\partial x_j} \right|_{\substack{x=x^* \\ y=y^*}} = 0, j=1, \dots, n, \\ g_i(x^*) = 0, i=1, \dots, m. \end{cases}$$

3. Классифицировать стационарные точки на основании достаточного условия.

4. Проверить условия:

Если $L''_{xx}(x, y^*) \Big|_{x=x^*} > 0$, то в точке x^* условный локальный минимум (или для задачи на максимум: если $L''_{xx}(x, y^*) \Big|_{x=x^*} < 0$, то в точке x^* условный локальный максимум),

если условия не выполнены, то: проверяют выполнение необходимого условия 2-го порядка $L''_{xx}(x, y^*) \Big|_{x=x^*} \geq 0$ (для задачи на максимум $L''_{xx}(x, y^*) \Big|_{x=x^*} \leq 0$), если они выполняются, то необходимы дополнительное исследование, если не выполняются, то в точке нет экстремума.

2.2.4 Конечномерная задача нелинейного программирования с ограничениями в виде равенств и неравенств

Конечномерной задачей нелинейного программирования с ограничениями в виде равенств и неравенств называется задача вида:

$$f(x) \rightarrow \text{extr} , \quad (2.12)$$

$$g_i(x) \leq 0, \quad i = 1, \dots, k, \quad (2.13)$$

$$g_i(x) = 0, \quad i = k + 1, \dots, m, \quad (2.14)$$

$$x = (x_1, \dots, x_n).$$

Алгоритм решения конечномерной задачей нелинейного программирования с ограничениями в виде равенств и неравенств методом множителей Лагранжа.

1. Составить функцию Лагранжа $L(x, y) = y_0 f(x) + \sum_{i=1}^m y_i g_i(x)$.

2. Выписать необходимые условия локального экстремума:

- условие стационарности по x (2.15):

$$\left. \frac{\partial L(x, y)}{\partial x_j} \right|_{\substack{x=x^* \\ y=y^*}} = 0, \quad j = 1, \dots, n ; \quad (2.15)$$

- условия стационарности обобщенной функции Лагранжа по y :

$$\left. \frac{\partial L(x, y)}{\partial y_i} \right|_{\substack{x=x^* \\ y=y^*}} = g_i(x^*) = 0, \quad i = k + 1, \dots, m ,$$

- приводящие к условиям допустимости решения (2.16):

$$g_i(x^*) = 0, i = k + 1, \dots, m; \quad (2.16)$$

- условие дополняющей нежесткости (2.17):

$$y_i^* g_i(x^*) = 0, i = 1, \dots, k; \quad (2.17)$$

- условие неотрицательности (2.18)

$$y_i^* \geq 0, i = 1, \dots, k; \quad (2.18)$$

кроме того $y_0^* \geq 0$ для задачи на минимум, $y_0^* \leq 0$ для задачи на максимум.

3. Найти стационарные точки, т.е. допустимые решения п.2, для которых не все множители Лагранжа равны нулю. При этом отдельно рассмотреть случай $y_0^* = 0$ и $y_0^* \neq 0$.

4. Определить какая из найденных стационарных точек является локальным минимумом или локальным максимумом.

5. Выяснить есть ли глобальный минимум или максимум.

ТЕОРЕМА 2.4 Необходимые условия локального минимума.

Пусть x^* – точка локального минимума задачи (2.12)-(2.14), и пусть функции f и g_i непрерывно-дифференцируемые, тогда существует ненулевой набор множителей Лагранжа $y_0^*, y_1^*, \dots, y_m^*$ (т.е. $\sum_{i=0}^m (y_i^*)^2 \neq 0$) такой, что для функции

Лагранжа выполняются условия:

- 1) стационарности по x (2.15);
- 2) условие дополняющей нежесткости (2.16);
- 3) условие допустимости решения (2.17);
- 4) условие неотрицательности (2.18).

Пример 2.3

Дана задача:

$$F(x_1, x_2) = 29x_1^2 - 54x_1x_2 + 26x_2^2 \rightarrow \min ,$$
$$\begin{cases} 3x_1 - 4x_2 = 1 \\ 2x_1 - x_2 \leq 0. \end{cases}$$

Составим функцию Лагранжа:

$$L(x, y) = y_0(29x_1^2 - 54x_1x_2 + 26x_2^2) + y_1(3x_1 - 4x_2 - 1) + y_2(2x_1 - x_2).$$

Запишем условия (2.15)-(2.18):

$$\begin{cases} \frac{\partial L}{\partial x_1} = y_0(58x_1 - 54x_2) + 3y_1 + 2y_2 = 0 \\ \frac{\partial L}{\partial x_2} = y_0(-54x_1 + 52x_2) - 4y_1 - y_2 = 0 \\ 3x_1 - 4x_2 - 1 = 0 \\ y_2(2x_1 - x_2) = 0 \\ y_2 \geq 0. \end{cases}$$

Рассмотрим случаи $y_0 = 0$ и $y_0 = 1$.

1. Пусть $y_0 = 0$, тогда:

$$\begin{cases} 3y_1 + 2y_2 = 0 \\ -4y_1 - y_2 = 0 \\ 3x_1 - 4x_2 - 1 = 0 \\ y_2(2x_1 - x_2) = 0 \end{cases}$$

Возможны два случая $y_2 = 0$ и $(2x_1 - x_2) = 0$.

Пусть $(2x_1 - x_2) = 0$, тогда

$$\begin{cases} 3y_1 + 2y_2 = 0 \\ -4y_1 - y_2 = 0 \\ 3x_1 - 4x_2 - 1 = 0 \\ 2x_1 - x_2 = 0 \end{cases} \begin{cases} y_1 = 0 \\ y_2 = 0 \\ x_1 = -0,2 \\ x_2 = -0,4 \end{cases},$$

решение системы не удовлетворяет условию $\sum_{i=1}^m (y_i^*)^2 \neq 0$.

Пусть $y_2 = 0$, тогда

$$\begin{cases} 3y_1 + 2y_2 = 0 \\ -4y_1 - y_2 = 0 \\ 3x_1 - 4x_2 - 1 = 0 \\ y_2 = 0 \end{cases} \begin{cases} y_1 = 0 \\ y_2 = 0 \\ x_1 = -0,2 \\ x_2 = -0,4 \end{cases}$$

решение системы не удовлетворяет условию $\sum_{i=1}^m (y_i^*)^2 \neq 0$.

Таким образом при $y_0=0$ стационарных точек нет.

2. Рассмотрим случаи $y_0 = 1$:

$$\begin{cases} 58x_1 - 54x_2 + 3y_1 + 2y_2 = 0 \\ -54x_1 + 52x_2 - 4y_1 - y_2 = 0 \\ y_2(2x_1 - x_2) = 0, \quad y_2 \geq 0 \\ 3x_1 - 4x_2 = 1, \quad 2x_1 - x_2 \leq 0 \end{cases}$$

складываем уравнения 1 и 2: $4x_1 - 2x_2 - y_1 + y_2 = 0 \Rightarrow x_2 = 2x_1 - \frac{y_1}{2} + \frac{y_2}{2}$,

подставим во 2-ое уравнение: $-54x_1 + 52\left(2x_1 - \frac{y_1}{2} + \frac{y_2}{2}\right) - 4y_1 - y_2 = 0$.

После преобразований получим:

$$x_1 = \frac{3}{5}y_1 - \frac{1}{2}y_2,$$

$$x_2 = \frac{7}{10}y_1 - \frac{y_2}{2}.$$

Из условия дополняющей нежесткости либо $y_2 = 0$, либо $2x_1 - x_2 = 0$.

1) Положим $y_2 = 0$, тогда:

$$\begin{cases} x_1 = \frac{3}{5}y_1 \\ x_2 = \frac{7}{10}y_1 \end{cases}$$

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

$$\begin{cases} x_1 = -\frac{3}{5} \\ x_2 = -\frac{7}{10} \\ y_1 = -1 \end{cases}$$

Проверяем выполняется ли условие: $2x_1 - x_2 \leq 0$:

$-6/5 + 7/10 \leq 0$, следовательно $(-3/5; -7/10)$ удовлетворяет необходимому условию локального минимума.

2) Положим $2x_1 - x_2 = 0$:

$$\begin{cases} 2x_1 - x_2 = 0 \\ 3x_1 - 4x_2 = 1 \end{cases} \begin{cases} x_1 = -\frac{1}{5} \\ x_2 = -\frac{2}{5} \end{cases} \begin{cases} \frac{3}{5}y_1 - \frac{1}{2}y_2 = -\frac{1}{5} \\ \frac{7}{10}y_1 - \frac{y_2}{2} = -\frac{2}{5} \end{cases} \begin{cases} y_2 = -2 \\ y_1 = -2 \end{cases}.$$

Т.к. $y_2 < 0$, т.е. не выполняется условие неотрицательности (2.18), следовательно, не удовлетворяют условию локального минимума.

Исследуем найденную точку $(-3/5; -7/10)$, используя определение локального минимума.

Проверим знак выражения Δf

$$\Delta f = f(x_1^* + \Delta x_1, x_2^* + \Delta x_2) - f(x_1^*, x_2^*),$$

$$\begin{aligned} \Delta f &= 29(x_1^* + \Delta x_1)^2 - 54(x_1^* + \Delta x_1)(x_2^* + \Delta x_2) + 26(x_2^* + \Delta x_2) - 29(x_1^*)^2 + 54(x_1^*)(x_2^*) - 26(x_2^*) = \\ &= 58x_1^* \Delta x_1 + 29\Delta x_1^2 - 54x_2^* \Delta x_1 - 54x_1^* \Delta x_2 - 54\Delta x_1 \Delta x_2 + 52x_2^* \Delta x_2 + 26(\Delta x_2)^2. \end{aligned}$$

$$1) x_1^* = -\frac{3}{5}x_2^* = -\frac{7}{10},$$

$$\Delta f = 3\Delta x_1 - 4\Delta x_2 + 29(\Delta x_1)^2 + 26(\Delta x_2)^2 - 54\Delta x_1 \Delta x_2.$$

Т.к. точка $(x_1^* + \Delta x_1, x_2^* + \Delta x_2)$ должна быть допустимой, то необходимо выполнение следующего условия:

$$\begin{cases} 3\left(-\frac{3}{5} + \Delta x_1\right) - 4\left(-\frac{7}{10} + \Delta x_2\right) = 1 \\ 2\left(-\frac{3}{5} + \Delta x_1\right) - \left(-\frac{7}{10} + \Delta x_2\right) \leq 0 \end{cases} \Rightarrow \begin{cases} \Delta x_2 = \frac{3}{4}\Delta x_1, \text{ (a)} \\ -\frac{1}{2} + 2\Delta x_1 - \Delta x_2 \leq 0, \text{ (b)} \end{cases}$$

Условие (b) выполняется при малых $\Delta x_1, \Delta x_2$. Условие (a) подставим в Δf :

$$\Delta f = 3\Delta x_1 - 4 \cdot \frac{3}{4}\Delta x_1 + 29(\Delta x_1^2) + 26\left(\frac{3}{4}\Delta x_1\right)^2 - 54\Delta x_1 \frac{3}{4}\Delta x_1 = \frac{25}{8}(\Delta x_1)^2 \geq 0 \Rightarrow$$

точка $(-3/5; -7/10)$ является точкой локального минимума.

Учитывая, что $F(x_1, x_2)$ является выпуклой функцией и допустимое множество выпуклое, следовательно, точка локального минимума $(-3/5; -7/10)$ является точкой глобального минимума.

2.2.5 Задача выпуклого нелинейного программирования

Пусть дана задача нелинейного программирования:

$$f(x) \rightarrow \min, \quad (2.19)$$

$$D: \begin{cases} g_i(x) \leq 0, \quad i = 1, \dots, m, \\ x \in X \subset \mathbb{R}^n. \end{cases} \quad (2.20)$$

$$(2.21)$$

Задача нелинейного программирования является задачей выпуклого нелинейного программирования, если целевая функция $f(x)$ выпуклая и множество допустимых решений задачи D выпукло (приложение Б).

УТВЕРЖДЕНИЕ. 2.1 Если функция $g_i(x)$ выпуклая, то ограничение $g_i(x) \leq 0$ задает выпуклое множество.

Свойства задачи выпуклого нелинейного программирования:

1) если $g_i(x)$ выпуклые для $i = \overline{1, m}$, то множество (2.20) выпуклое, если X также выпуклое множество, то множество D – выпуклое;

2) любое локальное решение задачи выпуклого нелинейного программирования является ее глобальным решением.

2.2.6 Общая теория нелинейного программирования

Пусть дана задача (2.19)-(2.21) и функция Лагранжа имеет вид:

$$L(x, y) = f(x) + \sum_{i=1}^m y_i g_i(x). \quad (2.22)$$

ОПРЕДЕЛЕНИЕ. Пара векторов (x^0, y^0) называется седловой точкой функции Лагранжа в области $x \in X, y \geq 0$, если выполняется неравенство:

$$L(x^0, y) \leq L(x^0, y^0) \leq L(x, y^0), \quad \forall x \in X, y \geq 0. \quad (2.23)$$

ТЕОРЕМА 2.5

Пусть (x^0, y^0) – седловая точка функции Лагранжа в области $x \in X, y \geq 0$, тогда точка x^0 является решением задачи (2.19)-(2.21), причем справедливо правило дополняющей нежесткости в седловой точке:

$$\sum_{i=1}^m y_i^0 g_i(x^0) = 0. \quad (2.24)$$

ДОКАЗАТЕЛЬСТВО:

По определению седловой точки должно выполняться соотношение:

$$L(x^0, y) \leq L(x^0, y^0) \leq L(x, y^0),$$

$$f(x^0) + \sum_{i=1}^m y_i g_i(x^0) \leq f(x^0) + \sum_{i=1}^m y_i^0 g_i(x^0) \leq f(x) + \sum_{i=1}^m y_i^0 g_i(x). \quad (2.25)$$

Нужно показать, что:

- 1) $x^0 \in D$;
- 2) $x^0 - \min f(x): f(x^0) \leq f(x), \quad \forall x \in D$.

Из левой части соотношения (2.25) следует:

$$\sum_{i=1}^m y_i g_i(x^0) \leq \sum_{i=1}^m y_i^0 g_i(x^0), \quad \forall y \geq 0, \quad (2.26)$$

следовательно, $g_i(x^0) \leq 0$, т.к. если бы это было не так, то для некоторых $i=k$

выбором достаточно большого y_k и остальных $y_i = 0, \forall i \neq k$ левую часть неравенства (2.26) можно сделать сколь угодно большой, что приводит к противоречию с фиксированной правой частью. Поэтому неравенство $g_i(x^0) \leq 0, i = 1, \dots, m$, верны, т.е. выполняются условия (2.20): $x^0 \in D$.

Далее положим в неравенстве (2.26) все $y_i = 0$, тогда $\sum_{i=1}^m y_i^0 g_i(x^0) \geq 0$, в то же

время при $y_i \geq 0$ и выполнении условия (2.20): $g_i(x) \leq 0, i = 1, \dots, m$, справедливо

условие $\sum_{i=1}^m y_i^0 g_i(x) \leq 0$, и, в частности $\sum_{i=1}^m y_i^0 g_i(x^0) \leq 0$, в итоге получаем:

$$\left. \begin{array}{l} 1) \sum_{i=1}^m y_i^0 g_i(x^0) \geq 0 \\ 2) \sum_{i=1}^m y_i^0 g_i(x^0) \leq 0 \end{array} \right\} \Leftrightarrow \sum_{i=1}^m y_i^0 g_i(x^0) = 0,$$

следовательно, справедливо правило дополняющей нежесткости (2.24).

Далее из правой части (2.25) следует :

$$f(x^0) \leq f(x) + \sum_{i=1}^m y_i^0 g_i(x), \quad \forall x \in X,$$

так как $\sum_{i=1}^m y_i^0 g_i(x) \leq 0$, тогда

$$f(x^0) \leq f(x) \quad \forall x \in D \Rightarrow x^0 = \arg \min_{x \in D} f(x).$$

Ч.Т.Д.

ЗАМЕЧАНИЕ. Утверждение обратное утверждению теоремы справедливо лишь для задачи выпуклого нелинейного программирования в предположении, что ограничения (2.20)-(2.21) удовлетворяют условию регулярности Слейтера.

Условие регулярности Слейтера: говорят, что функции $g_i(x)$, $i=1,..m$ задающие ограничения в задаче (2.19)-(2.21) удовлетворяют условиям регулярности Слейтера, если существует такая точка $x^* \in D$, что $g_i(x^*) < 0$, $i=1,..m$, т.е. x^* является внутренней точкой относительно ограничений $g_i(x)$, $i=1,..m$. Это условие эквивалентно линейной независимости градиентов ограничений $\nabla g_i(x)$ соответствующих активным ограничениям.

Ограничение называют активным, если оно в точке x выполняется как равенство, т.е. $g_i(x^*) = 0$.

УТВЕРЖДЕНИЕ. На экстремальные свойства целевой функции $f(x)$ в некоторой достаточно малой окрестности точки x^* влияют лишь те ограничения $g_i(x^*) \leq 0$, для которых в этой точке ограничения выполняются как равенства, т.е. активные ограничения. Обозначим $I = \{i : g_i(x^*) = 0, i = 1,..m\}$ – множество номеров активных в точке x^* ограничений.

ТЕОРЕМА 2.6 (Теорема Куна-Такера)

Пусть $f(x)$ и $g_i(x)$ выпуклые функции, множество X – выпукло и выполняется условие регулярности Слейтера. Точка x^0 является решением задач (2.19)-(2.21) тогда и только тогда, когда существует такой вектор $y^0 \geq 0$, что пара векторов (x^0, y^0) является седловой точкой функции Лагранжа.

ДОКАЗАТЕЛЬСТВО:

1. Достаточность – показано в рамках доказательства предыдущей теоремы.
2. Необходимость.

Пусть x^0 решение задачи (2.19)-(2.21), т.е. $f(x^0) \leq f(x)$, $\forall x \in D$. Покажем, что существует $y^0 \geq 0$, такой, что (x^0, y^0) – седловая точка функции Лагранжа.

Определим два множества U и Z точек $m+1$ -мерного пространства:

$$U = \{u_0, u_1, \dots, u_m\} : f(x) \leq u_0,$$

$$g_i(x) \leq u_i, \quad i = 1, ..m,$$

$$Z = \{z_0, z_1, \dots, z_m\} : g(x^0) > z_0,$$

$$z_i \leq 0, \quad i = \overline{1, m}$$

Множества U и Z – выпуклые и не пересекаются, т.к. учитывая, что x^0 – оптимально, то $\forall x \in X$ выполнено $f(x^0) \leq f(x)$, $f(x) \leq u_0$, $f(x^0) > z_0$, следовательно U и Z не пересекаются.

Тогда по теореме отделимости 3 (приложение Г) существует $c = (c_0, \dots, c_m)$, $c \neq 0$ такой что:

$$\min_{u \in U} \langle c, u \rangle \geq \max_{z \in Z} \langle c, z \rangle, \quad \forall u \in U, z \in Z. \quad (2.27)$$

Неравенство (2.27) справедливо и тогда, когда точка z принадлежит границе множества Z , поэтому выберем в качестве $u_0 = f(x)$, $u_i = g_i(x)$, $z_0 = f(x^0)$, $z_i = 0$. Тогда, учитывая (2.27) имеет место неравенство:

$$c_0 f(x^0) \leq c_0 f(x) + \sum_{i=1}^m c_i g_i(x), \quad \forall x \in X. \quad (2.28)$$

Покажем, что $c_0 > 0$. Предположим противоположное. Пусть $c_0 = 0$, тогда: $\sum_{i=1}^m c_i g_i(x) \geq 0$ из неравенства (2.28). Так как $c_i \neq 0$, а для $x \in D$, $g_i(x) \leq 0$, то при $c_i < 0$ равенство $g_i(x) = 0$ будет выполняться $\forall x \in D$, что противоречит условию регулярности, следовательно $c_0 > 0$.

Возьмем в качестве $y^0 = (y_1^0, \dots, y_m^0)$ такую точку, что: $y_i^0 = \frac{c_i}{c_0}$, $y_i^0 \geq 0$.

Поделим неравенство (2.28) на c_0 :

$$f(x^0) \leq f(x) + \sum_{i=1}^m y_i^0 g_i(x), \quad \forall x \in X. \quad (2.29)$$

Пусть $x=x^0$, тогда $f(x^0) \leq f(x^0) + \sum_{i=1}^m y_i^0 g_i(x^0)$, следовательно $\sum_{i=1}^m y_i^0 g_i(x^0) \geq 0$.

С другой стороны $g_i(x^0) \leq 0$ и $y_i^0 \geq 0$ поэтому $\sum_{i=1}^m y_i^0 g_i(x^0) \leq 0$, следовательно:

$$\sum_{i=1}^m y_i^0 g_i(x^0) = 0.$$

Учитывая $\sum_{i=1}^m y_i^0 g_i(x^0) = 0$ и (2.29) делаем вывод, что (x^0, y^0) седловая точка

функции Лагранжа.

ч.т.д.

ЗАМЕЧАНИЕ. Поиск седловой точки функции Лагранжа можно интерпретировать как игру двух лиц, где $L(x, y)$ платежная функция 1-ого игрока 2-ому $x \in X, y \geq 0$.

Из приведенных выше теорем следует правило множителей Лагранжа.

ТЕОРЕМА 2.7 Правило множителей Лагранжа. Теорема Куна-Такера.

Рассмотрим задачу

$$f(x) \rightarrow \min, \tag{2.30}$$

$$D: g_i(x) \leq 0, i = 1, \dots, m. \tag{2.31}$$

Чтобы точка x^* была решением задачи (2.30)-(2.31) достаточно, а в случае если (2.30)-(2.31) задача выпуклого нелинейного программирования, и область D удовлетворяет условиям Слейтера, необходимо, чтобы нашлись такие числа $y_i^* \geq 0$

$i = \overline{1, m}$, что выполнены условия $f'(x^*) + \sum_{i=1}^m y_i^* g_i'(x^*) = 0$, и при этом выполнено

условие дополняющей нежесткости $\sum_{i=1}^m y_i^* g_i(x^*) = 0$.

ТЕОРЕМА 2.8 Обобщенное правило множителей Лагранжа. Теорема Куна-Такера.

Рассмотрим задачу

$$f(x) \rightarrow \min, \quad (2.32)$$

$$D: \begin{cases} g_i(x) \leq 0, i = 1, \dots, m, \\ h_j(x) = 0, j = 1, \dots, l. \end{cases} \quad (2.33)$$

Пусть x^* – локальное оптимальное решение задачи. $I = \{i : g_i(x^*) = 0, i = 1, \dots, m\}$. Предположим, что $f, g_i, i \in I, h_j, j = 1, \dots, l$ дифференцируемы в точке x^* . Пусть выполняются условия регулярности Слейтера, тогда x^* является оптимальной точкой, т.е. существуют такие точки $y_i^* \geq 0 \quad \forall i \in I$ и $v_j^*, j = 1, \dots, l$, что выполняется условие:

$$f'(x^*) + \sum_{i \in I} y_i^* g_i'(x^*) + \sum_{j=1}^l v_j^* h_j'(x^*) = 0, \quad (2.34)$$

и выполняются условия дополняющей нежесткости:

$$\sum_{i=1}^m y_i^* g_i(x^*) = 0 \quad \text{и} \quad \sum_{j=1}^l v_j^* h_j(x^*) = 0. \quad (2.35)$$

2.2.7 Условия существования седловой точки

Пусть существует $L(x, y)$, выпуклая по переменным x и вогнутая по переменным y для $\forall x \geq 0, y \geq 0$, кроме того $L(x, y)$ – непрерывно дифференцируемая по x и по y функция.

ТЕОРЕМА 2.9

Чтобы пара векторов (x^0, y^0) была седловой точкой функции $L(x, y)$ в области $x \geq 0, y \geq 0$, необходимо и достаточно, чтобы выполнялись следующие условия:

$$I. \frac{\partial L(x^0, y^0)}{\partial x_j} \geq 0 \quad j = \overline{1, n};$$

$$\text{Обозначим: } \frac{\partial L^0}{\partial x_j} = \frac{\partial L(x^0, y^0)}{\partial x_j};$$

$$II. \langle x^0, \frac{\partial L^0}{\partial x} \rangle = 0 \sim x_j^0 \frac{\partial L^0}{\partial x_j} = 0, \quad j = \overline{1, n};$$

$$III. x^0 \geq 0 \sim x_j^0 \geq 0 \quad j = \overline{1, n};$$

$$IV. \frac{\partial L^0}{\partial y_i} \leq 0 \quad j = \overline{1, m};$$

$$V. \langle y^0, \frac{\partial L^0}{\partial y} \rangle = 0 \sim y_i^0 \frac{\partial L^0}{\partial y_i} = 0 \quad j = \overline{1, m};$$

$$VI. y^0 \geq 0 \sim y_i^0 \geq 0 \quad j = \overline{1, m}.$$

ЗАМЕЧАНИЕ. Условия (I)-(III) являются необходимыми условиями минимума (локального минимума) функции одной переменной.

2.2.8 Теория двойственности нелинейного программирования

Пусть дана задача

$$f(x) \rightarrow \min, \quad (2.40)$$

$$D: \begin{cases} g_i(x) \leq 0, \quad i = 1, \dots, m, \\ x \in X \subset \mathbb{R}^n, \end{cases} \quad (2.41)$$

$$(2.42)$$

и функция Лагранжа

$$L(x, y) = f(x) + \sum_{i=1}^m y_i g_i(x). \quad (2.43)$$

Рассмотрим две задачи:

I. Прямая

$$F(x) = \max_{y \geq 0} L(x, y) \rightarrow \min, \quad (2.44)$$
$$x \in X.$$

II. Двойственная

$$G(x) = \min_{x \in X} L(x, y) \rightarrow \max, \quad (2.45)$$
$$y \geq 0.$$

УТВЕРЖДЕНИЕ 2.2

Задача I эквивалентна задаче (2.40)-(2.42):

$$F(x) = \max_{y \geq 0} L(x, y) = \begin{cases} f(x), & x \in D \\ \infty, & x \notin D \end{cases},$$

$\min_{x \in X} F(x) \in D$ и совпадает с $\min_{x \in D} f(x)$, т.е. имеем

$$\min_{x \in X} \max_{y \geq 0} L(x, y) = \min_{x \in D} f(x).$$

УТВЕРЖДЕНИЕ 2.3

$$G(y) \leq F(x) \quad \forall x \in X, \quad y \geq 0,$$

$$\min_{x \in X} L(x, y) \leq L(x, y), \quad \forall y \geq 0 \Rightarrow \min_{x \in X} L(x, y) \leq \max_{y \geq 0} L(x, y).$$

УТВЕРЖДЕНИЕ 2.4

Пусть функция Лагранжа для задачи (2.40)-(2.42) имеет седловую точку в

области $x \in X, y \geq 0$. Тогда решение задач I и II существует и их значения в оптимальной точке равны: $\max_{y \geq 0} G(y) = \min_{x \in X} F(x)$.

СЛЕДСТВИЕ. Покажем, что задача линейного программирования является частным случаем задачи нелинейного программирования:

| | |
|--|---|
| <p>I.</p> $\sum_{j=1}^n c_j x_j \rightarrow \min,$ $\sum_{j=1}^n a_{ij} x_j \geq b_i \quad i = \overline{1, m},$ $x \geq 0.$ | <p>II.</p> $\sum_{i=1}^m b_i y_i \rightarrow \max,$ $\sum_{i=1}^m a_{ij} y_i \leq c_j \quad j = \overline{1, n},$ $y \geq 0.$ |
|--|---|

Для первой задачи функция Лагранжа:

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

$$G(y) = \min_{x \in X} L(x, y) = \min_{x \in X} \left(\sum_{j=1}^n c_j x_j + \sum_{i=1}^m y_i \left(b_i - \sum_{j=1}^n a_{ij} x_j \right) \right) =$$

$$= \min_{x \in X} \left[\sum_{j=1}^n x_j \left(c_j - \sum_{i=1}^m a_{ij} y_i \right) + \sum_{i=1}^m y_i b_i \right] =$$

$$= \begin{cases} \sum_{i=1}^m y_i b_i \rightarrow \max, & \text{если } \sum_{i=1}^m a_{ij} y_i \leq c_j, j = 1, \dots, n; \\ -\infty, & \text{иначе.} \end{cases}$$

Таким образом, мы пришли к задаче II.

2.2.9 Вопросы для самопроверки

1. Приведите общую постановку задачи НЛП.
2. Приведите примеры задач НЛП
3. Опишите геометрический метод решения задачи НЛП.
4. Приведите постановку классической задачи на условный экстремум, опишите способ ее решение методом множителей Лагранжа.
5. Приведите общую постановку задачи условной нелинейной оптимизации, опишите способ ее решения методом множителей Лагранжа.
6. Приведите определение седловой точки функции Лагранжа, ее свойства.
7. Приведите теорему Куна-Таккера для задачи выпуклого НЛП.
8. Охарактеризуйте понятие двойственной задачи НЛП.
9. Приведите условия существования седловой точки.

2.2.10 Задания для практических (лабораторных) работ

Задание 2.1 Геометрически найти решение задачи нелинейного программирования

$$f(x) = c_1x_1 + c_2x_2 \rightarrow \min(\max),$$

$$f(x) = (x_1 - a_1)^2 + (x_2 - a_2)^2 \leq b$$

Варианты к заданию 2.1 приведены в таблицах 2.1, 2.2.

Решить задачу с использованием ППП MathCad или MS Excel, сравнить полученные решения.

Таблица 2.1 – Варианты 1-10 к заданию 2.1

| вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------|----|----|----|---|---|---|----|----|----|----|
| c ₁ | 2 | -3 | -1 | 1 | 2 | 3 | -3 | -2 | 4 | 5 |
| c ₂ | -3 | 2 | 1 | 1 | 3 | 2 | -2 | -3 | 5 | -4 |
| a ₁ | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 4 | -3 | 1 |
| a ₂ | 3 | 4 | 1 | 2 | 1 | 2 | 3 | -3 | 2 | -2 |
| b | 3 | 3 | 4 | 4 | 2 | 2 | 5 | 5 | 3 | 3 |

Таблица 2.2 – Варианты 11-20 к заданию 2.1

| вариант | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|----------------|----|----|----|----|----|----|----|----|----|----|
| c ₁ | 2 | -5 | -4 | -1 | -2 | 3 | 3 | -5 | 3 | 2 |
| c ₂ | 5 | 2 | 3 | -1 | 3 | -2 | 2 | -2 | 6 | -5 |
| a ₁ | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 4 | -3 | 1 |
| a ₂ | 3 | 4 | 1 | 2 | 1 | 2 | 3 | -3 | 2 | -2 |
| b | 3 | 3 | 4 | 4 | 2 | 2 | 5 | 5 | 3 | 3 |

Задание 2.2 Геометрически найти решение задачи нелинейного программирования

$$f(x) = (x_1 + c_1)^2 + (x_2 + c_2)^2 \rightarrow \min(\max),$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \leq b_1 \\ a_{21}x_1 + a_{22}x_2 \leq b_2 \end{cases}$$

$$x_1, x_2 \geq 0$$

Варианты к заданию 2.2 приведены в таблице 2.3, 2.4.

Таблица 2.3 – Варианты 1-10 к заданию 2.2

| вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----------------|----|----|----|----|----|----|----|----|----|----|
| c ₁ | -2 | -3 | -1 | -1 | -2 | -3 | -3 | -2 | -4 | -5 |
| c ₂ | -3 | -2 | -1 | -1 | -4 | -2 | -2 | -3 | -5 | -4 |
| a ₁₁ | 2 | 5 | 2 | 2 | 5 | -5 | -5 | 6 | 6 | 6 |
| a ₁₂ | 3 | 3 | 6 | 3 | 2 | 2 | 4 | -4 | -4 | -4 |
| a ₂₁ | -4 | -4 | -4 | -4 | -4 | 6 | 6 | 6 | 2 | 2 |
| a ₂₂ | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| b ₁ | 12 | 12 | 12 | 12 | 10 | 5 | 3 | 4 | 4 | 6 |
| b ₂ | 6 | 6 | 10 | 6 | 5 | 25 | 20 | 20 | 20 | 16 |

Таблица 2.4 – Варианты 11-20 к заданию 2.2

| | | | | | | | | | | |
|-----------------|----|----|----|----|----|----|----|----|----|----|
| вариант | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| c ₁ | 2 | 3 | 1 | 1 | 2 | 3 | 3 | 2 | 4 | 5 |
| c ₂ | 3 | 2 | 1 | 1 | 4 | 2 | 2 | 3 | 5 | 4 |
| a ₁₁ | 2 | 5 | 2 | 2 | 5 | -5 | -5 | 6 | 6 | 6 |
| a ₁₂ | 3 | 3 | 6 | 3 | 2 | 2 | 4 | -4 | -4 | -4 |
| a ₂₁ | -4 | -4 | -4 | -4 | -4 | 6 | 6 | 6 | 2 | 2 |
| a ₂₂ | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| b ₁ | 12 | 12 | 12 | 12 | 10 | 5 | 3 | 4 | 4 | 6 |
| b ₂ | 6 | 6 | 10 | 6 | 5 | 25 | 20 | 20 | 20 | 16 |

Решить задачу с использованием пакета ППП MathCad/ MS Excel, сравнить полученные решения.

Задание 2.3 Методом множителей Лагранжа решить задачу. Пусть у потребителя имеются финансовые средства в объеме S условных единиц, которые он готов потратить на приобретение двух видов продуктов. Известно, что цена единицы продукции первого вида – P_1 , цена единицы продукции второго вида – P_2 . Найти, какое количество продукции каждого вида будет приобретать потребитель располагая средствами в размере S , чтобы максимизировать свою полезность $U(x_1, x_2)$, где x_1 – количество продукта первого вида, которое готов приобрести потребитель; x_2 – количество продукта второго вида, которое готов приобрести потребитель.

Варианты заданий приведены в таблице 2.5.

Таблица 2.5 – Варианты к заданию 2.3

| Номер варианта | $U(x_1, x_2)$ | P_1 | P_2 | S |
|----------------|-----------------------|-------|-------|-----|
| 1 | $x_1^{1/3} x_2^{2/3}$ | 10 | 20 | 450 |
| 2 | $x_1^{1/3} x_2^{2/3}$ | 15 | 17 | 840 |
| 3 | $x_1^{1/3} x_2^{2/3}$ | 43 | 17 | 670 |
| 4 | $x_1^{1/3} x_2^{2/3}$ | 34 | 12 | 560 |
| 5 | $x_1^{2/3} x_2^{1/3}$ | 23 | 43 | 800 |
| 6 | $x_1^{2/3} x_2^{1/3}$ | 32 | 45 | 820 |
| 7 | $x_1^{2/3} x_2^{1/3}$ | 12 | 42 | 830 |
| 8 | $x_1^{1/5} x_2^{4/5}$ | 45 | 21 | 900 |
| 9 | $x_1^{2/5} x_2^{3/5}$ | 67 | 22 | 920 |

Продолжение таблицы 2.5

| Номер варианта | $U(x_1, x_2)$ | P1 | P2 | S |
|----------------|-----------------------|----|----|-----|
| 10 | $x_1^{3/5} x_2^{2/5}$ | 45 | 23 | 930 |
| 11 | $x_1^{4/5} x_2^{1/5}$ | 23 | 29 | 940 |
| 12 | $x_1^{1/7} x_2^{6/7}$ | 35 | 20 | 930 |
| 13 | $x_1^{2/7} x_2^{5/7}$ | 36 | 30 | 900 |
| 14 | $x_1^{3/7} x_2^{4/7}$ | 15 | 29 | 850 |
| 15 | $x_1^{4/7} x_2^{3/7}$ | 17 | 32 | 820 |
| 16 | $x_1^{5/7} x_2^{2/7}$ | 18 | 33 | 810 |
| 17 | $x_1^{6/7} x_2^{1/7}$ | 21 | 34 | 800 |
| 18 | $x_1^{1/5} x_2^{4/5}$ | 22 | 35 | 760 |
| 19 | $x_1^{2/5} x_2^{3/5}$ | 25 | 47 | 750 |
| 20 | $x_1^{3/5} x_2^{2/5}$ | 26 | 46 | 740 |
| 21 | $x_1^{4/5} x_2^{1/5}$ | 27 | 45 | 730 |
| 22 | $x_1^{1/7} x_2^{6/7}$ | 28 | 44 | 710 |
| 23 | $x_1^{2/7} x_2^{5/7}$ | 29 | 43 | 700 |
| 24 | $x_1^{3/7} x_2^{4/7}$ | 30 | 43 | 690 |
| 25 | $x_1^{4/7} x_2^{3/7}$ | 31 | 41 | 680 |
| 26 | $x_1^{5/7} x_2^{2/7}$ | 33 | 40 | 670 |
| 27 | $x_1^{6/7} x_2^{1/7}$ | 36 | 20 | 660 |
| 28 | $x_1^{1/7} x_2^{6/7}$ | 37 | 19 | 650 |
| 29 | $x_1^{2/7} x_2^{5/7}$ | 39 | 18 | 640 |
| 30 | $x_1^{3/7} x_2^{4/7}$ | 40 | 17 | 630 |

Решить задачу с использованием ППП MathCad или MS Excel, сравнить полученные решения.

Задание 2.4 Методом множителей Лагранжа решить задачу 2.1. Решить задачу с использованием ППП MathCad или MS Excel, сравнить полученные решения.

Задание 2.5 Методом множителей Лагранжа решить задачу 2.2. Решить задачу с использованием ППП MathCad или MS Excel, сравнить полученные решения.

2.2.11 Задания для самостоятельной проработки и подготовке к контрольным работам

1. Геометрическим способом найти максимум функции $F(x)$ при соответствующих ограничениях:

$$\begin{aligned}F &= 4 \cdot x_1 + 3 \cdot x_2, \\x_1^2 - 2 \cdot x_1 + x_2^2 - 2 \cdot x_2 - 34 &\leq 0, \\x_1 &\geq 0, \\x_2 &\geq 0.\end{aligned}$$

2. Геометрическим способом найти максимум функции при соответствующих ограничениях:

$$\begin{aligned}F &= 4 \cdot x_1 + 3 \cdot x_2, \\(x_1 - 2)^2 + (x_2 + 2)^2 - 34 &\leq 0, \\x_1 &\geq 0, \\x_2 &\geq 0.\end{aligned}$$

3. Определить оптимальное решение задачи нелинейного программирования методом множителей Лагранжа:

$$\begin{aligned}F(\bar{x}) &= 2 \cdot x_1 + x_2 \rightarrow \min \\4 \cdot x_1^{1/5} \cdot x_2^{2/3} &= 3200 \\x_1 &\geq 0, \quad x_2 \geq 0\end{aligned}$$

4. Предпочтения потребителя описываются функцией полезности $U(x_1, x_2) = (x_1 - 6)^{1/4} (x_2 - 10)^{3/4}$, доход равен 500, а цены товаров $p_1 = 10$, $p_2 = 50$. Полагая, что поведение потребителя рационально требуется определить

оптимальный набор товаров, который выберет потребитель при фиксированном доходе и заданном векторе цен, а также достигнутый уровень полезности.

5. Определить оптимальное решение задачи нелинейного программирования методом Лагранжа:

$$\begin{aligned} F &= 4 \cdot x_1 + 3 \cdot x_2 \rightarrow \max, \\ x_1^2 - 2 \cdot x_1 + x_2^2 - 2 \cdot x_2 - 34 &\leq 0, \\ x_1 &\geq 0, \\ x_2 &\geq 0. \end{aligned}$$

6. Определить оптимальное решение задачи нелинейного программирования методом Лагранжа:

$$\begin{aligned} F &= 4 \cdot x_1 + 3 \cdot x_2 \rightarrow \min, \\ (x_1 - 2)^2 + (x_2 + 2)^2 - 36 &\leq 0. \end{aligned}$$

7. Определить оптимальное решение задачи нелинейного программирования методом Лагранжа:

$$\begin{aligned} F &= 4 \cdot x_1 + 6 \cdot x_2 \rightarrow \max, \\ (x_1 - 2)^2 + (x_2 + 2)^2 - 49 &\leq 0. \end{aligned}$$

2.3 Численные методы решения задач нелинейного программирования

Материал раздела выполнен в соответствии с источниками: 3, 5, 6, 7, 28, 37.

2.3.1 Метод проекции градиента

Рассмотрим задачу нелинейного программирования:

$$f(x) \rightarrow \min, \quad (2.46)$$

$$x \in X \subset \mathbb{R}^n. \quad (2.47)$$

Предположим, что X – замкнутое и выпуклое множество, а функция $f(x)$ дифференцируема на множестве X .

Проекцией точки a на множество X называют ближайшую к x точку $y: y \in X$, $y = \Pi_x(a)$, причем среди всех точек множества X :

$$\Pi_x(a) = \arg \min_{x \in X} \varphi(x), \quad \varphi(x) = \|x - a\|^2. \quad (2.48)$$

Таким образом, $\Pi_x(a)$ является решением задачи проектирования (2.48).

Свойства операции проектирования:

1. Если $a \in X$, то $\Pi_x(a) = a$.

2. Если X – замкнутое множество, то любая точка a имеет проекцию y на множество X : $y = \Pi_x(a)$, если кроме того X – выпуклое множество, то проекция точки a на множество X единственная.

3. Точка $\Pi_x(a)$ является проекцией точки a на множество X тогда и только тогда, когда $\langle z - \Pi_x(a), a - \Pi_x(a) \rangle \leq 0 \quad \forall z \in X$, т.е. угол между векторами $\overline{(z - \Pi_x(a))}$ и $\overline{(a - \Pi_x(a))}$ не является острым.

4. Если X – выпуклое замкнутое множество и $a_1, a_2 \in \mathbb{R}^n$, то справедливо неравенство:

$$\|\Pi_x(a_1) - \Pi_x(a_2)\| \leq \|a_1 - a_2\|,$$

т.е. длина проекции отрезка на выпуклое множество не превосходит длины самого отрезка (обладает свойством нерастяжения расстояния).

Доказательства этих свойств приведены в [21].

ЛЕММА 2.1

Пусть множество X выпукло и замкнуто, функция $f(x)=f(x_1, \dots, x_n)$ выпукла на множестве X и дифференцируема в точке $x^* \in X$, тогда x^* является решением задачи (2.46)-(2.47) тогда и только тогда когда:

$$x^* = \Pi_X(x^* - \alpha \operatorname{grad} f(x^*)) \quad \forall \alpha > 0.$$

Пусть X – замкнутое выпуклое множество подпространства R^n : $X \subset R^n$.

Рассмотрим идеологию метода. В методе проекции градиента в качестве очередного приближения к решению задачи (2.46)-(2.47) выбирается проекция той точки, которая получается по обычному градиентному методу.

Таким образом, на первом этапе находим по обычному градиентному методу точку $\tilde{x}^k = x^k - \alpha_k \operatorname{grad} f(x^k)$ очередного приближения к точке минимума функции $f(x)$ без учета ограничений, затем решается задача проектирования полученной точки \tilde{x}^k на множество X :

$$x^{(k+1)} = \Pi_X(x^{(k)} - \alpha_k \operatorname{grad} f(x^{(k)})), \quad (2.49)$$

где α_k – шаг, значение определяется в соответствии с выбранным градиентным методом оптимизации многомерной функции.

В качестве критерия останова можно использовать:

1) критерии, основанные на понятии оценки абсолютной погрешности

$$\|x^{k+1} - x^k\| \leq \varepsilon_1,$$

$$\begin{aligned} |f(x^{k+1}) - f(x^k)| &\leq \varepsilon_2, \\ \|\text{grad} f(x^{k+1})\| &\leq \varepsilon_3; \end{aligned} \quad (2.50)$$

2) критерии, основанные на понятии оценки относительной погрешности:

$$\begin{aligned} \|x^{k+1} - x^k\| &\leq \delta_1 (1 + \|x^{k+1}\|), \\ |f(x^{k+1}) - f(x^k)| &\leq \delta_2 (1 + |f(x^{k+1})|), \\ \|\text{grad} f(x^{k+1})\| &\leq \delta_3 (1 + \|\text{grad} f(x^{k+1})\|). \end{aligned} \quad (2.51)$$

Можно также задать максимальное число итераций. Отметим, что выполнение указанных критериев не гарантирует достижения необходимой точности решения задачи, поскольку они могут выполняться и вдали от искомой точки минимума [5].

Приведем теорему о сходимости метода проекции градиента в случае априорного выбора α_k по методу расходящегося ряда.

ТЕОРЕМА 2.10 Теорема о сходимости метода проекции градиента, если α_k выбирается по методу расходящегося ряда.

Пусть X выпукло и замкнуто, $f(x)$ сильно выпукла с константой M , дифференцируема на множестве X , а градиент $\text{grad} f(x)$ удовлетворяет условию Липшица с некоторой константой θ : $\|\text{grad} f(x) - \text{grad} f(x')\| \leq \theta \|x - x'\|, \forall x, x' \in X$. Тогда последовательность $\{x^k\}$, генерируемая по методу (2.49), где $x^0 \in X$ сходится к решению x^* задачи (2.47)-(2.48) со скоростью геометрической прогрессии.

ЗАМЕЧАНИЕ. В этом методе на каждой итерации требуется производить операцию проектирования на множество X , т.е. необходимо решение задачи минимизации функции $\varphi(x)$ при выбранном значении a , что трудоемко, если множество x задано сложно. В таких случаях прибегают к модификации приведенного метода, в которой проектирование на множество X заменяют проектированием на полиэдры, аппроксимирующий X в окрестности очередной точки [5].

Если же X достаточно простое множество (квадрат, куб), то проекцию можно найти в явном виде. Рассмотрим наиболее часто встречающиеся множества.

1. X – шар, радиусом R , x^c – центр:

$$X = \{x : \|x - x^c\| \leq R\}, \quad \Pi_x(a) = x^c - \frac{a - x^c}{\|a - x^c\|} R. \quad (2.52)$$

2. X – координатный параллелепипед:

$$X = \{x : b_j \leq x_j \leq c_j\}, \quad \Pi_x(a) = \begin{cases} b_j, & a_j < b_j \\ a_j, & b_j \leq a_j \leq c_j \\ c_j, & a_j > c_j \end{cases}, \quad j = \overline{1, n}; \quad (2.53)$$

3. X – полиэдр:

$$X = \{x : Ax \leq b\}, \quad \Pi_x(a) = a - A^T(A^T A)^{-1}(Aa - b); \quad (2.54)$$

4. X – гиперплоскость:

$$X = \{x : \langle p, x \rangle = \beta\}, \quad \Pi_x(a) = a + (\beta - \langle p, a \rangle) \cdot \frac{p}{\|p\|^2}. \quad (2.55)$$

Алгоритм метода проекции градиента:

Шаг 1. Задаем критерий останова (или любое их сочетание), точность вычислений ε , начальную точку $x^0 \in X$.

Шаг 2. Полагаем $k = 0$.

Шаг 3. Методом градиентного спуска находим $\tilde{x}^{k+1} = x^k - \alpha_k \text{grad}f(x^k)$.

Выбор шага α_k производится любым методом.

Шаг 4. Находим проекцию $x^{k+1} = \Pi_X(\tilde{x}^{k+1})$ на множество X по одной из формул (2.52)-(2.55).

Шаг 5. Проверяем достижение заданной степени точности. Если точность достигнута, то шаг 6. Иначе полагаем $x^{k+1} = x^k$, $k = k + 1$ и переходим к шагу 3.

Шаг 6. Принимаем $x^* := x^{k+1}$, $f(x^*) = f(x^{k+1})$.

Конец алгоритма.

2.3.2 Метод условного градиента

Пусть поставлена задача вида (2.46)-(2.47), где $f(x)$ дифференцируема на X , X – компакт.

Рассмотрим метод вида:

$$x^{k+1} = x^k + \alpha_k h^k, \quad \alpha_k > 0, \quad (2.56)$$

где α_k – шаг, выбирается из условия: $f(x^{k+1}) < f(x^k) : x^{k+1} \in X$;

h^k задает направление убывания.

Для нахождения направления спуска будем использовать минимизацию линейной аппроксимации функции $f(x)$ в точке x^{k-1} :

$$f_k(x) = f(x^{k-1}) + \langle \text{grad} f(x^{k-1}), x - x^{k-1} \rangle. \quad (2.57)$$

Так как $f(x^{k-1}) = \text{const}$, то задачу (2.57) можно записать в виде:

$$\langle \text{grad} f(x^{k-1}), x - x^{k-1} \rangle \rightarrow \min_{x \in X}. \quad (2.58)$$

Пусть $\tilde{x}^k = \arg \min_{x \in X} \langle \text{grad}f(x^{k-1}), x - x^{k-1} \rangle$. Обозначим

$\eta_k = \langle \text{grad}f(x^{k-1}), \tilde{x}^k - x^{k-1} \rangle$ – значение функции (2.58) в точке \tilde{x}^k .

Т.к. X – компакт, а $f(x)$ дифференцируема на множестве X , то такая точка \tilde{x}^k всегда существует. Учитывая, что $\tilde{x}^k \in X$, имеем $\eta_k \leq \langle \text{grad}f(\tilde{x}^k), x - \tilde{x}^k \rangle = 0$. Поэтому возможны только два случая: $\eta_k = 0$ или $\eta_k < 0$.

1. Пусть $\eta_k = 0$,

Т.к. $\tilde{x}^k \in X \Rightarrow \eta_k \leq \langle \text{grad}f(\tilde{x}^k), x - \tilde{x}^k \rangle, \forall x \in X$,

$0 = \eta_k \leq \langle \text{grad}f(\tilde{x}^k), x - \tilde{x}^k \rangle \quad \forall x \in X$.

Т.е. в этом случае точка \tilde{x}^k удовлетворяет необходимому условию оптимальности и является стационарной точкой задачи (2.46)-(2.47). В этом случае работа алгоритма заканчивается, если $f(x)$ выпуклая функция, то \tilde{x}^k – решение задачи (2.46)-(2.47), в других случаях точку \tilde{x}^k необходимо исследовать на оптимальность вне рамок приведенного метода.

2. Если $\eta_k \neq 0$ ($\eta_k < 0$), тогда:

$$\langle \text{grad}f(x^{k-1}), \tilde{x}^k - x^{k-1} \rangle < 0,$$

$$x^k = x^{k-1} + \alpha_k (\tilde{x}^k - x^{k-1}), \quad \alpha_k \in [0,1], \quad x^k \in X.$$

Тогда полагаем:

$$h^k = \tilde{x}^k - x^{k-1}.$$

Вектор h^k принято называть условным антиградиентом функции f в точке \tilde{x}^k . Заметим, что h^k является направлением спуска функции $f(x)$, поскольку $\langle \text{grad}f(\tilde{x}^k), h^k \rangle = \eta_k < 0$. Кроме того, используя выпуклость множества X , для

любого $\alpha \in (0,1]$ имеем $x^{k-1} + \alpha \cdot h^k = \alpha \cdot \tilde{x}^k + (1-\alpha) \cdot x^{k-1} \in X$, то есть h^k определяет допустимое направление спуска.

Таким образом, вспомогательное приближение \tilde{x}^k на каждой k -ой итерации задает условный антиградиент $h^k = \tilde{x}^k - x^{k-1}$ (условный антиградиент), определяющий на этой итерации допустимое направление в точке $x^{k-1} \in X$. В общем случае он не является направлением антиградиента целевой функции. Новая точка x^k определяется следующим образом:

$$x^k = x^{k-1} + \alpha_k h^k, \quad (2.59)$$

$$h^k = \tilde{x}^k - x^{k-1}. \quad (2.60)$$

Шаг $\alpha_k > 0$ можно выбрать из условного минимума функции $\varphi_k(\alpha) = f(x^{k-1} + \alpha h^k)$ по аргументу α в полуинтервале $(0,1]$ или другим методом исходя из условия релаксации $f(x^k) \leq f(x^{k-1})$. Обоснование выбора шага α заключено в следующей лемме.

ЛЕММА 2.2

Пусть X – компакт, $f(x)$ дифференцируема на множестве X , $\text{grad } f(x)$ удовлетворяет условию Липшица с константой равной θ , тогда справедливо неравенство:

$$0 < \alpha < \frac{-(1-\varepsilon)f(\tilde{x}^k)}{\theta \|h^k\|^2}, \quad (2.61)$$

где $f(\tilde{x}^k) < 0$, $\forall \varepsilon \in (0,1)$.

Опишем свойства предельных точек последовательности $\{x^k\}$, генерируемой по правилу (2.59), где x^0 – произвольная точка из X , h^k определяется по формуле (2.60), а α_k удовлетворяет описанным выше условиям. При этом считаем, что

процесс генерирования продолжается бесконечно, т.е. $\eta_k < 0$ при всех $k = 0, 1, \dots$

ТЕОРЕМА 2.11 Теорема о сходимости метода условного градиента.

Пусть X – выпуклый компакт, тогда точка x_k^* , предельная точка последовательности $\{x_k\}: x_k^* = \lim_{k \rightarrow \infty} x_k$, порожденной методом условного градиента, является стационарной. Если $f(x)$ выпуклая функция, то x^* – решение задачи (2.46)-(2.47) и $\lim_{k \rightarrow \infty} f(x_k) = f(x^*)$.

Недостатки метода:

1) низкая скорость сходимости;

2) метод оправдан в тех случаях, когда решение задачи минимизации (2.58) удается представить в явном виде или достаточно просто находится. Ниже приведены примеры таких множеств и соответствующие решения задачи (2.58).

1. Если X – шар, радиусом R и центром в точке $x^c : x = \{x \in \mathbb{R}^n : \|x - x^c\| \leq R\}$,

тогда:

$$\tilde{x}^{k+1} = x^c - \frac{\text{grad}f(x^k)}{\|\text{grad}f(x^k)\|} R; \quad (2.62)$$

1. Если X – координатный параллелепипед вида $X = \{x \in \mathbb{R}^n : x_j \in [a_j, b_j], i = 1, n\}$, тогда

$$\tilde{x}_j^{k+1} = \begin{cases} a_j, & \text{при } \frac{\partial f(x^k)}{\partial x_j} > 0, \\ b_j, & \text{при } \frac{\partial f(x^k)}{\partial x_j} < 0, \\ \forall z \in [a_j, b_j], & \text{при } \frac{\partial f(x^k)}{\partial x_j} = 0, \end{cases} \quad j = \overline{1, n}; \quad (2.63)$$

3. Если X – многогранное множество $X = \{x \in \mathbb{R}^n : \langle a_i, x \rangle \leq b_i, i = 1, \dots, m\}$, или

$X = \{x \in \mathbb{R}^n : \langle a_i, x \rangle \leq b_i, i = 1, \dots, m, Bx = d\}$, где B – матрица размерности $l \times n$, а $d \in \mathbb{R}^l$, то задача (2.58) сводится к задаче линейного программирования, которая может быть решена симплекс-методом.

Алгоритм метода условного градиента:

Шаг 1. Задаем точность вычислений ε , начальную точку $x^0 \in X$;

Шаг 2. Полагаем $k = 1$;

Шаг 3. Решаем задачу $\tilde{x}^k = \arg \min_{x \in X} (\text{grad}f(x^{k-1}), x - x^{k-1})$ (например,

используя формулы (2.62), (2.63)). Вычисляем $\eta_k = (\text{grad}f(x^{k-1}), \tilde{x}^k - x^{k-1})$;

Шаг 4. Проверяем условие окончания итераций. Если $\eta_k < \varepsilon$, то переходим к шагу 7. Иначе получаем направление $h^k = \tilde{x}^k - x^{k-1}$;

Шаг 5. Осуществляем спуск в этом направлении: $x^k = x^{k-1} + \alpha_k h^k$. Выбор шага α_k производится любым методом;

Шаг 6. Полагаем $x^{k+1} = x^k$, $k = k + 1$ и переходим к шагу 3;

Шаг 7. Принимаем $x^* = \tilde{x}^{k+1}$, $f(x^*) = f(\tilde{x}^{k+1})$.

Конец алгоритма.

2.3.3 Методы штрафных функций, общая идеология

Пусть поставлена задача нелинейной условной оптимизации вида:

$$f(x) = f(x_1, \dots, x_n) \rightarrow \min, \quad (2.64)$$

$$X = \begin{cases} h_i(x) = 0, i = 1, \dots, l, \\ g_i(x) \leq 0, i = 1, \dots, m. \end{cases} \quad (2.65)$$

$$(2.66)$$

И пусть $f(x)$ непрерывна и дифференцируема на X . Множество допустимых

решений X задано дважды непрерывно дифференцируемыми функциями g_i и h_i .

Метод основывается на введении функции штрафа, зависящей от штрафного параметра и обладающей свойствами:

- 1) на большей части допустимого множества эти функции близки к 0;
- 2) каждая из них достаточно быстро возрастает при приближении к границе допустимого множества изнутри (метод внутренних штрафных функций) или при выходе за его пределы (метод внешних штрафных функций);
- 3) степень близости штрафа к 0 и скорость возрастания зависит от значения штрафного параметра.

Идея метода: функция штрафа добавляется к целевой функции, после чего решается параметрическое семейство получившихся задач безусловной оптимизации, при различных значениях параметров. Общий вид вспомогательной функции:

$$F(x) = f(x) + R(x) \rightarrow \min, \quad (2.67)$$

где $R(x)$ – штрафная функция, такая что:

$$\begin{aligned} R(x) &= 0 \quad \text{при } x \in X, \\ R(x) &= \infty \quad \text{при } x \notin X. \end{aligned}$$

При такой постановке точка минимума функций $f(x)$ в задаче (2.64)-(2.66) и функции $F(x)$ в задаче (2.67) совпадают, но функция штрафа имеет точки разрыва 2-го рода на границе допустимого множества X . Это не позволяет применять известные методы безусловной минимизации к задаче (2.67). Но в случае, когда множество X задано ограничениями типа равенств и неравенств, используя фигурирующие в ней функции можно построить такие непрерывные функции штрафа $R(x, r^k)$, где r^k – параметр штрафа на k -ой итерации, что в каждой фиксированной точке $x \in R^n$:

$$\lim_{k \rightarrow \infty} R(x, r^k) = R(x),$$

тогда задача (2.67) примет вид:

$$\min_{x \in \mathbb{R}^n} \left(f(x) + \lim_{k \rightarrow \infty} R(x, r^k) \right) \sim \lim_{k \rightarrow \infty} \left(\min_{x \in \mathbb{R}^n} \left(f(x) + R(x, r^k) \right) \right).$$

Таким образом, получим задачу:

$$F(x, r^k) = f(x) + R(x, r^k) \rightarrow \min, \quad x \in \mathbb{R}^n, k \in \mathbb{N}. \quad (2.68)$$

Т.е. исходная задача нелинейного программирования (2.64)-(2.66) переходит в последовательность задач безусловной минимизации функций $F(x, r^k)$ и в качестве решения задачи (2.64)-(2.66) x^* будем рассматривать предел (если он существует) последовательности $\{x_*^k\}$ решения задач (2.68).

2.3.4 Метод внешних штрафных функций

Пусть имеется задача вида:

$$\begin{aligned} f(x) &\rightarrow \min, \\ h(x) &= 0. \end{aligned}$$

Рассмотрим для этого случая штрафную функцию вида:

$$F(x, r^k) = f(x) + r^k h^2(x), \quad x \in \mathbb{R}^n,$$

где r^k – некоторое большое число.

Ясно, что на оптимальном решении x^* значение $h^2(x^*)$ должно быть близкими к 0, т.к. в противном случае всегда можно сдвинуться в другую точку, в которой приращение $f(x)$ окажется при достаточно большом r^k меньше, чем $r^k \cdot h^2$.

Рассмотрим задачу:

$$\begin{aligned} f(x) &\rightarrow \min, \\ g(x) &\leq 0. \end{aligned}$$

Очевидно, что форма $f(x) + r^k g^2(x)$ не целесообразна, т.к. при $g(x) \neq 0$ штраф будет взиматься независимо от знака $g(x)$, поэтому приемлемой будет задача:

$$f(x) + r^k (\max\{0, g(x)\})^2,$$

В этом случае, если $g(x) \leq 0$, то $\max\{0, g(x)\} = 0$ и штраф не берется. Если $g(x) > 0$, то выражение в скобках больше 0, т.е. штраф взимается.

Обычно подходящая штрафная функция должна определять положительный штраф в недопустимых точках и не штрафовать допустимые.

Если ограничения имеют вид:

$$\begin{aligned} h_i(x) &= 0, \quad i = 1, \dots, l, \\ g_j(x) &\leq 0, \quad j = 1, \dots, m, \end{aligned}$$

то целесообразна штрафная функция вида:

$$R(x, r^k) = \frac{r^k}{2} \left\{ \sum_{i=1}^m \psi(g_i(x)) + \sum_{j=1}^l \varphi(h_j(x)) \right\},$$

при этом функции ψ и φ удовлетворяют условиям:

$$\psi(y) = \begin{cases} 0, & y \leq 0 \\ > 0, & y > 0 \end{cases}, \quad \varphi(y) = \begin{cases} 0, & y = 0 \\ > 0, & y \neq 0 \end{cases}.$$

Типичными являются функции вида:

$$\psi(y) = [\max\{0, y\}]^p,$$

$$\varphi(y) = |y|^p,$$

где p – целое положительное число.

Как правило, для ограничений типа равенств используют квадратичный штраф, а для ограничений типа неравенств – квадрат срезки:

$$R(x, r^k) = \frac{r^k}{2} \left\{ \sum_{j=1}^l [h_j(x)]^2 + \sum_{i=1}^m [g_i^+(x)]^2 \right\}, \quad (2.69)$$

где $g_i^+(x)$ срезка, функция вида:

$$g_i^+(x) = \max\{0, g_i(x)\} = \begin{cases} g_i(x), & \text{при } g_i(x) > 0 \\ 0, & \text{при } g_i(x) \leq 0 \end{cases}.$$

Начальная точка поиска может задаваться как вне множества допустимых решений X , так и внутри него. На каждой k -ой итерации ищется точка $x^*(r^k)$ минимума вспомогательной функции $F(x, r^k)$ при заданном параметре r^k с помощью одного из методов безусловной минимизации. Полученная точка $x^*(r^k)$

используется в качестве начальной на следующей итерации, выполняемой при возрастающем значении параметра штрафа.

Приведем теорему о сходимости метода.

ТЕОРЕМА 2.12

Пусть x^* – локально-единственное решение задачи поиска условного минимума, а функции $f(x)$, $g_i(x)$, $h_j(x)$ $i=1,..m$, $j=\overline{1,1}$ непрерывны дифференцируемы в окрестности x^* , тогда при достаточно больших r^k найдется точка x^* зависящая от r^k (будем обозначать $x^*(r^k)$) локального минимума функций $F(x, r^k) = f(x) + R(x, r^k)$ в окрестности x^* и $x^*(r^k) \rightarrow x^*$.

Недостаток: При неограниченном возрастании r^k последовательность точек $x^*(r^k)$ стремится к точке условного минимума x^* , но поскольку процедура расчетов завершается при некотором конечном значении параметра штрафа, то приближенное решение, как правило, не принадлежит области допустимых решений, то есть ограничения задачи не выполняются.

Алгоритм метода внешних штрафных функций:

Шаг 1. Задаем начальную точку x^0 , параметр r^0 , константу для увеличения штрафа $C > 1$ и ε – точность $\varepsilon > 0$, $k=0$, $C \in [2, 10]$;

Шаг 2. Строим функцию:

$$F(x, r^k) = f(x) + \frac{r^k}{2} \left\{ \sum_{j=1}^m (g_j^+(x))^2 + \sum_{i=1}^1 [h_i(x)]^2 \right\};$$

Шаг 3. Находим точку $x^*(r^k)$ безусловного минимума функции $F(x, r^k)$ по x с помощью любого метода (нулевого, первого или второго порядка): $x^*(r^k) = \operatorname{argmin}_{x \in R^n} F(x, r^k)$. В качестве начального приближения берем x^k ;

Шаг 4. Вычисляем значение штрафной функции в этой точке $R(x^*(r^k), r^k)$;

Шаг 5. Проверяем условие окончания итераций:

если $R(x^*(r^k), r^k) \leq \varepsilon$ тогда перейти к ш.6;

если $R(x^*(r^k), r^k) > \varepsilon$, тогда $r^{k+1} = Cr^k, x^{k+1} = x^*(r^k)$, $k = k + 1$ перейти к шагу 2;

Шаг 6. Принимаем $x^* = x^*(r^k)$, $f(x^*) = f(x^*(r^k))$.

Конец алгоритма.

2.3.5 Метод внутренних (барьерных) штрафных функций

Пусть имеется задача:

$$f(x) \rightarrow \min, \quad (2.70)$$

$$x \in X = \{g_i(x) \leq 0, i = 1, \dots, m\}, \quad (2.71)$$

и пусть допустимое множество (2.71) содержит внутренние точки, т.е. точки, в которых все ограничения выполнены как строгие неравенства.

Функцию $R(x)$, определенную во всех внутренних точках допустимого множества X , называют функцией внутреннего штрафа, если:

$$R(x) = \begin{cases} > 0, & x \in X \setminus \partial X, \\ +\infty, & x \notin X \setminus \partial X. \end{cases}$$

Ее роль состоит в создании «барьера» с крутыми склонами вдоль границы допустимого множества X .

Идея метода сохраняется, но в качестве штрафных функций используются:

1) обратная штрафная функция:

$$R(x, r^k) = -r^k \sum_{j=1}^m \frac{1}{g_j(x)}; \quad (2.72)$$

2) логарифмическая штрафная функция:

$$R(x, r^k) = -r^k \sum_{j=1}^m \ln[-g_j(x)]. \quad (2.73)$$

Обе штрафные функции определены и непрерывны внутри множества X и стремятся к бесконечности при приближении к границам множества изнутри.

При $r^k > 0$ обратная штрафная функция положительна, логарифмическая штрафная функция положительна при $g(x) \in (-1, 0)$ и отрицательна при $g(x) < -1$, т.е. внутренним точкам отдается предпочтение перед граничными. Начальная точка задается только внутри множества X . Барьерные (внутренние) штрафные функции как бы препятствуют выходу из множества X , а если решение задачи лежит на границе, то процедура метода приводит к движению изнутри области к границам.

УТВЕРЖДЕНИЕ 2.5

Пусть функции $f(x)$ и $g_i(x)$, $i=1, \dots, m$ выпуклы и конечны. Множество X решений задачи поиска условного минимума непустое и ограничено. И существует точка $x^0 \in X$ такая что $g_i(x^0) < 0$, $i = \overline{1, m}$, т.е. x^0 является внутренней точкой. Тогда в методе барьерных функций $x_k^* = \arg \min_{x \in R^n} F(x, r^k)$ образуют непустое множество; функции $F(x, r^k)$ выпуклы, последовательность $\{x(r^k)\}_{k=0}^{\infty}$ порождаемая алгоритмом, ограничена и все ее предельные точки принадлежат X , причем $f(x^*(r^k)) \geq f(x^*)$, $x^* \in X$.

Алгоритм метода внутренних штрафных функций:

Шаг 1. Задаем начальную точку x^0 , лежащую внутри области X , r^0 , $C > 1$ для уменьшения параметра штрафа, точность вычислений $\varepsilon > 0$, $k=0$;

Шаг 2. Составим функцию штрафа:

$$F(x, r^k) = f(x) + R(x, r^k),$$

где $R(x, r^k)$ определены по формулам (2.72) или (2.73);

Шаг 3. Находим точку $x^*(r^k)$ безусловного минимума функции $F(x, r^k)$ по x с помощью любого метода (нулевого, первого или второго порядка):
 $x^*(r^k) = \arg \min_{x \in R^n} F(x, r^k)$. В качестве начальной точки взять x^k ;

Шаг 4. Вычисляем значение штрафной функции в этой точке $R(x^*(r^k), x^k)$;

Шаг 5. Проверка условия окончания итераций:

если $R(x^*(r^k), r^k) \leq \varepsilon$, тогда перейти к шагу 6;

если $R(x^*(r^k), r^k) > \varepsilon$, тогда $r^{k+1} = \frac{r^k}{C}$, $x^{k+1} = x^*(r^k)$, $k = k + 1$ и перейти к

шагу 2.

Шаг 6. Принимаем $x^* = x^*(r^k)$, $f(x^*) = f(x^*, r^k)$.

Конец алгоритма.

ЗАМЕЧАНИЕ 1. Т.к. шаг вблизи границы может привести в точку вне допустимой области, что в свою очередь может привести к ложному успеху, т.е. уменьшению вспомогательной функции в точке, где она теоретически не определена, в алгоритме необходимо предусмотреть явную проверку того, что точка не покинула допустимую область.

ЗАМЕЧАНИЕ 2. Эффективность алгоритма зависит от выбора начального значения r^0 параметра штрафа. Общих рекомендаций здесь не существует. Выбор подходящего значения r^0 связан со спецификой решаемой задачи и способом задания функции штрафа. Например, если штраф определяют по формуле (3), то в качестве начального значения:

$$r^0 = \frac{\langle \text{grad}f(x), \text{grad}Z(x) \rangle}{\|\text{grad}Z(x)\|^2},$$

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

ЗАМЕЧАНИЕ 3. Условие прекращения поиска точки $x^*(r^k)$ на шаге 3 зависит от использованного алгоритма безусловной минимизации. Часто, в случае применения методов 1 или 2-го порядка, в качестве такого условия выбирают:

$$\|\text{grad}F(x, r^k)\| \leq \varepsilon,$$

однако использование этого условия может приводить к трудностям вычислительного характера и не всегда практически возможно, т.к. при больших k значение $\text{grad}F(x, r^k)$ может быть большим, даже в малой окрестности x_*^k .

2.3.6 Комбинированный метод штрафных функций

Пусть задача имеет вид:

$$f(x) \rightarrow \min \quad (2.74)$$

$$g_i(x) \leq 0, \quad i = 1, \dots, m, \quad (2.75)$$

$$h_i(x) = 0, \quad i = 1, \dots, l. \quad (2.76)$$

Тогда для ограничений типа (2.76) можно использовать внешние штрафные функции, а для (2.75) – внутренние:

$$F(x, r^k) = f(x) + \frac{1}{r^k} \sum_{i=1}^l h_i^2(x) - r^k \sum_{i=1}^m \frac{1}{g_i(x)}. \quad (2.77)$$

В остальном идеология алгоритмов сохраняется.

2.3.7 Вопросы для самопроверки

1. Опишите численные методы нелинейной условной оптимизации: метод проекции градиента, сходимость, алгоритм.
2. Опишите численные методы нелинейной условной оптимизации: метод условного градиента, сходимость, алгоритм.
3. Охарактеризуйте общую идею метода штрафных функций.
4. Опишите метод внутренних (барьерных) штрафные функции, приведите алгоритм.
5. Опишите метод внешних штрафных функции, приведите алгоритм.

2.3.8 Задания для практических (лабораторных) работ

Рекомендации к выполнению и оформлению практических (лабораторных) работ по разделу приведены в источнике 17.

Задание 2.7 Разработать программное средство для решения задач условной оптимизации функций двух переменных методами проекции градиента. Варианты заданий приведены в таблице 2.7.

Таблица 2.7 – Индивидуальные задания для решения задач методом проекции градиента

| № | Функция | Вид допустимой области |
|---|---|-----------------------------|
| 1 | 2 | 3 |
| 1 | $2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_2 - 4x_1$ | координатный параллелепипед |
| 2 | $2x_1^2 - 4x_1 + x_2^2 - 2x_2 + x_3$ | шар радиуса R |
| 3 | $10x_1^2 - 4x_1x_2 + 7x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | гиперплоскость |
| 4 | $(x_1 - 5)^2 + (x_2 - 3)^2 + (x_3 - 7)^2$ | полиэдр |

Продолжение таблицы 2.7

| 1 | 2 | 3 |
|----|---|-----------------------------|
| 5 | $1/3(x_1 + 1)^3 + x_2 - 2x_1x_2$ | полиэдр |
| 6 | $(x_1 - 5)^2 + (x_2 - 3)^2$ | гиперплоскость |
| 7 | $(x_1 - 4)^2 + (x_2 + 1)^2 + x_1x_2$ | координатный параллелепипед |
| 8 | $1/3(x_2 + 2)^3 + x_1$ | шар радиуса R |
| 9 | $x_1x_2x_3$ | координатный параллелепипед |
| 10 | $2x_1^2 + 2x_1 + 4x_2 - 3x_3$ | шар радиуса R |
| 11 | $(x_1 - 15)^2 + (x_2 - 3)^2 + x_3$ | шар радиуса R |
| 12 | $\frac{5}{2x_1} + \frac{9}{x_2} + 2x_1 + x_2$ | координатный параллелепипед |
| 13 | $x_2^3 - 3x_1x_2$ | полиэдр |
| 14 | $12x_1^2 - 2x_1x_2 + 6x_2^2 - 4\sqrt{5}(3x_1 - x_3) - 10$ | гиперплоскость |
| 15 | $x_1^2 + x_2^2$ | координатный параллелепипед |
| 16 | $x_1x_2x_3 + 5x_2$ | шар радиуса R |
| 17 | $(x_1 - 4)^2 + (x_2 + 1)^2 + x_1x_2$ | шар радиуса R |
| 18 | $10x_1^2 - 4x_1^2x_2 + 7x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | полиэдр |
| 19 | $\frac{1}{3}(x_1 + 1)^3 + 7x_2$ | гиперплоскость |
| 20 | $\frac{1}{3}(x_1 + 1)^3 + x_2 - 2x_1x_2 + x_3^2$ | полиэдр |

Задание 2.8 Разработать программное средство для решения задач условной оптимизации функций двух переменных методами проекции градиента. Варианты заданий приведены в таблице 2.8.

Таблица 2.8 – Индивидуальные задания для решения задач методом условного градиента

| № | Функция | Вид допустимой области |
|---|---|---|
| 1 | 2 | 3 |
| 1 | $8x_1^2 - 4x_1 + x_2^2 - 12x_2 + 7$ | многогранное множество, координатный параллелепипед |
| 2 | $x_1x_2x_3$ | многогранное множество, шар радиуса R |
| 3 | $15x_1^2 - 5x_1x_2 + 2x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | многогранное множество, координатный параллелепипед |

Продолжение таблицы 2.8

| 1 | 2 | 3 |
|----|---|---|
| 4 | $2x_1^2 + 2x_1 + 4x_2 - 3x_3$ | многогранное множество, координатный параллелепипед |
| 5 | $\frac{4}{x_1} + \frac{9}{x_2} + x_1^2 + x_2^2$ | многогранное множество, координатный параллелепипед |
| 6 | $x_1x_2x_3 + 5x_1$ | многогранное множество, шар радиуса R |
| 7 | $x_2^3 - 3x_1x_2 + x_2^2$ | многогранное множество, шар радиуса R |
| 8 | $x_1 + 2x_2$ | многогранное множество, координатный параллелепипед |
| 9 | $x_1^2 + 3x_2^2$ | многогранное множество, шар радиуса R |
| 10 | $10x_1^2 - 4x_1x_2 + 7x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | многогранное множество, шар радиуса R |
| 11 | $\frac{1}{3}(x_1 + 1)^3 + x_2$ | многогранное множество, шар радиуса R |
| 12 | $-x_1^2 - 2x_2^2$ | многогранное множество, шар радиуса R |
| 13 | $x_2^3 + 6x_1x_2 + x_2^2 + x_3$ | многогранное множество, шар радиуса R |
| 14 | $x_2^3 - 3x_1x_2$ | многогранное множество, координатный параллелепипед |
| 15 | $12x_1^2 - 2x_1x_2 + 6x_2^2 - 4\sqrt{5}(3x_1 - x_3) - 10$ | многогранное множество, шар радиуса R |
| 16 | $x_1^2 + x_2^2$ | многогранное множество, координатный параллелепипед |
| 17 | $\frac{5}{2x_1} + \frac{9}{x_2} + 2x_1 + x_2$ | многогранное множество, координатный параллелепипед |
| 18 | $12x_1^2 - 3x_1x_2 + 6x_2^2 - 4\sqrt{5}(3x_1 - x_2) - 11$ | многогранное множество, координатный параллелепипед |
| 19 | $\frac{1}{3}(x_1 + 1)^3 + x_2 - 2x_1x_2$ | многогранное множество, координатный параллелепипед |
| 20 | $(x_1 - 5)^2 + (x_2 - 3)^2$ | многогранное множество, шар радиуса R |

Задание 2.9 Разработать программное средство для решения задач условной оптимизации функций двух переменных. Варианты заданий приведены в таблице 2.9.

Таблица 2.9 – Индивидуальные задания для решения задач методами внешних и внутренних штрафных функций

| № | Функция | Ограничения | Вид штрафных функций |
|----|---|---|----------------------|
| 1 | 2 | 3 | 4 |
| 1 | $10x_1^2 - 4x_1x_2 + 7x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | $(x_1 - 1)^2 + (x_2 - 3)^2 - 4 \leq 0$ | внутренние |
| 2 | $x_1^2 + x_2^2$ | $x_1 + x_2 - 2 = 0$ | внешние |
| 3 | $x_1^2 + x_2^2$ | $x_1 - 1 = 0, x_1 + x_2 - 2 \leq 0$ | внешние |
| 4 | $10x_1^2 - 4x_1x_2 + 7x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | $x_1 - 2 \leq 0, x_1 + x_2 - 1 \leq 0$ | внутренние |
| 5 | $15x_1^2 - 5x_1x_2 + 2x_2^2 - 4\sqrt{5}(5x_1 - x_2) - 16$ | $x_1^2 + (x_2 + 2)^2 - 4 \leq 0$ | внутренние |
| 6 | $(x_1 - 5)^2 + (x_2 - 3)^2$ | $x_1 + x_2 \leq 3, -x_1 + 2x_2 \leq 4$ | внутренние |
| 7 | $x_2^3 - 3x_1x_2$ | $4x_1 + 10x_2 \leq 40,$ $x_1 - 2x_2 = 5, x_1 \geq 0, x_2 \geq 0$ | внешние |
| 8 | $x_2^3 - 3x_1x_2 + x_2^2$ | $2x_1 + 5x_2 \leq 20,$ $2x_1 - 4x_2 = 10$ | внешние |
| 9 | $\frac{1}{3}(x_1 + 1)^3 + x_2$ | $x_1 - 1 \leq 0, x_2 \geq 0$ | внутренние |
| 10 | $8x_1^2 - 4x_1 + x_2^2 - 12x_2 + 7$ | $2x_1 + 3x_2 = -6$ | внешние |
| 11 | $\frac{4}{x_1} + \frac{9}{x_2} + x_1 + x_2$ | $x_1 + x_2 \leq 6, x_1 \geq 0, x_2 \geq 0$ | внутренние |
| 12 | $(x_1 + 4)^2 + (x_2 - 4)^2$ | $2x_1 - x_2 \leq 2, x_1 \geq 0, x_2 \geq 0$ | внешние |
| 13 | $x_1^2 + 2x_2^2$ | $-x_1 + x_2 \leq 0,$ $-x_1 - x_2 - 1 \leq 0$ | внутренние |
| 14 | $2x_1^2 + 2x_1 + 4x_2 - 3x_3$ | $8x_1 - 3x_2 + 3x_3 \leq 40,$ $-2x_1 + x_2 - x_3 = -3$ | внешние |
| 15 | $2x_1^2 + x_2^2$ | $-x_1 - x_2 + 2 \leq 0,$ $x_1 - 2x_2 + 1 \leq 0,$ $-2x_1 + x_2 \leq 0$ | внутренние |
| 16 | $\frac{5}{2x_1} + \frac{9}{x_2} + 2x_1 + x_2$ | $x_1 + x_2 \leq 12, x_1 \geq 0, x_2 \geq 0$ | внешние |
| 17 | $x_1^2 + x_2^2$ | $2x_1 + x_2 \geq 2, 2x_1 + x_2 \leq 8,$ $x_1 + x_2 \geq 6, x_1 \geq 0, x_2 \geq 0$ | внутренние |

Продолжение таблицы 2.9

| 1 | 2 | 3 | 4 |
|----|---|--|------------|
| 18 | $x_1^2 + x_2^2$ | $(x_1 - 1)^3 - x_2^2 \leq 0$ | внешние |
| 19 | $\frac{4}{x_1} + \frac{9}{x_2} + x_1^2 + x_2^2$ | $x_1 + x_2 \leq 6, x_1 \geq 0, x_2 \geq 0$ | внутренние |
| 20 | $x_1 + 2x_2$ | $8x_1 - 3x_2 + 3x_3 \leq 40,$ $8x_1 - 3x_2 + 3x_3 \leq 40,$ $x_1 \geq 0, x_2 \geq 0$ | внешние |

2.3.9 Задания для самостоятельной проработки и подготовке к контрольным работам

1. Выполнить 2 этапа поиска точки минимума по методу проекции градиента для задачи:

$$F = 4 \cdot x_1 + 3 \cdot x_2 \rightarrow \min,$$

$$(x_1 - 2)^2 + (x_2 + 2)^2 - 34 \leq 0.$$

2. Выполнить 2 этапа поиска точки минимума по методу условного градиента для задачи:

$$F = 4 \cdot x_1 + 3 \cdot x_2 \rightarrow \min,$$

$$(x_1 - 2)^2 + (x_2 + 2)^2 - 34 \leq 0.$$

3. Сформулировать штрафную функцию для задачи нелинейного программирования (по методу внешних штрафных функций):

$$F = 4 \cdot x_1 + 3 \cdot x_2 \rightarrow \max,$$

$$x_1^2 - 2 \cdot x_1 + x_2^2 - 2 \cdot x_2 - 34 \leq 0.$$

4. Сформулировать штрафную функцию для задачи нелинейного программирования (по методу внутренних штрафных функций):

$$F = 4 \cdot x_1 + 3 \cdot x_2 \rightarrow \max,$$
$$x_1^2 - 2 \cdot x_1 + x_2^2 - 2 \cdot x_2 - 34 \leq 0.$$

5. Сформулировать штрафную функцию для задачи нелинейного программирования (по методу внутренних штрафных функций):

$$(x_1 - 5)^2 + (x_2 - 3)^2 \rightarrow \min,$$
$$x_1 + x_2 \leq 3,$$
$$-x_1 + 2x_2 \leq 4.$$

6. Сформулировать штрафную функцию для задачи нелинейного программирования (по методу внешних штрафных функций):

$$x_2^3 - 3x_1x_2 + x_2^2 \rightarrow \min,$$
$$2x_1 + 5x_2 \leq 20,$$
$$2x_1 - 4x_2 = 10.$$

7. Составить схему алгоритма метода проекции градиента.
8. Составить схему алгоритма метода условного градиента.
9. Составить схему алгоритма метода внешних штрафных функций.
10. Составить схему алгоритма метода внутренних штрафных функций.

3 Динамическое программирование

Рассмотрим задачи оптимизации, в которых алгоритм принятия решений допускает разбиение на несколько этапов. В этом случае для построения решения (для реализации алгоритма) рекомендуется применять динамическое программирование (ДП). Метод динамического программирования основан на принципе оптимизации, сформулированном Р. Белманом [8, 9], состоящем в том, что конкретная задача оптимизации включается в семейство аналогичных многошаговых задач и сводится к рекуррентным соотношениям и функциональным уравнениям относительно оптимального значения целевой функции, их анализ позволяет последовательно получать решение для исходной задачи.

3.1 Принцип оптимальности Белмана

Допустим рассматривается задача, например распределение (продажа) ресурсов, поэтапное планирование инвестиций и т.д. – управляемая система S . Предположим, что процесс управления системой может быть разбит на m шагов и пусть S_i – состояние системы после i -ого шага. $i \in [0, m]$. Система S , под влиянием управления переходит из начального состояния S_0 в конечное состояние S_m . Последовательное преобразование системы достигается с помощью некоторых мероприятий x_1, x_2, \dots, x_m . (рисунок 3.1)

Обозначим показатель эффективности управления – W , а показатели эффективности на отдельном этапе (шаге) – через W_i , $i=1, m$. Полагая, что W обладает свойством аддитивности:

$$W = \sum_{i=1}^m W_i. \quad (3.1)$$

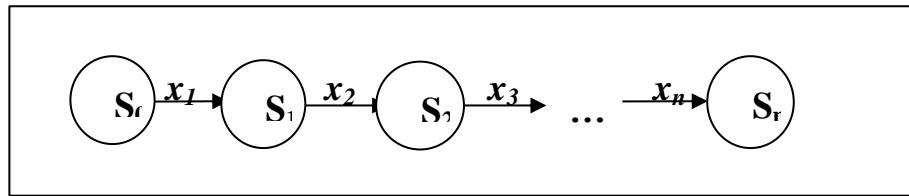


Рисунок 3.1 – Управляемая система

ОПРЕДЕЛЕНИЕ. Переменная x_i , определяющая выигрыш на i -м шаге (этапе), называется шаговым управлением.

ОПРЕДЕЛЕНИЕ. Управлением процесса в целом (за весь период – за m -этапов) называется вектор из последовательности шаговых управлений $x = (x_1, x_2, \dots, x_m)$.

Обозначим X – область допустимых управлений.

ОПРЕДЕЛЕНИЕ. Оптимальное управление $x^* = (x_1^*, \dots, x_m^*)$ – это вектор управления, при котором значение $W(x^*) = \sum_{i=1}^m W_i^*$ является максимальным:

$$W(x^*) = \max_{x \in X} W(x), \quad (3.2)$$

Для нахождения x^* используют принцип оптимальности Белмана, согласно которому управление на каждом шаге надо выбирать так, чтобы оптимальной была сумма выигрышей на всех оставшихся до конца процесса этапах (шагах), включая выигрыш на данном шаге.

Отметим, что при решении задачи ДП на каждом шаге выбирается управление, которое должно привести к оптимальному выигрышу, но при этом нельзя считать все шаги независимыми друг от друга. Доказательство от противного: если считать все шаги независимыми друг от друга, то оптимальным шаговым управлением будет то, которое приносит оптимальный доход (выигрыш) именно на данном шаге. Приведем пример опровергающий справедливость такого допущения: пусть осуществляется замена оборудования, но при покупке новой

техники взамен устаревшей затрачиваются средства и поэтому прибыль от эксплуатации вначале может быть небольшой, а впоследствии будет расти. Если же технику не заменить, то это в дальнейшем может привести к значительным убыткам. Пример демонстрирует то обстоятельство, что в многошаговых процессах все шаги зависят друг от друга и управление на каждом шаге надо выбирать с учетом его будущего воздействия на весь процесс.

Также при выборе шагового управления следует учитывать возможные варианты окончания предыдущего шага, определяющего состояние процесса.

Итак, при выборе шагового управления необходимо учитывать:

- 1) возможные исходы предыдущего шага;
- 2) влияние управления на все оставшиеся до конца процесса шаги.

Таким образом, ДП состоит в том, что на каждом шаге (этапе) алгоритма для всех ситуаций, соответствующих этому шагу (этапу), вычисляются накопленные значения вектора решения задачи (вектора управления), причем к следующему шагу допускаются лишь оптимальные значения в каждой ситуации предыдущего шага. При этом выполняется принцип оптимальности Белмана, сформированный выше, который можно сформулировать и так: оптимальная траектория решения задачи, состоящей из « m » шагов, должна следующим обладать свойством любая ее часть, состоящая из первых « k »-шагов ($k = 1, \dots, m$) также является оптимальной.

В задачах ДП первый пункт учитывают, делая на каждом шаге предположения о возможных вариантах окончания предыдущего шага и приводя для каждого из вариантов условную оптимизацию. Выполнение второго пункта обеспечивается проведением условной оптимизации от конца процесса к началу: первым оптимизируется последний « m » шаг, на котором надо учитывать возможные воздействия выбранного управления X_m на все следующие шаги (их нет). Выдвигая предположение об условиях окончания $(m-1)$ -го шага, для каждого из них проводят условную оптимизацию m -го шага и определяют условное оптимальное управление X_m .

Аналогично поступают для $(m-1)$ -го шага, выдвигаются предположения об

исходах окончания $(m - 2)$ -го шага и определяя условное управление на $(m - 1)$ -м шаге, приносящее оптимальный выигрыш на двух последних шагах – $(m - 1)$ -м и $(m) - m$, и т.д.

На первом шаге не требуется (обычно) выдвигать условных предположений, поскольку состояние системы перед этим шагом, как правило, известно.

Для этого состояния выбирают оптимальное шаговое управление, дающее оптимальный выигрыш на первом и всех последующих шагах. В итоге определяются оптимальные значения выигрыша и безусловные оптимальные управления на всех шагах.

3.2 Моделирование задачи ДП

Введем обозначения:

s – состояние процесса;

S_i – множество возможных состояний процесса перед i -м шагом;

W_i – выигрыш с i -го шага до конца процесса $i = \overline{1, m}$.

Выделим особенности моделей динамического программирования:

- задача оптимизации интерпретируется как n -шаговый процесс управления;
- целевая функция равна сумме функция каждого шага;
- выбор управления на i -м шаге зависит только от состояния системы к этому шагу, не влияет на предшествующие шаги (отсутствие обратной связи);
- состояние S_i после i -го шага управления зависит только от предшествующего состояния S_{i-1} и управления X_i (отсутствие последействия);
- на каждом шаге управление X_i зависит от конечного числа управляющих переменных, а состояние S_i от конечного числа параметров.

Основные этапы построения модели ДП:

1. Разбиение задачи на этапы (шаги). Шаг выбирают не слишком мелким

(чтобы расчеты не были слишком громоздкими) и не слишком большим (чтобы избежать усложнения процесса шаговой оптимизации).

2. Выбор переменных, характеризующих состояние s – моделируемого процесса перед каждым шагом и выявление налагаемых на них ограничений. В качестве таких переменных следует выбирать факторы, представляющие интерес для исследователя, например, годовую прибыль при планировании деятельности предприятия, размер инвестиций и т.п.

3. Определяем множества шаговых управлений $X_i, i = \overline{1, m}$ и налагаемых на них ограничений, т.е. области допустимых управлений X .

4. Определение выигрыша

$$\varphi(x_i), \quad (3.3)$$

который принесет на i -м шаге управление x_i , если система перед этим находилась в состоянии s .

5. Определение состояния s' , в которое переходит система из состояния s под влиянием управления x_i .

$$s' = f_i(s, x_i), \quad (3.4)$$

где f_i – функция перехода из состояния s в состояние s' .

6. Составление уравнения, определяющего условный оптимальный выигрыш на последнем шаге, для состояния s моделируемого процесса.

$$\begin{aligned} W_m(s) &= \varphi_m(s, x_m), \\ x_m &\in X. \end{aligned} \quad (3.5)$$

7. Составление основного функционального уравнения ДП, определяющего

условий оптимальный выигрыш для данного состояния S_i с i -го шага и до конца процесса через уже известный условий оптимальный выигрыш с $(i + 1)$ -го шага и до конца процесса.

$$W_i(s) = \varphi_i(s, x_j) + W_{i+1}f_i(s, x_j), \quad (3.6)$$

$$x_j \in X.$$

В уравнении (3.6) в уже известную функцию $W_{i+1}(s)$ характеризующую условный оптимальный выигрыш с $i + 1$ -шага до конца процесса вместо состояния s , поставляют новое состояние $s' = f(s, x_i)$, в которое система переходит на i -м шаге под влиянием управления x_i .

3.3 Этапы решения задачи ДП

Пусть построена математическая модель задачи ДП. Опишем этапы ее решения.

1. Определение множества возможных состояний S_m для последнего шага.
2. Проведение условной оптимизации для каждого состояния $s \in S_m$ на последнем m -м шаге по формуле (3.5) и определение условного оптимального управления $x(s), s \in S_m$.
3. Определение множества возможных состояний s_j для i -го шага, $i = 2, 3, \dots, m - 1$.
4. Проведение условной оптимизации i -го шага, $i = 2, 3, \dots, m - 1$ для каждого состояния $s \in S_i$ по формуле (3.6) и определении условного оптимального управления $x_i(s), s \in S_i, i = 2, 3, \dots, m - 1$.

5. Определение начального состояния системы S_1 , оптимального выигрыша $W_1(s_1)$ и оптимального управления $x_1(s_1)$ по формуле (3.5) при $i=1$ это и есть оптимальный выигрыш всей задачи $W^* = W_1^*(x_1^*)$.

6. Проведение безусловной оптимизации управления. Для чего найденное на первом шаге оптимальное управление $x_1^* = x_1(s_1)$ подставим в формулу (3.4) и определим следующее состояние системы $s_2 = f_2(s_1, x_1^*)$.

7. Для s_2 находим оптимальное управление $x_2^* = x_2(s_2)$ подставим его в (3.2) и т.д.

3.4 Оптимальное распределение инвестиций как задача ДП

Инвестор выделяет средства в размере D у.е., которые должны быть распределены между “ m ” предприятиями. Каждое i -е предприятие при инвестировании в него средств в размере x у. ед. приносит прибыль $\varphi_i(x)$ у.е., $i = \overline{1, m}$. Требуется выбрать оптимальное распределение инвестиций между предприятиями, обеспечивающее максимальную прибыль (т.е. максимальный выигрыш, приносимый “ m ” предприятиями).

Приведем математическую модель задачи. Пусть x_i – количество ресурса, выделяемое i -ому предприятию. Общая прибыль от вложений:

$$F = \sum_{i=1}^m \varphi_i(x_i) \rightarrow \max \quad (3.7)$$

Ограничения на средства:

$$\sum_{i=1}^m x_i = D \quad (3.8)$$

Задача (3.7-3.8) являются одноэтапной. Чтобы решить ее методом динамического программирования, введем многошаговость:

I этап состоит в оценке эффективности выделения ресурсов на одно предприятие (например, на предприятие m);

II этап состоит в оценке эффективности выделения ресурсов на два предприятия (например, на предприятие m и $m-1$ -е);

...

m этап – состоит в оценке эффективности выделения ресурсов на все предприятия.

Замечание: принято считать, что

1) прибыль i -ого предприятия не зависит от вложения средств в другие предприятия;

2) прибыль каждого предприятия выражается в одинаковых условных единицах;

3) суммарная прибыль равна сумме прибыли от каждого предприятия.

Запишем соотношения Беллмана для задачи распределения ресурсов.

Управлением i -ого шага будут являться решения об объеме ресурса, выделяемого i -ому предприятию.

Пусть $W_i(S)$ – максимальный доход, получаемый при распределении ресурса в объеме S между i предприятиями. Уравнения Беллмана для этой задачи имеют вид:

$$W_i(S) = \max_{0 \leq x_i \leq S} \{ \varphi_i(x_i) + W_{i-1}(S - x_i) \}, \quad i = \overline{1, m}, \quad \forall S = \{0, 1, \dots, D\}, \quad (3.9)$$

$$W_m(S) = \max_{0 \leq x_m \leq S} \{ \varphi_m(x_m) + W_{m-1}(S - x_m) \}. \quad (3.10)$$

Пример 3.1

Пусть планируется деятельность трех промышленных предприятий на очередной год. Конкретно $D = 4$ тыс. у.е. распределяется между $m=3$ получателями в количестве кратном $h = 1$ тыс. у.е. В результате выделения средств i -му

предприятию в размере x - тыс. у.е. получатель принесет инвестору прибыль $\varphi_i(x)$, $i = \overline{1, m}$ в соответствии с таблицей 3.1.

Таблица 3.1 – Распределение доход от инвестиций

| x - тыс. у.е. | $\varphi_1(x)$ тыс. у.е. | $\varphi_2(x)$ тыс. у.е. | $\varphi_3(x)$ тыс. у.е. |
|-----------------|--------------------------|--------------------------|--------------------------|
| 0 | 0 | 0 | 0 |
| 1 | 5 | 4 | 7 |
| 2 | 9 | 8 | 9 |
| 3 | 11 | 12 | 10 |
| 4 | 12 | 14 | 11 |

1. Число шагов $m = 3$ – число предприятий.

2. Определяем состояние системы. Состояние системы на каждом шаге характеризуется количеством средств s , имеющихся в наличии перед данным шагом, $s \leq D$.

3. Выбираем шаговые управления. Управлением на i -м шаге x_i , $i = \overline{1, m}$ является количество средств, инвестируемых в i -е предприятие.

4. Функция выигрыша на i -м шаге $\varphi_i(x_j)$ – это прибыль, которую приносит i -е предприятие при инвестировании в него средств x_j . При этом общий выигрыш есть:

$$W = \sum_{i=1}^m \varphi_i(x_j). \quad (3.11)$$

5. Определяем функцию перехода в новое состояние

$$f_i(s, x) = s - x. \quad (3.12)$$

Это означает, что если на i -м шаге и система находилась в состоянии s , а выбрано управление x , то на $(i+1)$ -м шаге система будет находиться в состоянии $s-x$. Иначе, если в наличии имеются средства s и в i -е предприятие инвестируется сумма в размере x , то для дальнейшего инвестирования остается $(s-x)$ у.е.

6. Составляем уравнения для $i=m$:

$$W_m(s) = \varphi_m(s), \quad (3.13)$$

$$x_m(s) = (s), \quad (3.14)$$

т.е. перед инвестированием средств на последнее предприятие, условное оптимальное управление соответствует количеству средств, имеющихся в наличии. Таким образом, сколько средств осталось, столько и надо вложить в последнее предприятие.

7. Далее составляем основное функциональное уравнение подставив в (3.4) выражение (3.5) и (3.6):

$$\begin{aligned} W_i(s) &= \max(\varphi(x) + W_{i+1}(s-x)), \\ x &\leq s. \end{aligned} \quad (3.15)$$

Это означает, что если перед i -м шагом у инвестора остались средства в S у.е., тогда x у.е. он может вложить в i -е предприятие и получить доход $\varphi_i(x)$, оставшиеся $(s-x)$ у.е. в остальные предприятия с $(i+1)$ до m -го. Условный оптимальный выигрыш от такого вложения $W_{i+1}(s-x)$. Оптимальным будет то условное управление x при котором сумма $\varphi_i(x)$ и $W_{i+1}(s-x)$ максимальна.

Вернемся, к примеру, представленному таблицей 3.1.

1. Проведем условную оптимизацию для последнего шага $i=3$.

В первой колонке записываются возможные значения состояния системы s , в верхней строке номера шага $i=3,2,1$. На каждом шаге определяются условные оптимальное управление $x_i(s)$ и условные оптимальные выигрыши.

Составляем таблицу 3.2, исходя из того, что $W_3(s) = \varphi_3(s)$.

Таблица 3.2 – Распределение дохода от инвестиций при $i=3$

| s | i=3 | |
|---|----------|----------|
| | $x_3(s)$ | $W_3(s)$ |
| 1 | 1 | 7 |
| 2 | 2 | 9 |
| 3 | 3 | 10 |
| 4 | 4 | 11 |

При $i=3, s=4$: $W_3(4) = 11, x_3(4) = 4$.

Условная оптимизация для $i=2$. Функциональное уравнение:

$$W_2(s) = \max(\varphi_2(x) + W_3(s - x)), \quad x \leq s. \quad (3.16)$$

Варьируем s и строим вспомогательные таблицы.

Для $s=1$ распределение дохода от инвестиций представлено в таблице 3.3.

На основании расчетов, приведенных в таблице при $i=2, s=1$:
 $\max(7,4) = 7 \rightarrow W_2(1) = 7, x_2(1) = 0$.

Таблица 3.3 – Распределение дохода от инвестиций при $i=2, s=1$

| x | 1-x | $\varphi_2(x)$ | $W_3(1-x)$ | $\varphi_2(x) + W_3(1-x)$ |
|---|-----|----------------|------------|---------------------------|
| 0 | 1 | 0 | 7 | $0+7=7$ |
| 1 | 0 | 4 | 0 | $4+0=4$ |

Для $s=2$ распределение дохода от инвестиций представлено в таблице 3.4.

Таблица 3.4 – Распределение дохода от инвестиций при $i=2, s=2$

| x | 2-x | $\varphi_2(x)$ | $W_3(2-x)$ | $\varphi_2(x) + W_3(2-x)$ |
|---|-----|----------------|------------|---------------------------|
| 0 | 2 | 0 | 9 | $0+9=9$ |
| 1 | 1 | 4 | 7 | $4+7=11$ |
| 2 | 0 | 8 | 0 | $8+0=8$ |

При $i=2, s=2$: $\max(9,11,8) = 11, W_2(2) = 11, x_2(2) = 1$.

Для $s=3$ распределение дохода от инвестиций представлено в таблице 3.5.

Таблица 3.5 – Распределение дохода от инвестиций при $i=2, s=3$

| x | 3-x | $\varphi_2(x)$ | $W_3(3-x)$ | $\varphi_2(x) + W_3(3-x)$ |
|---|-----|----------------|------------|---------------------------|
| 0 | 3 | 0 | 10 | 0+10=10 |
| 1 | 2 | 4 | 09 | 4+9=13 |
| 2 | 1 | 8 | 7 | 8+7=16 |
| 3 | 0 | 12 | 0 | 12-10=0 |

При $i=2, s=3$: $\max(10,13,16,0) = 16, W_2(3) = 16, x_2(3) = 2$.

Для $s=4$ распределение дохода от инвестиций представлено в таблице 3.6.

Таблица 3.6 – Распределение дохода от инвестиций при $i=2, s=4$

| x | 4-x | $\varphi_2(x)$ | $W_3(4-x)$ | $\varphi_2(x) + W_3(4-x)$ |
|---|-----|----------------|------------|---------------------------|
| 0 | 4 | 0 | 11 | 11 |
| 1 | 3 | 4 | 10 | 14 |
| 2 | 2 | 8 | 9 | 17 |
| 3 | 1 | 12 | 7 | 19 |
| 4 | 0 | 14 | 0 | 14 |

При $i=2, s=4$: $\max(11,14,17,19,14) = 19, W_2(4') = 19, x_2(4) = 3$.

Проведем условную оптимизацию при $i=1$. Перед первым шагом состояние системы известно $S = D=4$ тыс. у.е. и следовательно варьировать S не нужно, т.е. $S=4$. Распределение дохода от инвестиций при $i=1, s=4$ приведено в таблице 3.7.

При $i=1, s=4$: $\max(19,21,20,18,12) = 21, W_2(4^*) = 21, x_2(4) = 1$. Таким образом, оптимальная прибыль $W^* = W_1(4) = 21$.

Таблица 3.7 – Распределение дохода от инвестиций при $i=1, s=4$

| x | 4-x | $\varphi_2(x)$ | $W_3(4-x)$ | $\varphi_2(x) + W_3(4-x)$ |
|---|-----|----------------|------------|---------------------------|
| 0 | 4 | 0 | 19 | 19 |
| 1 | 3 | 5 | 16 | 21 |
| 2 | 2 | 9 | 11 | 20 |
| 3 | 1 | 11 | 7 | 18 |
| 4 | 0 | 12 | 0 | 12 |

Проводим безусловную оптимизацию:

$$i=1: S_1 = 4, W_1(4) = 21, x_1^* = x_1(4) = 1.$$

$$i=2: S_2 = S_1 - x_1^* = 4 - 1 = 3, W_2(3) = 16, x_2^* = x_2(3) = 2.$$

$$i=3: S_3 = S_2 - x_2^* = 1, W_3(1) = 7, x_3^* = x_3(1) = 1.$$

Итак, оптимальное управление $x^* = (1, 2, 1)$ т.е. в первое предприятие вкладываем 1000 у.е., во второе 2000 у.е., в третье 1000 у.е. и получаем максимальную прибыль $W^* = 21000$ у.е.

3.5 Задача о замене оборудования

Задача о замене оборудования состоит в определении оптимальных сроков замены старого оборудования (станков, производственных зданий и т.п.). Старение оборудования включает его физический и моральный износ. В результате чего увеличиваются производственные затраты, растут затраты на обслуживание и ремонт, снижается производительность труда и ликвидная стоимость. Критерием оптимальности является либо прибыль от эксплуатации оборудования, либо суммарные затраты на эксплуатацию. При построении модели принято считать, что решение о замене выполняется в начале каждого промежутка эксплуатации и, что в принципе оборудование можно использовать не ограниченно долго. Основная характеристика оборудования – это его возраст (t), параметр состояния. При составлении динамической модели процесс замены рассматривают, как n шаговый, разбивая весь период эксплуатации на n шагов. Возможное управление на каждом шаге характеризуется качественными признаками, например, x^c – сохранить оборудование; x^3 – заменить; x^p – сделать ремонт.

Рассмотрим решение этой задачи на примере.

Пример 3.2

Пусть оборудование эксплуатируется в течении четырех лет, после этого

продается. В начале каждого года можно принять решение сохранить оборудование или заменить его новым. Стоимость нового оборудования $p_0 = 3000$. После t лет эксплуатации ($1 \leq t \leq 4$) оборудование можно продать за $g(t) = \frac{p_0}{t+1}$ (ликвидная стоимость). Затраты на содержание в течение года $r(t) = 500(t+1)$. Определить оптимальную стратегию эксплуатации оборудования, чтобы суммарные затраты с учетом начальной покупки и заключительной продажи были минимальны.

Параметр состояния – возраст машины. В начале первого года эксплуатации машина новая $S_0 = 0, S_1 = 1, \dots, S_{k-1} = t$. Управление на каждом шаге зависит от двух переменных x^c – сохранить оборудование; x^3 – заменить. Уравнение состояния зависит от управления:

$$S_k = \begin{cases} t+1, x_k = x^c \\ 1, x_k = x^3 \end{cases}, k = 1, 2, 3.$$

Показатель эффективности k -ого шага зависит от управления и от возраста машины:

$$f_k(x_k, t) = \begin{cases} 500(t+1), x_k = x^c \\ 3000 + 500(0+1) - \frac{3000}{t+1} = 3500 - \frac{3000}{t+1}, x_k = x^3 \end{cases}, k = 1, 2, 3.$$

Пусть $W_k^*(t)$ – условные оптимальные затраты на эксплуатацию машины, начиная с k -ого шага до конца при условии, что к началу k -ого шага машина имеет возраст t :

$$W_4^* = \begin{cases} 500(t+1) - \frac{3000}{2t+1}, x_4 = x^c, \\ 3500 - \frac{3000}{t+1} - \frac{3000}{(t+1)+1} = 3500 - \frac{3000}{t+1} - \frac{3000}{t+2}, x_4 = x^3, \end{cases}$$

$$W_k^* = \min \begin{cases} 500(t+1) + w_{k+1}^*(t+1), x_k = x^c \\ 3500 - \frac{3000}{t+1} + w_{k+1}^*(t+1), x_k = x^3 \end{cases}, k = 3, 2, 1,$$

$$W_{\min}^* = W_1^*(0).$$

Приведем геометрическое решение. На оси абсцисс будем откладывать номер шага k . На оси ординат – возраст машины t . Точка $(k-1; t)$ соответствует началу k -ого периода эксплуатации машины возраста t лет.

Состояние начала эксплуатации соответствует точке S_0 с координатами $(0;0)$. Конец – точкам S с координатами $(4;t)$.

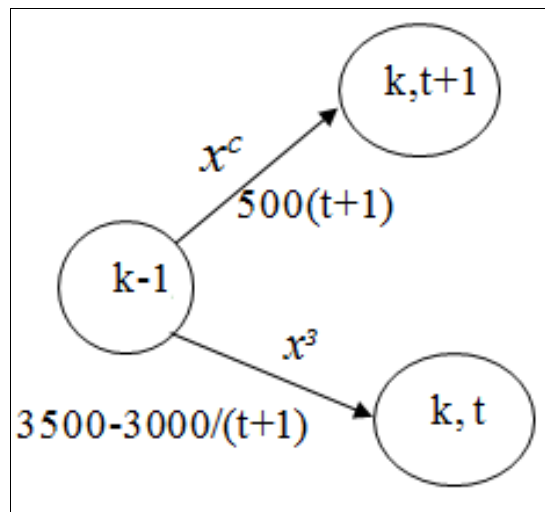


Рисунок 3.2 – Геометрическое представление точки $(k-1; t)$ задачи о замене оборудования

На первом этапе, начиная с точки $(0,0)$, над каждым отрезком, соединяющим состояния пишут соответствующие затраты, определяемые по (3.10). При $t=4$ по условию задачи оборудование продают, полученная прибыль от оборудования возраста t лет $g(t) = \frac{P_0}{t+1}$ записана в соответствующих состояниях внутри кружков.

Далее выполняют этап безусловной оптимизации, для этого начиная с состояний $t=4$ переходят к состояниям $t=3$, при этом вычисляют сумму затрат записанных в соответствующем состоянии и над стрелкой, минимальное значение записывают в

текущее состояние и стрелку, дающую минимум, помечают. Затем начиная с состояния $(0,0)$ выписывают оптимальные управления, следуя по помеченным стрелкам.

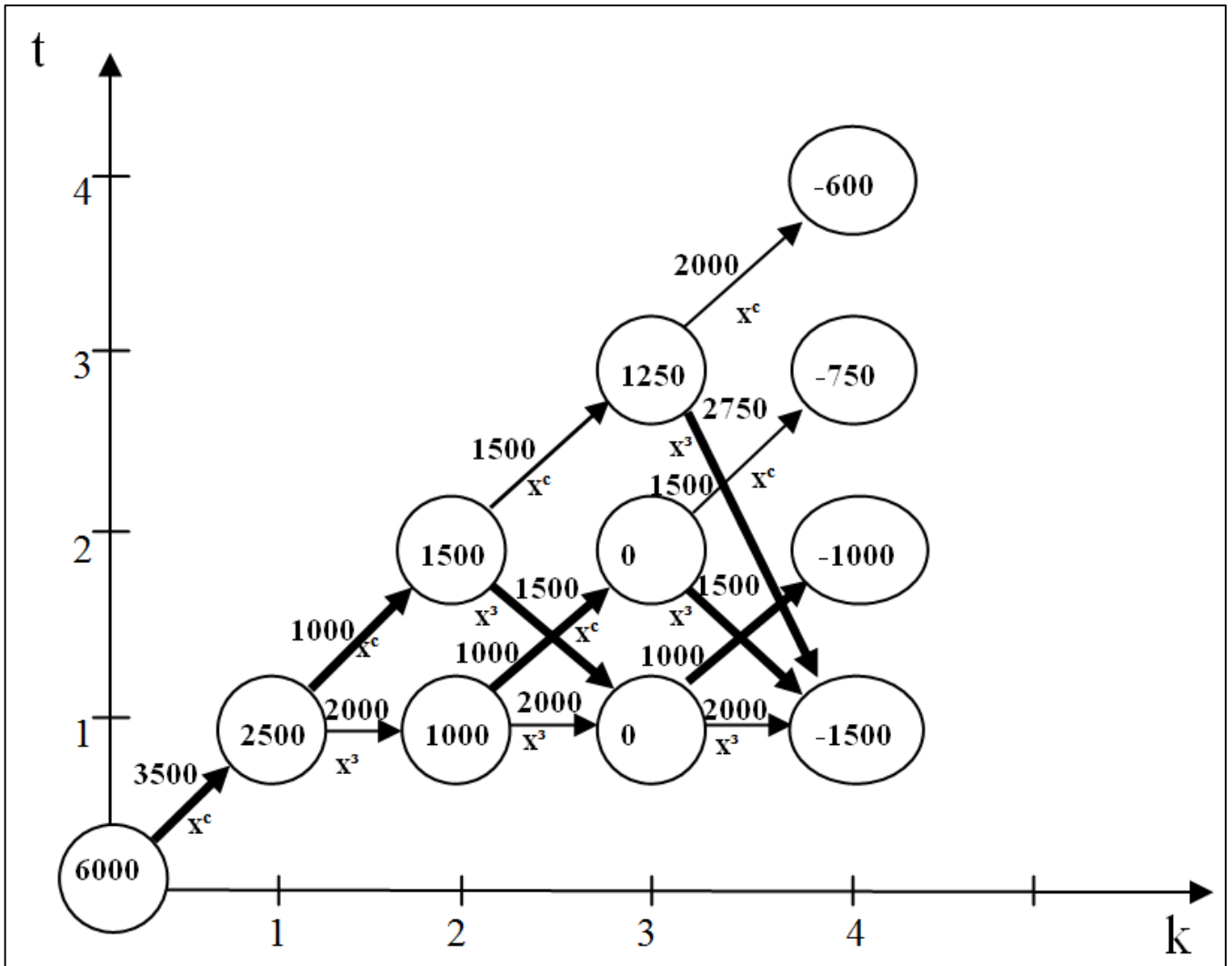


Рисунок 3.3 – Геометрическое представление решения задачи о замене оборудования

Таким образом, оптимальным управлением в рассматриваемой задаче, будет вектор (x^c, x^c, x^3, x^c) , или в 1 год: сохранить оборудование, 2 год – сохранить, 3 год – заменить, 4 год – сохранить. Суммарные затраты составят 6000 у.ед.

3.6 Задача о рюкзаке

Имеется транспортировочное средство грузоподъемностью W . Требуется заполнить его грузом, состоящим из предметов N различных типов таким образом, чтобы стоимость всего груза была максимальной.

Введем обозначения:

P_i – вес одного предмета i -ого типа

V_i – стоимость одного предмета i -ого типа

X_i – число предметов i -ого типа, которые будут загружаться на транспортировочное средство.

Нужно подобрать груз максимальной ценности с учетом грузоподъемности транспортировочного средства W :

$$\begin{aligned} W(x) &= \sum_{i=1}^N x_i V_i \rightarrow \max, \\ \sum_{i=1}^N x_i P_i &\leq S, \\ x_i &\in Z \text{ (множеству целых чисел)}. \end{aligned} \quad (3.17)$$

Решение задачи разбивается на n этапов, на каждом из которых определяется максимальная стоимость груза, состоящего из предметов 1-го типа (1-ый этап), 1-го и 2-го типов (2-ой этап) и т.д.

Приведем рекуррентное уравнение Беллмана для данной задачи. Пусть $W_i(C)$ – максимальная стоимость груза, состоящего из предметов типа $k = \overline{1, i}$, тогда

$$\begin{aligned} W_i(C) &= \max_{0 \leq x_i \leq \left\lfloor \frac{C}{P_i} \right\rfloor} \{x_i V_i + W_{i-1}(C - x_i P_i)\}, \quad \forall C: 0 \leq C \leq S, \\ W_0(C) &= 0, \quad \forall C: 0 \leq C \leq S, \end{aligned} \quad (3.18)$$

где $x_i \cdot V_i$ – стоимость взятых предметов i -ого типа;

$W_{i-1}(C - x_i P_i)$ – максимальная стоимость груза, состоящего из предметов типа $k = \overline{1, i-1}$ с общим весом не более $(C - x_i P_i)$.

Последовательно найдем значения функций $W_1(C), W_2(C) \dots, W_{N-1}(C), \forall C$, а также $W_N(S)$.

Пример 3.3

Пусть $S=10$ – грузоподъемность транспортного средства, которое необходимо заполнить предметами 4-х видов, информация о которых представлена в таблице 3.8.

Таблица 3.8 – Исходные данные о весе и ценности предметов

| | | | | |
|-------|----|----|----|---|
| i | 1 | 2 | 3 | 4 |
| p_i | 4 | 3 | 2 | 1 |
| v_i | 28 | 20 | 13 | 6 |

1 Шаг. Заполним транспортное средство только предметами 1-го вида, для этого решим задачу:

$$W_1(C) = \max_{0 \leq x_1 \leq \left[\frac{10}{4} \right]} \{x_1 \cdot 28\}, x_1 = 0, 1, 2.$$

Результаты этого этапа сведем в таблицу 3.9.

Таблица 3.9 – решение задачи о рюкзаке для $i=1$

| | | | |
|----------|-----|-----|------|
| C | 0-3 | 4-7 | 8-10 |
| $W_1(C)$ | 0 | 28 | 56 |
| x_1 | 0 | 1 | 2 |

2 Шаг. Заполним транспортное средство только предметами 1-го и 2-го типов, для этого решим задачу:

$$W_2(S) = \max_{0 \leq x_2 \leq \left\lfloor \frac{C}{3} \right\rfloor} \{x_2 \cdot 20 + W_1(C - X_2 \cdot 3)\}, x_2 = 0,1,2,3.$$

Результаты второго этапа сведем в таблицу 3.10.

Таблица 3.10 – Решение задачи о рюкзаке для $i=2$

| | | | | | | | | |
|----------|-----|----|-----|----|----|----|----|----|
| C | 0-2 | 3 | 4-5 | 6 | 7 | 8 | 9 | 10 |
| $W_2(C)$ | 0 | 20 | 28 | 40 | 48 | 56 | 60 | 68 |
| x_2 | 0 | 1 | 0 | 2 | 1 | 0 | 3 | 2 |

3 Шаг. Заполним транспортное средство только предметами 1-го, 2-го и 3-го типа, для этого решим задачу:

$$W_3(S) = \max_{0 \leq x_3 \leq \left\lfloor \frac{C}{2} \right\rfloor} \{x_3 \cdot 13 + W_2(C - X_3 \cdot 2)\}, x_3 = 0,1,2,3,4,5.$$

Результаты второго этапа сведем в таблицу 3.10.

Таблица 3.11 – Решение задачи о рюкзаке для $i=2$

| | | | | | | | | | | |
|----------|-----|----|----|----|----|----|----|----|----|----|
| C | 0-1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $W_3(C)$ | 0 | 13 | 20 | 28 | 33 | 41 | 48 | 56 | 61 | 69 |
| x_3 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

4 шаг. На последнем шаге будем планировать локально оптимальное решение для заполнения транспортного средства предметами всех 4-х типов:

$$W_4(S) = \max_{0 \leq x_4 \leq \left\lfloor \frac{S}{1} \right\rfloor} \{x_4 \cdot 6 + W_3(C - X_4 \cdot 1)\}, x_4 = 0,1,2,\dots,10.$$

Для последнего шага получили $W_4(10) = 69$ при $x_4 = 0$.

Выполним обратный ход: $x_4 = 0: W_4(10)$ при $x_4^0 = 0$
 $\Rightarrow W_3(10 - 0) = W_3(10) = 69$ при $x_3^0 = 1 \Rightarrow W_2(10 - 2 \cdot 1) = W_2(8) = 56$ при $x_2^0 = 0$
 $\Rightarrow W_1(8 - 0) = W_1(8) = 56$ при $x_1^0 = 2$.

Таким образом, оптимальное заполнение транспортного средства $x^0 = (2,0,1,0)$.

3.7 Вопросы для самопроверки

1. Сформулируйте принцип "отсутствия последействия".
2. Сформулируйте общую постановку задачи управления.
3. Каким требованиям должна удовлетворять задача, решаемая методом ДП?
4. Запишите математическую модель задачи распределения ресурсов.
5. Выпишите соотношения Беллмана для решения задачи распределения ресурсов.
6. Выпишите соотношения Беллмана для решения задачи о рюкзаке.
7. Сформулируйте постановку задач о замене оборудования.
8. Как проводится геометрическая интерпретация задачи о замене оборудования?

3.8 Задания для практических (лабораторных) занятий

Задание к лабораторным работам или практическим занятиям по теме: решить задачи методом динамического программирования.

Задача 3.1 Для развития трех предприятий выделено 6 млн. руб. Эффективность капитальных вложений в каждое предприятие, заданная функцией полезности $g_i(x)$ ($i=1, 2, 3, 4$). Составить оптимальный план распределения средств между предприятиями, предположив, что оно проводится в целых числах (0, 1, 2, 3, 4, 5 и 6 млн руб.). Исходные данные задачи приведены в таблице 3.12 (где N – номер варианта).

Таблица 3.12 – Исходные данные задачи 3.1

| x | 1 | 2 | 3 | 4 | 5 | 6 |
|----------|-------------------|-------------------|-----------------|-------------------|-----------------|-----------------|
| $g_1(x)$ | $4,1 \cdot N+2$ | $4 \cdot (N+1)$ | $5,1 \cdot N-2$ | $6,7 \cdot N-1$ | $7,0 \cdot N-2$ | $7,1 \cdot N+1$ |
| $g_2(x)$ | $4,0 \cdot (N+1)$ | $5,0 \cdot (N+2)$ | $5,5 \cdot N$ | $6,0 \cdot (N+1)$ | $8,0 \cdot N-2$ | $8,5 \cdot N-1$ |
| $g_3(x)$ | $3,1 \cdot N$ | $4,7 \cdot (N+1)$ | $5,3 \cdot N$ | $5,9 \cdot N+1$ | $6,5 \cdot N-1$ | $7 \cdot N-1$ |
| $g_4(x)$ | $2,1 \cdot N$ | $4,0 \cdot N+1$ | $5,9 \cdot N$ | $6,8 \cdot N+2$ | $7,5 \cdot N-1$ | $7,8 \cdot N-1$ |

Задача 3.2 Оборудование эксплуатируется в течении 6 лет, после этого продается. В начале каждого года можно принять решение сохранить оборудование или заменить новым. Стоимость нового оборудования p_0 рублей (для вариантов 3,4,7,9,11 стоимость нового оборудования зависит от периода k его покупки и составляет p_k рублей). После t лет эксплуатации ($1 \leq t \leq 6$) оборудование можно продать за $g(t)$ рублей (ликвидная стоимость). Затраты на содержание оборудования в течении года зависят от возраста t оборудования и равны $r(t)$. Определить оптимальную стратегию эксплуатации оборудования, чтобы суммарные затраты с учетом начальной покупки и заключительной продажи были минимальны. (к-год эксплуатации оборудования)

Таблица 3.13 – Варианты заданий к задаче 3.2 (вариант 1-14)

| вариант | p_0 | $g(t)$ | $r(t)$ |
|---------|-------------------------|----------------------|-----------------------------------|
| 1 | 6000 | $p_0 \cdot 2^{-t}$ | $0.2 \cdot p_0 \cdot (t+1)$ |
| 2 | 8000 | $p_0(2^{-t} + 1/10)$ | $0.3 \cdot p_0 \cdot (t+1)$ |
| 3 | $p_k = 5000 + 500(k-1)$ | $p_k 2^{-t}$ | $0.1 p_k (t+1)$ |
| 4 | $p_k = 3000 + 500(k-1)$ | $(p_k + 100)2^{-t}$ | $0.2 p_k (t+1)$ |
| 5 | 9000 | $8000-1000 \cdot t$ | $0.2 p_0 \cdot (t+1)$ |
| 6 | 10000 | $9000-750 \cdot t$ | $0.5 p_0 \cdot (t+1)$ |
| 7 | $p_k = 4000 + 700(k-1)$ | $(p_k + 200)2^{-t}$ | $0.2 p_k (t+1)$ |
| 8 | 15000 | $p_0 \cdot 2^{-t}$ | $0.1 \cdot p_0 \cdot (t+1)$ |
| 9 | $p_k = 4000 + 400(k-1)$ | $(p_k + 150)2^{-t}$ | $0.1 p_k (t+1)$ |
| 10 | 6000 | $p_0 \cdot 2^{-t}$ | $0.1 \cdot (p_0+100) \cdot (t+1)$ |
| 11 | $p_k = 4500 + 600(k-1)$ | $(p_k + 100)2^{-t}$ | $0.15 p_k (t+1)$ |
| 12 | 12000 | $p_0 \cdot 2^{-t}$ | $0.3 \cdot p_0 \cdot (t+1)$ |
| 13 | 4000 | $p_0 \cdot 2^{-t}$ | $0.1 \cdot p_0 \cdot (t+1)$ |
| 14 | 4500 | $p_0(2^{-t} + 1/2)$ | $0.15 \cdot p_0 \cdot (t+1)$ |

Таблица 3.14 – Варианты заданий к задаче 3.2 (вариант 15-20)

| вариант | p0 | k | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---------|----------|------|------|------|------|------|------|------|------|
| 15 | p0= 8000 | g(t) | - | 6000 | 5000 | 3000 | 1000 | 500 | 250 |
| | | r(t) | 600 | 800 | 1100 | 1500 | 2000 | 2100 | - |
| 16 | p0= 9000 | g(t) | - | 7000 | 6000 | 5000 | 4000 | 2000 | 1000 |
| | | r(t) | 500 | 700 | 900 | 1100 | 1300 | 2000 | - |
| 17 | p0= 7000 | g(t) | - | 5500 | 5000 | 3500 | 2000 | 1500 | 1250 |
| | | r(t) | 800 | 900 | 1100 | 1600 | 2200 | 2500 | - |
| 18 | p0= 6000 | g(t) | - | 4500 | 4200 | 3500 | 3000 | 2000 | 1500 |
| | | r(t) | 1500 | 1700 | 1900 | 2100 | 2300 | 2500 | - |
| 19 | p0= 7700 | g(t) | - | 6000 | 5500 | 3600 | 2500 | 1500 | 1200 |
| | | r(t) | 800 | 1000 | 1500 | 2100 | 2500 | 2800 | - |
| 20 | p0= 5000 | g(t) | - | 4500 | 4000 | 3500 | 3000 | 2000 | 1000 |
| | | r(t) | 600 | 750 | 900 | 1200 | 1300 | 1900 | - |

Задача 3.3 Имеются 6 видов предметов, для каждого из которых известна масса в килограммах v_j и стоимость p_j ($j=1,6$). Определить, какие предметы необходимо положить в рюкзак, чтобы их общая масса не превышала S килограммов, а общая ценность была максимальной.

Таблица 3.15 – Варианты заданий к задаче 3.3

| Вариант 1 | | | | | | |
|-----------|----|----|----|----|----|----|
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 6 | 5 | 4 | 3 | 1 | 1 |
| p_j | 48 | 30 | 28 | 23 | 15 | 10 |
| Вариант 2 | | | | | | |
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 6 | 5 | 4 | 3 | 1 | 1 |
| p_j | 28 | 20 | 26 | 23 | 15 | 14 |
| Вариант 3 | | | | | | |
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 6 | 5 | 4 | 3 | 2 | 1 |
| p_j | 40 | 36 | 28 | 25 | 18 | 10 |
| Вариант 4 | | | | | | |
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 7 | 5 | 4 | 3 | 2 | 1 |
| p_j | 50 | 40 | 28 | 20 | 18 | 10 |
| Вариант 5 | | | | | | |
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 10 | 8 | 7 | 5 | 3 | 1 |
| p_j | 52 | 40 | 32 | 26 | 20 | 10 |
| Вариант 6 | | | | | | |
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 8 | 6 | 5 | 4 | 2 | 1 |
| p_j | 48 | 30 | 28 | 23 | 15 | 10 |

Продолжение таблицы 3.15

| Вариант 7 | | | | | | |
|------------|----|----|----|----|----|----|
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 10 | 9 | 8 | 6 | 4 | 2 |
| p_i | 36 | 30 | 27 | 24 | 16 | 12 |
| Вариант 8 | | | | | | |
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 6 | 5 | 4 | 3 | 1 | 1 |
| p_i | 48 | 30 | 28 | 23 | 15 | 10 |
| Вариант 9 | | | | | | |
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 7 | 6 | 5 | 3 | 2 | 1 |
| p_i | 44 | 38 | 26 | 20 | 15 | 10 |
| Вариант 10 | | | | | | |
| - | 1 | 2 | 3 | 4 | 5 | 6 |
| v_j | 9 | 7 | 6 | 4 | 3 | 1 |
| p_i | 40 | 30 | 26 | 18 | 16 | 9 |

3.9 Задания для самостоятельной проработки и подготовке к контрольным работам

1. Решить задачу распределения ресурсов: между тремя предприятиями распределяются 3 млн. руб. Доход каждого предприятия зависит от выделенной суммы средств x . Значения дохода задаются в виде таблицы $g_i(x)$, $i = 1, 2, 3$. Найти такой план распределения 6 млн. руб. между предприятиями, при котором общий доход будет максимальным (средства могут распределяться в объеме кратном 1 млн. руб.).

Таблица 3.16 – исходные данные

| Средства x , млн. руб. | Доход предприятий, млн. руб. | | |
|--------------------------|------------------------------|----------|----------|
| | $g_1(x)$ | $g_2(x)$ | $g_3(x)$ |
| 0 | 0 | 0 | 0 |
| 1 | 10 | 5 | 14 |
| 2 | 23 | 20 | 21 |
| 3 | 35 | 38 | 34 |

2. В мебельном цехе деревообрабатывающий станок эксплуатируется в течение 5 лет, после чего продается. В начале каждого года руководство принимает решение сохранить станок или заменить новым аналогичным (при этом старый продается, а полученные средства направляются на покрытие части стоимости нового). Первоначальная стоимость нового станка составляет 138 000 руб. Известны затраты технического обслуживания станка в год ($r(t)$), и ликвидная стоимость оборудования – $g(t)$ (таблица 3.17).

Таблица 3.17 – Исходные данные задачи замены оборудования

| t | 0 | 1 | 2 | 3 | 4 | 5 |
|--------|-----|--------|-------|-------|-------|-------|
| $r(t)$ | 900 | 1200 | 1900 | 2500 | 3900 | – |
| $g(t)$ | – | 105000 | 90000 | 50000 | 40000 | 30000 |

Определить оптимальную стратегию (план-график) эксплуатации станка, обеспечивающую минимальные суммарные затраты на эксплуатацию в течение рассматриваемого периода в предположении, что цена нового станка каждый год увеличивается на величину инфляции 4,5%.

3. Студент собирается на выходные в однодневный поход и принимает решение, какие вещи взять с собой. Известна информация о стоимости и весе груза (таблица 3.18). Предложите наполнение рюкзака вместимостью 15 кг., имеющую максимальную стоимость.

| Название | Вес, кг. | Стоимость |
|---------------------|----------|-----------|
| Книга | 1 | 600 |
| Бинокль | 2 | 5000 |
| Котелок | 1 | 500 |
| Аптечка | 4 | 1500 |
| Внешний аккумулятор | 3 | 2000 |
| Сосиски | 1 | 150 |
| Спальник | 5 | 7000 |

Предложите свою собственную оценку полезности указанных предметов и составьте наполнение рюкзака, максимизирующего полезность взятых с собой предметов.

4 Модели и методы многокритериальной оптимизации

4.1 Математическая модель задачи многокритериальной оптимизации

В большинстве практических задач принятия решений возникают ситуации, когда решение должно быть принято с учетом множества критериев. Например, при определении лучшего кандидата на некоторую должность должны быть учтены следующие требования: опыт работы, образование, возраст, знание иностранных языков, пол и т.д. Или же при покупке жилого дома: стоимость, жилая площадь, удаленность от социальных объектов, наличие подъездных путей и т.п.

Если количество критериев, по которым производится сравнение исходов (возможных альтернатив, решений) больше 1, то такую задачу принятия решений принято называть многокритериальной. Основная сложность логического анализа многокритериальных задач состоит в том, что в них, в отличие от однокритериальных задач появляется эффект несравнимости исходов. Скажем, если исходы оцениваются по двум критериям, несводимым один к другому, и исход 1 лучше исхода 2 по первому критерию, но хуже по второму критерию, то исходы 1 и 2 будут несравнимыми между собой. Другой отличительной особенностью многокритериальных задач является то, что последствия выбора того или иного варианта решения могут быть оценены лишь на основе предыдущего опыта и интуиции.

Рассмотрим формальную постановку многокритериальной задачи [36].

Пусть D – произвольное множество допустимых решений, $f_1(x), f_2(x), \dots, f_m(x)$ – числовые функции, заданные на множестве D , $f(x) = (f_1(x), f_2(x), \dots, f_m(x))^T$ – векторный критерий.

$$f(x) \rightarrow \max(\min)_{x \in D}, \quad (4.1)$$

Или

$$\begin{aligned}
 f_1(x) &\rightarrow \max(\min) \\
 &\dots \\
 f_m(x) &\rightarrow \max(\min) \\
 x &\in D \quad D \subset \mathbb{R}^n.
 \end{aligned}
 \tag{4.1'}$$

Задача (4.1) называется задачей многокритериальной оптимизации.

ЗАМЕЧАНИЕ. Напомним, что любой частный критерий вида $f_i(x) \rightarrow \min$ можно заменить на критерий: $-f_i(x) \rightarrow \max$, $i = \overline{1, m}$.

Пример 4.1

Предприниматель покупает в одном месте мужские свитера (в количестве не более 60 штук), в другом – женские (не более 40 штук). С помощью мягкой щетки он делает начес и продает по цене 2 условные единицы за мужские и по 4 единицы за женские. За некоторый единичный интервал времени он может начесать не более 80 свитеров. Пусть закупочные цены в условных единицах таковы: мужские свитера по 1 у.е, женские по 3 у.е. Предприниматель хочет удержаться и на рынке мужских свитеров, и на рынке женских свитеров, и он интересуется не только максимумом прибыли, а также минимумом затрат.

Запишем математическую модель оптимизационной задачи. Пусть x_1 – объемы закупок мужских свитеров, x_2 – объемы закупок женских свитеров. Тогда критерии имеют вид:

$$\begin{aligned}
 f_1(x_1, x_2) &= 2x_1 + 4x_2 \rightarrow \max, \\
 f_2(x_1, x_2) &= x_1 + 3x_2 \rightarrow \min,
 \end{aligned}$$

где $f_1(x_1, x_2)$ – функция прибыли от продажи женских и мужских свитеров;

$f_2(x_1, x_2)$ – функция затрат на покупку женских и мужских свитеров;

Ограничения имеют вид:

$$\begin{cases} x_1 \leq 60 \\ x_2 \leq 40 \\ x_1 + x_2 \leq 80 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

ОПРЕДЕЛЕНИЕ. Идеальным в задаче (4.1) называется решение, образующее пересечение множеств оптимальных решений всех однокритериальных задач, т.е.

$$\bigcap_{i=1}^m \arg \max_{x \in D} f_i(x) = x^* .$$

Пример 4.2

Рассмотрим простейшую задачу многокритериальной оптимизации с множеством допустимых решений:

$$D = \{(x_1, x_2) : 1 \leq x_1 \leq 3, 2 \leq x_2 \leq 4\},$$

и критериями оптимальности:

$$\begin{aligned} f_1(x_1, x_2) &= x_1 \rightarrow \max, \\ f_2(x_1, x_2) &= x_2 \rightarrow \max. \end{aligned}$$

Геометрическая интерпретация решения поставленной задачи представлена на рисунке 4.1.

В точке с координатами (3; 4) одновременно достигается максимум по каждому частному критерию, поэтому задача имеет идеальное решение в точке (3; 4).

На практике случаи существования идеального решения в многокритериальной задаче крайне редки.

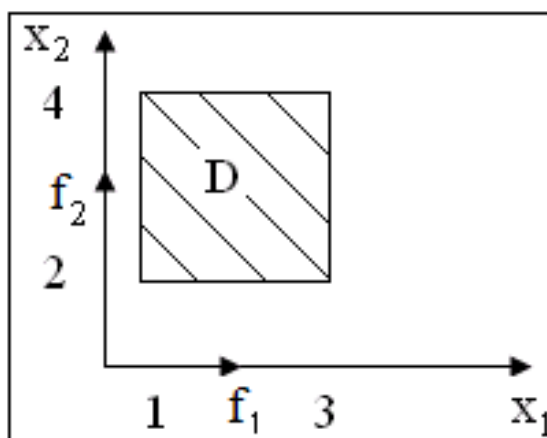


Рисунок 4.1 – Геометрическая интерпретация существования идеального решения в задаче многокритериальной оптимизации

Пример 4.3

Пусть задача многокритериальной оптимизации имеет вид:

$$f_1(x_1, x_2) = x_1 \rightarrow \min$$

$$f_2(x_1, x_2) = x_2 \rightarrow \min$$

$$(x_1 - 1)^2 + (x_2 - 1)^2 \leq 1.$$

Геометрическая интерпретация решения задачи представлена на рисунке 4.2.

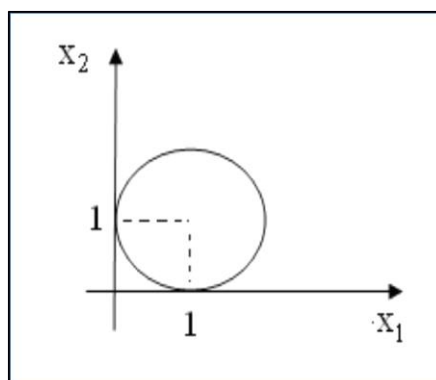


Рисунок 4.2 – Отсутствие существования идеального решения в задаче многокритериальной оптимизации

Поскольку $\arg \min_{x \in D} f_1(x) = (0; 1)$, $\arg \min_{x \in D} f_2(x) = (1; 0)$. Следовательно,

идеального решения в задаче не существует.

В связи с этим, основная проблема при рассмотрении многокритериальных задач состоит в формализации некоторого принципа компромисса, т.е. в определении того, в каком смысле одно допустимое решение лучше других.

4.2 Принцип доминирования по Парето

Не ограничивая общности рассуждений, будем считать, что все критерии в задаче (4.1) стремятся к максимуму.

Основное отношение, по которому производится сравнение допустимых решений – это отношение доминирования по Парето, которое определяется следующим образом.

ОПРЕДЕЛЕНИЕ. Пусть $x^1 = (x_1^1, \dots, x_n^1)$, $x^2 = (x_1^2, \dots, x_n^2)$ – некоторые допустимые решения (альтернативы или исходы) задачи многокритериальной оптимизации, а $z_j^k = f_j(x^k)$, $j = \overline{1, m}$, $k = 1, 2$ – критериальный вектор, состоящий из множества значений j -ого критерия оптимальности для k -ой альтернативы. Говорят, что альтернатива x^1 доминирует альтернативу x^2 по Парето (записывается $x^1 \succ x^2$), если для всех критериев $j = \overline{1, m}$ выполняется неравенство $z_j^1 \geq z_j^2$, причем по крайней мере для одного индекса j неравенство должно выполняться строго. В случае если все неравенства выполняются строго, то есть $z_j^1 > z_j^2 \quad \forall j$, имеем отношение доминирования по Слейтеру.

Очевидно, что при поиске решения в дальнейшем нет смысла сохранять доминируемую альтернативу x^2 , поскольку имеется исход с более лучшими значениями критериев. В результате такой процедуры отбрасывания заведомо непригодных, невыгодных альтернатив множество D сужается: в нём остаются только так называемые эффективные (иначе «Парето-оптимальные») исходы.

ОПРЕДЕЛЕНИЕ. Исход $x^* \in D$ называется Парето-оптимальным исходом во множестве D , если он не доминируется по Парето никаким другим исходом из множества D .

ОПРЕДЕЛЕНИЕ. Множество векторных оценок, соответствующих множеству эффективных исходов, называют областью компромиссов (переговорным множеством) или множеством (границей) Парето в области критериев.

Пример 4.4.

Дано 6 критериальных векторов, значения по всем критериям указаны в таблице 4.1. Необходимо указать отношение доминирования в исходах.

Таблица 4.1 – Значения критериальных векторов

| | f_1 | f_2 | f_3 |
|-------|-------|-------|-------|
| z^1 | -1 | 3 | 4 |
| z^2 | 2 | 4 | 6 |
| z^3 | 2 | 2 | 5 |
| z^4 | 3 | 2 | 5 |
| z^5 | 8 | 3 | -1 |
| z^6 | 8 | 3 | 0 |

Исход $x^2 \succ x^3$ доминирует по Парето (слабо), так как $z_j^2 \geq z_j^3$ $j = \overline{1,3}$ и аналогично $x^4 \succ x^3$.

Исход $x^2 \succ x^1$ доминирует (сильно) по Слейтеру ($z_j^1 > z_j^2, j = \overline{1,3}$).

Пусть имеется задача с двумя критериями:

$$\begin{aligned} f_1(x_1, x_2) &= x_1 \rightarrow \max \\ f_2(x_1, x_2) &= x_2 \rightarrow \max. \end{aligned} \tag{4.2}$$

Если в задаче (4.2) множество допустимых исходов является непрерывным,

то множество Парето-оптимальных оценок представляет собой «северо-восточную» границу области допустимых решений (рисунок 4.3, кривая BQ). В двумерном пространстве критериев эту линию называют компромиссной кривой. Она может состоять из несвязных кривых и (или) содержать изолированные точки.

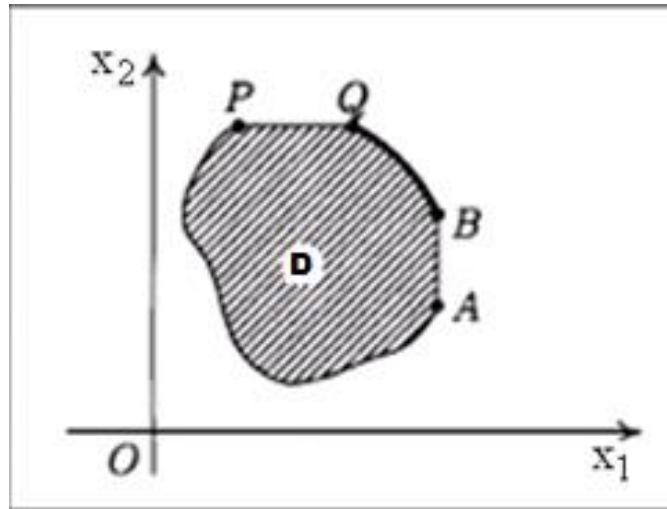


Рисунок 4.3 – Парето-оптимальное множество для непрерывного множества решений

Укажем простое геометрическое правило, посредством которого можно выделять из заданного плоского множества его границу Парето.

Рассмотрим прямой угол, стороны которого сонаправлены координатным осям x_1 и x_2 (рисунок 4.4).

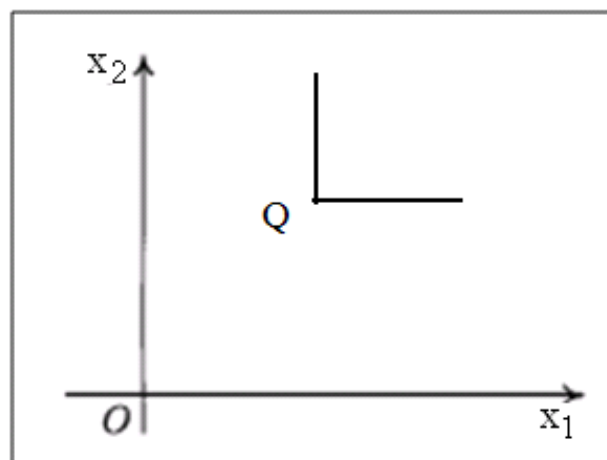


Рисунок 4.4 – Расположение прямого угла в пространстве координат x_1 и x_2

Положение этого угла на плоскости однозначно определяется его вершиной Q . Перемещая прямой угол (параллельно самому себе), будем собирать только те точки заданного множества допустимых решений, которые можно совместить с точкой Q так, чтобы ни одна другая точка множества D не попадала ни внутрь этого угла, ни на одну из его сторон (рисунок 4.2.4).

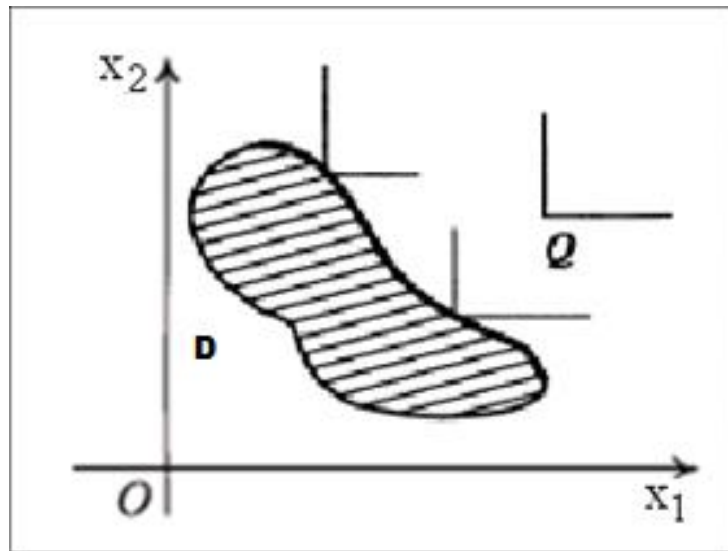


Рисунок 4.5 – Перемещение прямого угла

Совокупность всех таких точек и будет искомой границей Парето для множества D . Аналогичные рассуждения можно провести и для других задач с линейными критериями.

Пример 4.5

Пусть дана задача многокритериальной оптимизации следующего вида:

$$f_1(x) = -x_1 + 2x_2 \rightarrow \max,$$

$$f_2(x) = 2x_1 + x_2 \rightarrow \max,$$

$$f_3(x) = x_1 - 3x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4 \end{cases} .$$

Найдем множество решений, оптимальных по Парето. Для этого отобразим ОДЗ и вектор градиент для каждого критерия.

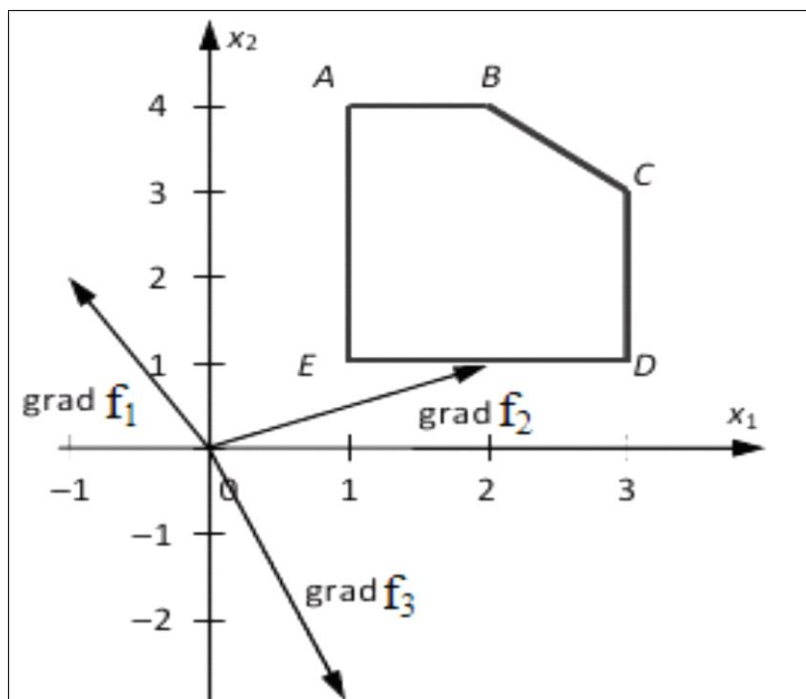


Рисунок 4.6 – Геометрическая интерпретация примера 4.5 (ОДЗ совпадает с множеством Парето оптимальных решений)

Очевидно, в данной задаче множество Парето совпадает с областью допустимых решений, т.е. с пятиугольником ABCDE (рисунок 4.6). Действительно, возьмем любую точку множества допустимых решений. Если мы от нее сдвинемся вправо, то значения критериев f_2 и f_3 увеличатся, значение критерия f_1 уменьшится. Если мы сдвинемся влево, то значения критериев f_2 и f_3 уменьшатся, но значение критерия f_1 увеличится. Если мы сдвинемся ниже, то значения критериев f_1 и f_2 увеличатся, но значение критерия f_3 уменьшится. Если мы сдвинемся выше, то значения критериев f_1 и f_2 уменьшатся, но значение критерия f_3 увеличится.

Таким образом, ни одна из точек множества допустимых решений не доминируется другими, т. е. все допустимые точки оптимальны по Парето.

Парето-оптимальность исхода x^* означает, что он не может быть улучшен ни по одному из критериев без ухудшения по какому-нибудь другому критерию. В связи с этим, в задаче многокритериальной оптимизации решение почти всегда ищется в области компромиссов.

Так как практически любая задача многокритериальной оптимизации допускает разные компромиссные решения, то и методы нахождения их решений многочисленны и весьма разнообразны.

Существующие методы решения многокритериальных задач можно классифицировать в две группы:

- методы, использующие дополнительную информацию о критериях и свойствах оптимального решения для сужения множества Парето-оптимальных исходов;

- интерактивные методы, в которых принятие решений происходит с участием ЛПР.

Согласно первой группе методов, сужение множества Парето-оптимальных исходов (в идеале до одного элемента) производится на основе некоторой дополнительной информации, что облегчает окончательный выбор исхода для лица, принимающего решение (ЛПР). Необходимо отметить, что получение дополнительной информации о критериях можно рассматривать как снятие неопределенности с них. При использовании второй группы выбор конкретного «оптимального» решения производится с учетом предпочтений ЛПР.

4.3 Методы решения многокритериальных задач, использующие дополнительную информацию о критериях и свойствах оптимального решения

Рассмотрим некоторые методы сужения Парето-оптимального множества более подробно

4.3.1 Метод оптимизации основного частного критерия

Пусть имеется задача многокритериальной оптимизации вида:

$$\begin{aligned} f_1(x) &\rightarrow \max, \\ &\dots \\ f_m(x) &\rightarrow \max, \\ x &\in D \quad D \subset \mathbb{R}^n. \end{aligned} \tag{4.3}$$

Решение принимается на основе частного критерия: для этого выделяют один максимизирующий критерий с номером k , а для остальных критериев задаются нижние границы γ_i (уровни достижения цели):

$$\begin{aligned} f_k(x) &\rightarrow \max, \\ f_i(x) &\geq \gamma_i, \quad i \neq k, \\ x &\in D \end{aligned} \tag{4.4}$$

Оптимальным при этом считается исход, максимизирующий выделенный критерий на множестве тех исходов, оценки которых по остальным критериям не ниже назначенных.

Пусть, например, f_1 – выбранный критерий, а по всем остальным критериям заданы нижние границы γ_j , $j=1,\dots,m$. Тогда оптимальным решением задачи многокритериальной оптимизации считается решение следующей задачи:

$$\begin{cases} f_1(x) \rightarrow \max, \\ f_i(x) \geq \gamma_i, \quad i = \overline{2, m}, \\ x \in D, \quad D \subset \mathbb{R}^n. \end{cases} \tag{4.5}$$

Пример 4.6

Пусть дана задача многокритериальной оптимизации следующего вида (условие примера 4.5):

$$f_1(x) = -x_1 + 2x_2 \rightarrow \max,$$

$$f_2(x) = 2x_1 + x_2 \rightarrow \max,$$

$$f_3(x) = x_1 - 3x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

Решим задачу с помощью метода субоптимизации. Допустим, в качестве главного критерия выбран первый критерий $f_1(x)$, а для остальных критериев заданы нижние границы $\gamma_2 = 4$ и $\gamma_3 = -8$. В итоге получим следующую задачу линейного программирования:

$$f_1(x) = -x_1 + 2x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4 \\ 2x_1 + x_2 \geq 4 \\ x_1 - 3x_2 \geq -8. \end{cases}$$

Решением поставленной задачи будет точка с координатами (1; 3).

С помощью метода субоптимизации задача многокритериальной оптимизации превращается в задачу однокритериальной оптимизации на суженном допустимом множестве. Выделение одного из критериев, а также указание нижних границ для остальных критериев основано на дополнительной информации, получаемой от ЛПР. Следовательно, окончательное решение здесь также носит субъективный характер.

4.3.2 Метод свертки критериев (построения обобщенного скалярного критерия)

Под сверткой критериев в многокритериальной задаче оптимизации (4.3) понимается такая процедура, которая осуществляет сведение многих критериев к одному путем введения весовых коэффициентов для каждого критерия. При этом более важный критерий получает больший вес.

Наиболее часто используется аддитивная (линейная) свертка критериев:

$$\begin{aligned} F(x) &= \sum_{i=1}^m \alpha_i \cdot f_i(x) \rightarrow \max, \\ x &\in D, \quad D \subset \mathbb{R}^n, \\ \alpha_i &\geq 0, \quad i = 1, m. \end{aligned} \quad (4.6)$$

Коэффициенты α_j называют весовыми коэффициентами. Иногда дополнительно требуют, чтобы $\sum_{j=1}^m \alpha_j = 1$.

Весовой коэффициент α_i интерпретируется как «показатель относительной важности» i -ого критерия: чем больше вес критерия α_i , тем большее влияние он окажет на окончательный результат отбора. Подбирая различные значения весовых коэффициентов, можно определенным образом воздействовать на процесс оптимизации, используя имеющуюся априорную информацию.

В качестве весового коэффициента можно выбрать $\alpha_i = \frac{1}{\max f_i(x)}$.

УТВЕРЖДЕНИЕ. Пусть $Q \in Z$ – произвольное множество векторных оценок.

Если векторная оценка x^* доставляет максимум функции $F(z) = \sum_{i=1}^m \alpha_i f_i(x)$, то

векторная оценка z^* является Парето оптимальным в множестве Q . Обратное

утверждение не всегда верно.

УТВЕРЖДЕНИЕ. Пусть $Q \in Z$ – выпуклое множество, $z^* \in Q$ – Парето оптимальная векторная оценка на множестве Q , тогда найдутся такие неотрицательные α_i , что $F(z) = \sum_{i=1}^m \alpha_i f_i(x)$ достигает максимум на множестве Q в точке z^* .

Помимо линейной, при решении задач многокритериальной оптимизации наиболее часто используются:

- мультипликативная свертка:

$$F(x) = \prod_{i=1}^m f_i^{\alpha_i}(x) \rightarrow \max, \quad (4.7)$$

- максиминная:

$$F(x) = \min_{j=1,m} f_j(x) \rightarrow \max, \quad (4.8)$$

а также другие различные способы свертки критериев, в том числе нелинейные.

Стоит отметить, что аддитивная свертка наиболее приемлема для задач, в которых критерии представляют собой однородные по смыслу и близкие по масштабу значений величины. При наличии существенно разнохарактерных частных критериев обычно бывает достаточно сложно указать окончательный набор весовых коэффициентов, исходя из неформальных соображений, связанных, как правило, с результатами экспертного анализа.

Применение мультипликативной свертки оправдано в том случае, если множество значений всех критериев неотрицательно. Кроме того, при использовании мультипликативной свертки следует учитывать, что если значение одного из критериев и (или) весового коэффициента равно нулю, то и мультипликативная свертка также равна нулю. В отличие от аддитивной, в мультипликативной свертке большее влияние оказывают те критерии, которые для

данного объекта имеют низкие значения.

Например, комбинируя «математическое ожидание прибыли по логнормальному распределению» и «математическое ожидание прибыли по эмпирическому распределению», естественно взять в качестве критерия их сумму. С другой стороны, для свертывания таких классов критериев, как «математическое ожидание прибыли» и «вероятность получения заданного значения прибыли» (по любому из распределений), лучше применять мультипликативную свертку.

На целевой функционал максиминной свертки оказывает влияние только тот частный критерий оптимальности, которому в данной точке x соответствует наименьшее значение соответствующей функции $f_j(x)$. И если в случае линейной свертки, вообще говоря, возможны «плохие» значения некоторых критериев за счет достаточно «хороших» значений остальных целевых функций, то в случае максиминного критерия производится расчет «на наихудший случай».

По значению функции (4.8) можно определить гарантированную нижнюю оценку для всех критериев $f_j(x)$. Этот факт расценивается как преимущество максиминного критерия перед методом линейной свертки.

Пример 4.7

Решим поставленную ранее задачу из примера 4.5, используя различные способы свертки критериев.

Так, например, при $\alpha_1 = 0,7$, $\alpha_2 = 0,2$ и $\alpha_3 = 0,1$ при использовании аддитивной свертки задача примет вид:

$$F(x) = 0,7 \cdot (-x_1 + 2x_2) + 0,2 \cdot (2x_1 + x_2) + 0,1 \cdot (x_1 - 3x_2) \rightarrow \max,$$
$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

В данном случае решением задачи будет точка $(1; 4)$, $F^*(x) = 5$.

При использовании максиминной свертки задача примет вид:

$$F(x) = \min \{-x_1 + 2x_2; \quad 2x_1 + x_2; \quad x_1 - 3x_2\} \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

Решением задачи будет точка $(2,5; 1)$, $F^*(x) = -0,5$.

4.3.3 Лексикографическая оптимизация

Этот метод предполагает, что критерии упорядочены по относительной важности: номер целевой функции отражает ранг соответствующего критерия. Процедура нахождения оптимального решения заключается в последовательной оптимизации по всем критериям в соответствии с их рангами. Для этого на первом шаге выбираются исходы, которые имеют максимальную оценку по критерию с первым рангом. Если такой исход единственный, то его и считают оптимальным. Если же таких исходов несколько, то среди них выбирают те, которые имеют максимальную оценку по критерию со вторым рангом и т.д. В результате такой процедуры всегда остается (по крайней мере, в случае конечного множества исходов) единственный исход – он и будет оптимальным.

Запишем алгоритм метода.

1. Находим $D^{1*} : \arg \min_{x \in D} f_1(x)$, где D^{1*} – множество всех допустимых решений из D^* , которое минимизирует $f_1(x)$.

2. Находим $D^{2*} : \arg \min_{x \in D^{1*}} f_2(x)$.

3. и так далее.

m. Находим $D^{m*} : \arg \min_{x \in D^{(m-1)*}} f_m(x)$.

Пример 4.8

Вернемся к примеру 4.5 и решим задачу с применением метода лексикографической оптимизации.

Допустим, имеется дополнительная информация о важности критериев: первый (критерий с первым рангом) – f_1 , за ним следует менее важный критерий f_2 с рангом два, затем последний критерий f_3 с рангом три.

На первом этапе решается задача:

$$f_1(x) = -x_1 + 2x_2 \rightarrow \max,$$
$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

Если решение не единственно, то решается задача:

$$f_2(x) = 2x_1 + x_2 \rightarrow \max,$$
$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

Если решение не единственно, то решается задача:

$$f_3(x) = x_1 - 3x_2 \rightarrow \max,$$
$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

В данном случае решением задачи будет точка (1;4), полученная на первом этапе.

Основными недостатками метода лексикографической оптимизации являются

следующие:

- при практическом применении метода возникают содержательные сложности в установлении полной упорядоченности критериев в соответствии с их рангами;

- фактически при использовании этого метода принимается во внимание только первый критерий с первым рангом. Например, следующий за ним по рангу критерий будет учитываться только тогда, когда первый критерий достигает максимума на нескольких исходах.

4.3.4 Метод последовательных уступок

Решение ищется также как и в методе лексикографической оптимизации, но с тем отличием, что осуществляется поиск не единственного точного оптимума, а некоторой области решений, близких к оптимальному – т.н. квазиоптимального множества). Это позволяет преодолеть недостатки метода лексикографической оптимизации.

Метод последовательных уступок включает следующие шаги:

1) расположить критерии $f_1(x), \dots, f_m(x)$ по их значимости (наиболее важный с точки зрения ЛПР располагается первым);

2) найти оптимальное значение f_1^* для целевой функции f_1 :

$$\begin{aligned} f_1(x) &\rightarrow \max \\ x &\in D; \end{aligned} \quad (4.9)$$

3) сделать уступку по первому показателю эффективности, т.е. ухудшить величину f_1^* до значения $f_1^{**} = f_1^* - \Delta_1$;

4) ввести в задачу дополнительное ограничение $f_1 \geq f_1^{**}$;

5) найти оптимальное значение f_2^* относительно целевой функции f_2 ;

$$\begin{aligned} f_2(x) &\rightarrow \max \\ f_1(x) &\geq f_1^* - \Delta_1 \\ x &\in D; \end{aligned} \quad (4.10)$$

6) сделать уступку по второму показателю эффективности, т.е. ухудшить величину f_2^* до значения $f_2^{**} = f_2^* - \Delta_2$;

7) ввести в задачу дополнительное ограничение $f_2 \geq f_2^{**}$;

8) новую задачу с двумя дополнительными ограничениями решить по третьему показателю эффективности

$$\begin{aligned} f_3(x) &\rightarrow \max, \\ \left\{ \begin{array}{l} f_1(x) \geq f_1^* - \Delta_1 \\ f_2(x) \geq f_2^* - \Delta_2 \\ x \in D. \end{array} \right. \end{aligned} \quad (4.11)$$

и т.д.

Процесс решения задачи заканчивается, когда решение будет получено по всем показателям.

Пример 4.9

Решим рассмотренный выше пример 4.5 с помощью метода последовательных уступок. Предположим, что критерии упорядочены согласно их нумерации.

Сначала решим задачу по первому критерию:

$$f_1(x) = -x_1 + 2x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

Получим ответ: $x=(1;4)$, $f_1^*=7$. Примем значение уступки $\Delta_1 = 2$. Введем в задачу дополнительное ограничение $-x_1 + 2x_2 \geq 7 - 2$ или после преобразований $x_1 - 2x_2 \leq -5$.

На втором этапе решается задача:

$$f_2(x) = 2x_1 + x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4 \\ x_1 - 2x_2 \leq -5. \end{cases}$$

Получим ответ: $x=(2,333;3,667)$, $f_2^* = 6$. Примем значение $\Delta_2 = 2$.

Введем в задачу дополнительное ограничение $2x_1 + x_2 \geq 6 - 2$ или $2x_1 + x_2 \geq 4$. Решим задачу

$$f_3(x) = x_1 - 3x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4 \\ x_1 - 2x_2 \leq -5 \\ 2x_1 + x_2 \geq 4. \end{cases}$$

Получим окончательный ответ: $x=(1;3)$, $f_3^* = -8$.

Метод последовательных уступок целесообразно применять для решения тех многокритериальных задач, в которых все частные критерии естественным образом упорядочены по степени важности.

Недостатком метода уступок является сложность подбора подходящих уступок, их выбор требует очень тщательного подхода. При задании слишком малых значений уступок возможна такая ситуация, что оптимизация по менее значимым критериям может быть вовсе не проведена, что не всегда будет устраивать ЛПР.

4.3.5 Метод идеальной точки

Обычно ЛПР формирует цель в виде желаемых значений показателей, при этом в качестве координат целевых функций он указывает сочетание наилучших значений критериев. Как правило, эта точка при заданных ограничениях не достигается, вследствие чего эта точка получила название идеальной точки или точки утопии.

ОПРЕДЕЛЕНИЕ. Под идеальной точкой понимают такой вектор $f^* = (f_1^*, \dots, f_m^*)$, компоненты которого находятся в результате решения следующих оптимизационных задач:

$$f_j^* = f_j(x) \rightarrow \max_{x \in D, D \subset \mathbb{R}^n} \quad i = \overline{1, m}, \quad (4.12)$$

Идеальная точка достижима только в том случае, когда идеальное решение многокритериальной задачи (4.1) существует (рисунок 4.7).

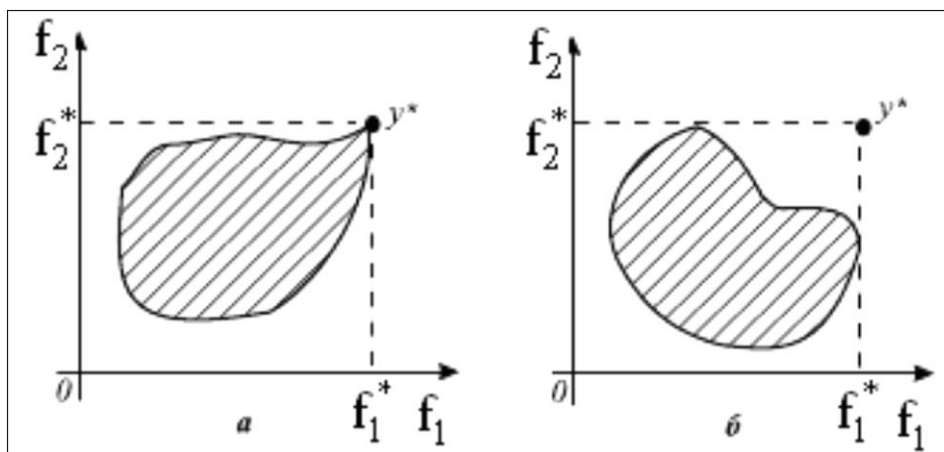


Рисунок 4.7 – Примеры множеств, в которых вектор f^* является а) достижимым и б) недостижимым

В основу метода идеальной точки положен поиск решения x из множества Парето, для которого расстояние от значений критериев до своих оптимальных значений минимально. Иначе говоря, от задачи (4.1) переходят к задаче вида:

$$\rho(f(x) - f^*) \rightarrow \min, \quad (4.13)$$

$$x \in D, D \subset \mathbb{R}^n$$

На практике наиболее часто используется обычное евклидово расстояние:

$$\sqrt{\sum_{j=1}^m (f_j(x) - f_j^*)^2} \rightarrow \min, \quad (4.14)$$

или взвешенное евклидово расстояние (при наличии дополнительной информации о весовых коэффициентах критериев $w_j, j=1, m$):

$$\sqrt{\sum_{j=1}^m w_j \cdot (f_j(x) - f_j^*)^2} \rightarrow \min. \quad (4.15)$$

Также могут использоваться:

- расстояние Чебышева:

$$\max_j |f_j(x) - f_j^*| \rightarrow \min ; \quad (4.16)$$

- взвешенное расстояние Чебышева:

$$\max_j \{w_j \cdot |f_j(x) - f_j^*|\} \rightarrow \min . \quad (4.17)$$

Также могут быть выбраны и любые другие метрики [2].

Пример 4.10

Найдем для примера 4.5 идеальную точку. Для нахождения идеального значения по первому критерию оптимальности решим задачу:

$$f_1(x) = -x_1 + 2x_2 \rightarrow \max, \\ \begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

Получим, что $f_1^* = 7$.

Для нахождения идеального значения по второму критерию оптимальности решим задачу:

$$f_2(x) = 2x_1 + x_2 \rightarrow \max, \\ \begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

Получим, что $f_2^* = 9$.

Для нахождения идеального значения по третьему критерию оптимальности решим задачу:

$$f_3(x) = x_1 - 3x_2 \rightarrow \max,$$
$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

Получим, что $f_3^* = 0$.

Таким образом, «идеальная» точка имеет координаты: $f^* = (7; 9; 0)$.

Применяя обычное евклидово расстояние, получим оптимизационную задачу:

$$\rho(x) = \sqrt{(7 + x_1 - 2x_2)^2 + (9 - 2x_1 - x_2)^2 + (0 - x_1 + 3x_2)^2} \rightarrow \min,$$
$$\begin{cases} x_1 + x_2 \leq 6 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 4. \end{cases}$$

решением которой будет точка: $x = (3; 2.286)$. Расстояние до «идеальной» точки составит 6,698.

Метод идеальной точки имеет ряд недостатков, наиболее серьезными из которых являются:

- расстояние между значениями по шкале одного критерия в общем случае не может отражать величину ценности альтернативы по данному критерию, которая является скорее функцией расстояния от идеальной альтернативы;

- альтернативы с одинаковыми функциями ценности могут находиться на различном расстоянии от «идеальной точки»;

- расстояние от «идеальной точки» может быть одинаковым при самых различных сочетаниях значений по отдельным критериям;
- не учитывается относительная важность критериев.

Достоинством метода является возможность отыскивать точки, лежащие на невыпуклой части границы Парето-оптимального множества решений.

Ни один из методов, представленных выше, не позволяет выделить единственное оптимальное решение. Решения, соответствующие различным наборам весовых коэффициентов, являются равноправными элементами Парето-эффективного множества решений. Однако с практической точки зрения, например, в задачах выбора вариантов (при покупке или заказе товаров, при выборе партнеров по бизнесу, при выборе вариантов программных средств и т.д.) часто требуется выбрать единственное решение. Для этого должна привлекаться некоторая дополнительная информация о предпочтениях лица, принимающего решения. В связи с этим перейдем к рассмотрению интерактивных методов решения задач многокритериальной оптимизации, в которых выбор окончательного решения осуществляется с учетом предпочтений ЛПР.

4.4 Интерактивные методы решения задач многокритериальной оптимизации

Наиболее распространенным подходом к решению многокритериальных задач на основе предпочтений ЛПР является построение некоторого функционала $U(x)$, определенного на множестве оценок альтернатив и позволяющего формально свести многокритериальную задачу к однокритериальной. Такой функционал называется функцией полезности, а проблема выбора заключается в выборе варианта с максимальной полезностью [23, 35].

Формально, постановка задачи заключается в следующем.

Пусть D – произвольное множество допустимых решений (альтернатив). Тогда идеальным решением многокритериальной задачи является отыскание решения следующей задачи:

$$\begin{aligned} U(x) \rightarrow \max, \\ x \in D, \quad D \subset R^n. \end{aligned} \quad (4.18)$$

Теоретически наиболее проработаны и практически наиболее часто используются методы построения функции полезности от значений частных критериев, имеющих аддитивную и мультипликативную структуру. При этом делается допущение о взаимной независимости критериев по полезности. Сама процедура построения функции осуществляется на основе результатов диалога с ЛПР и является достаточно трудоемкой задачей, поскольку число вопросов, требующих ответа ЛПР может оказаться непомерно большим [29].

Теория полезности строится исходя из следующих предпосылок.

1. ЛПР в состоянии сравнивать между собой любые две альтернативы из множества допустимых решений D , а именно, для каждой пары $x^1 \in D, x^2 \in D$ имеет место одно из трех соотношений:

$x^1 \prec x^2$ – набор x^1 предпочтительнее x^2 ;

$x^1 \succ x^2$ – набор x^1 менее предпочтителен, чем x^2 ;

$x^1 \sim x^2$ – для ЛПР оба набора обладают одинаковой степенью предпочтения.

2. Система предпочтений ЛПР удовлетворяет некоторым достаточно очевидным с позиций «здравого смысла» требованиям (аксиомам, определяющим основные свойства отношения предпочтения « \succ »).

Пусть на множестве D определена операция центра тяжести для любых двух оценок x^1 и x^2 :

$$x_{12} = \alpha x^1 + (1 - \alpha)x^2, \quad 0 < \alpha < 1, \quad (4.19)$$

тогда справедливы следующие аксиомы:

2.1) аксиома транзитивности: из одновременного выполнения условий $x^1 \prec x^2$ и $x^1 \succ x^2$, следует, что $x^1 \sim x^2$;

2.2) аксиома растворимости: из одновременного выполнения условий $x_2 \succ x_3 \succ x_1$, следует, что $\exists \alpha \in (0;1): x_{12} \sim x_3$;

2.3) архимедова аксиома: из одновременного выполнения условий $x_1 \succ x_3 \succ x_2$, следует, что $\exists \alpha \in (0;1): x_3 \succ x_{21}$;

3. Выполняются некоторые требования (аксиомы) «независимости» для частных критериев оптимальности. Выполнение условий независимости позволяет гарантировать существование функции полезности с определенными свойствами (например, свойством аддитивности), что существенно облегчает ее практическое построение.

Справедливо следующее утверждение, доказанное фон Нейманом и Моргенштерном.

ТЕОРЕМА. Если система предпочтений ЛПР удовлетворяет аксиомам 2.1 – 2.3, то существует скалярная функция (функционал) $U(x)$, которая ставит в соответствие каждому $x \in D$ такое вещественное число, что

$$\begin{aligned} x_1 \succ x_2 &\rightarrow U(x_1) > U(x_2); \\ x_1 \sim x_2 &\rightarrow U(x_1) = U(x_2). \end{aligned} \quad (4.20)$$

Необходимо отметить, что, несмотря на то, что теория полезности является удобным инструментом для решения многокритериальных задач оптимизации, при ее практическом использовании необходимо учитывать некоторые особенности. Во-первых, поскольку основным источником дополнительной информации при принятии решений является ЛПР, то использование теории полезности имеет принципиально субъективный характер. Во-вторых, в условиях многокритериальности выбора предполагается существование единственного «наилучшего» решения, который в действительности может не существовать. В

третьих, как уже отмечалось ранее, применение теории полезности становится затруднительным с ростом количества частных критериев.

Для работы с большим числом критериев получены результаты, позволяющие существенно сократить число проверок критериев на независимость их по предпочтению. Например, доказано, что для облегчения построения функции полезности удобно использовать частично аддитивные функции полезности. Другой рациональный подход связан с выделением естественных групп критериев, не связанных по смыслу между собой.

Например, в задаче выбора места для строительства какого-то промышленного объекта каждый вариант его размещения может характеризоваться разнородными показателями, определяющими набор частных критериев. Такие показатели могут характеризовать денежные расходы (бюджет строительства), воздействие на экосистему региона, социально-политические аспекты. По-видимому, с достаточной долей уверенности можно утверждать, что перечисленные показатели являются взаимно независимыми. А это позволяет надеяться на существование аддитивной функции полезности для этих трех групп критериев. Здесь подразумевается нахождение на некотором верхнем уровне иерархической системы критериев, описывающих решаемую проблему. Установив независимости представленных показателей на данном уровне, на следующем уровне конкретизируют представление об оптимальности, вводя соответствующие группы частных критериев [38].

Другим популярным подходом к решению многокритериальных задач является метод анализа иерархий (МАИ), разработанный американским математиком Томасом Саати в 1970 году.

МАИ не предписывает ЛПР, какого-либо «правильного» решения, а позволяет ему в интерактивном режиме найти такой вариант (альтернативу), который наилучшим образом согласуется с его пониманием сути проблемы и требованиями к ее решению.

В основе МАИ лежит декомпозиция проблемы на более простые составляющие, которая позволяет понятным и рациональным образом

структурировать сложную проблему принятия решений в виде иерархии, сравнить и выполнить количественную оценку альтернативных вариантов решения посредством многокритериального ранжирования [25].

Решение задачи с использованием МАИ состоит из четырех основных этапов:

1) первый этап заключается в формализации задачи в виде иерархической структуры с несколькими уровнями: цель, критерии, альтернативы;

2) на втором этапе эксперты или ЛПР выполняют попарные сравнения элементов каждого уровня. Результаты сравнений представляются в виде совокупности матриц парных сравнений;

3) на основе полученных матриц парных сравнений вычисляются коэффициенты важности для элементов каждого уровня. При этом проверяется согласованность суждений экспертов или ЛПР при помощи индекса согласованности;

4) подсчитывается итоговый вес каждой из альтернатив и определяется наилучшая альтернатива или они ранжируются.

Первый этап заключается в построении иерархии. Известно, что в основе процесса познания человеком окружающей действительности лежат декомпозиция и синтез. При изучении какой-либо системы, человек производит ее декомпозицию на подсистемы, и затем, выявив отношения между подсистемами, производит ее синтез. Декомпозиция и синтез используются в МАИ для создания структуры задачи принятия решений – иерархии. Построение такой структуры помогает проанализировать все аспекты проблемы и глубже проникнуть в суть задачи.

В вершине иерархии (рисунок 4.8) располагается основная цель, которую хочет достичь исследователь в результате решения проблемы. На последующих уровнях находятся подцели или факторы, которые оказывают влияние на главную цель. Если рассматриваемый фактор является сложным, то его можно разделить на более простые составляющие. И, наконец, на самом нижнем уровне располагаются альтернативы, среди которых производится выбор (и) или ранжирование. Цель, подцели, альтернативы далее будем называть объектами или элементами иерархии.

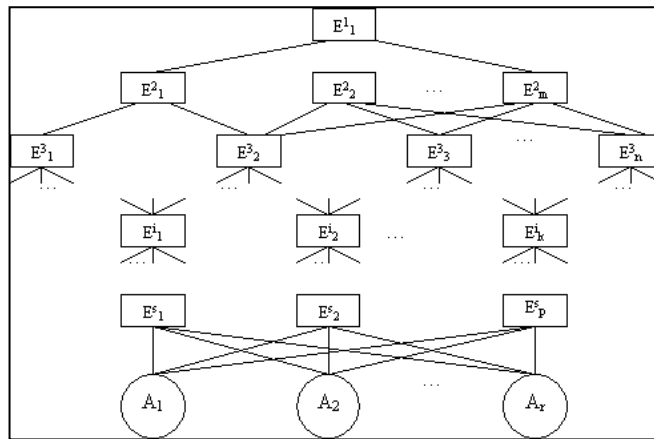


Рисунок 4.8 – Общий вид иерархического представления проблемы

На рисунке 4.8 приведен общий вид иерархии, где E^i_j – элементы иерархии; A_j – альтернативы; i – уровень иерархии; j – порядковый номер элемента.

Основное назначение иерархии в МАИ – оценка высших уровней иерархии исходя из взаимодействия ее различных уровней.

На втором этапе осуществляются парные сравнения отдельных компонент иерархии. Для этого в иерархии выделяются элементы двух типов: элементы-«родители» и элементы-«потомки». Элементы-«потомки» воздействуют на соответствующие элементы вышестоящего уровня иерархии, являющиеся по отношению к первым элементами-«родителями». Матрицы парных сравнений строятся для всех элементов-«потомков», относящихся к соответствующему элементу-«родителю». Таким образом, на втором этапе будут построены матрицы парных сравнений для элементов всех уровней иерархии, кроме последнего.

Для каждой матрицы оценивают значимость исследуемых элементов относительно всех элементов, находящихся на предыдущем уровне иерархии и проверяют согласованность мнений экспертов путем вычисления индекса согласованности и отношения согласованности. Более подробно процесс построения матриц парных сравнений и процедура проверки согласованности экспертных мнений описаны в [25].

На третьем этапе производится иерархический синтез, который предназначен для взвешивания собственных векторов матриц парных сравнений альтернатив

весами критериев (элементов), имеющих в иерархии.

На четвертом этапе решения задачи, после решения задачи иерархического синтеза, оценивается согласованность всей иерархии. Для этого суммируются показатели согласованности мнений экспертов всех уровней (рассчитываемых по матрицам парных сравнений), приведенных путем «взвешивания» к первому иерархическому уровню, где находится корневая вершина. Более подробную информацию об алгоритме иерархического синтеза можно найти в [25].

Достоинствами МАИ являются его универсальность и простота. Однако этот метод имеет ряд очевидных недостатков. Одним из главных недостатков является необходимость построения большого числа матриц парных сравнений и определения большого числа оценок парных сравнений альтернатив и критериев. Другим недостатком является то, что МАИ работает только с жесткими оценками альтернатив и не допускает неопределенности в суждениях экспертов или ЛПР. Это условие не выполняется в большинстве случаев, так как эксперт обычно неточен и ненадежен.

Кроме того, очевидно, что МАИ является расширением обычной линейной свертки критериев при условии, что веса критериев и веса альтернатив (значений векторного критерия) вычисляются при помощи матриц парных сравнений. При нескольких уровнях критериев веса линейной свертки, образованной k -м уровнем критериев, определяются при помощи линейных сверток критериев $(k-1)$ -го уровня. Если имеется только один уровень критериев, то МАИ полностью совпадает с линейной сверткой критериев. Отсюда следует, что все недостатки, присущие линейной свертке, являются недостатками МАИ.

4.5 Нормирование критериев

Частные критерии имеют, как правило, разные единицы и масштабы измерения, и это делает невозможным их непосредственное сравнение. Операция

приведения критериев к единому масштабу и безразмерному виду называется нормированием. Для того чтобы обеспечить однородность частных критериев, на практике часто используют простые приемы эквивалентного преобразования неоднородных частных критериев к единому, безразмерному виду.

Наиболее распространенным способом нормирования является замена абсолютных значений критериев их:

1) безразмерными относительными величинами:

$$\bar{f}_i(x) = \frac{f_i(x)}{f_i^*}, i = \overline{1, m}; \quad (4.21)$$

2) относительными значениями отклонений от оптимальных значений критериев f_i^* :

$$\bar{f}_i(x) = \frac{f_i^* - f_i(x)}{f_i^*}, i = \overline{1, m}. \quad (4.22)$$

Успешное решение проблемы нормализации в данном случае во многом зависит от того, насколько правильно и объективно определены идеальные значения f_i^* . Рассмотрим основные способы выбора идеального вектора $f^* = (f_1^*; f_2^*; \dots; f_m^*)$, поскольку именно от него зависят результаты нормализации.

Способ 1: Идеальный вектор определяется известными эталонными значениями критериев (например, из международных стандартов):

$$f^* = (f_1^{\text{Эт}}; f_2^{\text{Эт}}; \dots; f_m^{\text{Эт}}).$$

Практическое использование этого способа затруднено поиском эталонных значений.

Способ 2: Идеальный вектор определяется экспертно:

$$f^* = (f_1^{\text{Эк}}; f_2^{\text{Эк}}; \dots; f_m^{\text{Эк}})$$

Недостатком этого способа является субъективность полученных оценок.

Способ 3. В качестве идеального вектора выбирают вектор, параметрами которого являются максимально возможные значения локальных критериев:

$$f^* = (f_{1\max}; f_{2\max}; \dots; f_{m\max}).$$

Недостатком этого способа является то, что компоненты вектора f^* необходимо определять в результате решения m задач одномерной оптимизации. К тому же полученные максимальные значения функций не должны быть равны нулю (в противном случае метод не применим), а при наличии отрицательных компонент вектора f^* поиск оптимального решения задачи может ошибочно вестись в другом направлении.

Способ 4. В качестве параметров идеального вектора принимают максимально возможный разброс соответствующих локальных критериев, т. е. $f^* = f_{\max}^* - f_{\min}^*$.

Для применения этого способа должны быть известны диапазоны изменения значений критериев.

4.6 Целевое программирование

4.6.1 Постановка задачи

Идея целевого программирования заключается в том, чтобы установить некоторый уровень достижения целей по каждому критерию.

Особенности целевого программирования:

- 1) любой критерий трактуется как цель;
- 2) достижению отдельных целей приписываются приоритеты;
- 3) вводятся переменные v_i^+ и d_i^- , являющиеся мерой отклонения от целевых или пороговых уровней сверху и снизу соответственно («бонус» и «штраф»);
- 4) минимизируется взвешенная сумма переменных отклонений с целью найти решения, наилучшим образом удовлетворяющее целям.

Постановка линейной задачи целевого программирования:

$$\begin{aligned} f_i(x_1, x_2, \dots, x_n) &= c_1^j x_1 + c_2^j x_2 + \dots + c_n^j x_n, \\ f_i(x) &= c^i \cdot x \quad i = \overline{1, m}. \end{aligned} \quad (4.23)$$

При ограничениях вида:

$$\begin{aligned} Ax &= b, \\ x &\in X \quad X \subset \mathbb{R}^n, \end{aligned} \quad (4.24)$$

где A – заданная матрица размерности $k \times n$ коэффициентов системы ограничений;
 b – известный вектор размерности k правых частей системы ограничений.

Выделим следующие типы целевых критериев:

1) целевая функция должна быть больше задаваемого порогового значения t_1 :

$$f_1(x) = c^1 \cdot x \geq t_1. \quad (4.25)$$

Функция полезности ЛПР для такой цели имеет вид, представленный на рисунке 4.9.

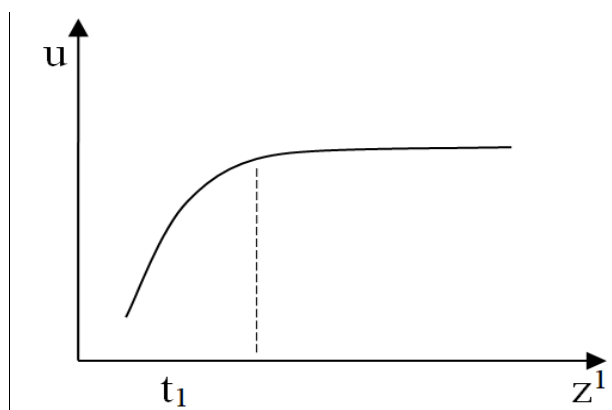


Рисунок 4.9 – Вид функции полезности ЛПР для первого типа задания цели

2) целевая функция не должна превышать заданного порогового значения t_2 :

$$f_2(x) = c^2 \cdot x \leq t_2. \quad (4.26)$$

Функция полезности ЛПР имеет вид, представленный на рисунке 4.10.

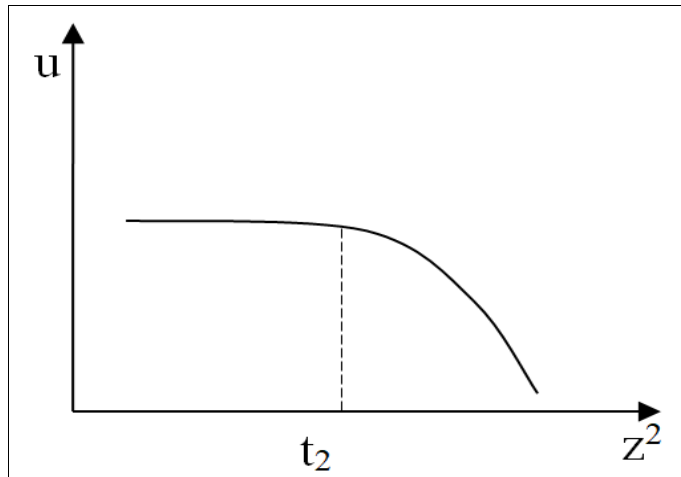


Рисунок 4.10 – Вид функции полезности ЛПР для второго типа задания цели

3) целевая функция должна совпадать с заданным пороговым значением t_3 :

$$f_3(x) = c^3 \cdot x = t_3. \quad (4.27)$$

Функция полезности ЛПР в этом случае имеет вид, представленный на рисунке 4.11.

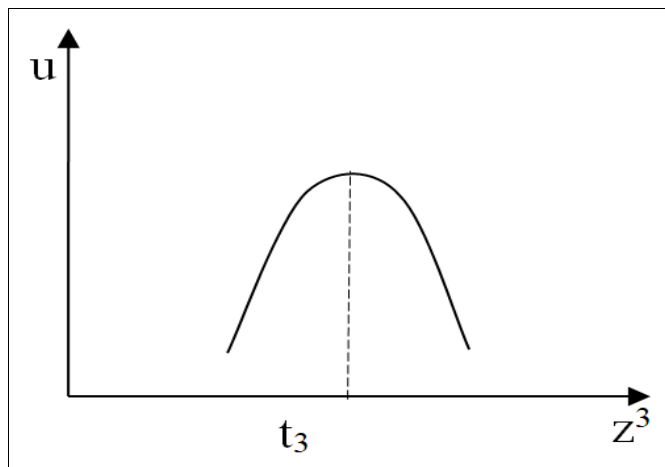


Рисунок 4.11 – Вид функции полезности ЛПР для третьего типа задания цели

4) значение целевой функции должно лежать в заданном интервале $[t_4^d, t_4^u]$:

$$f_4(x) = c^4 \cdot x \in [t_4^l, t_4^m]. \quad (4.28)$$

Это комбинация первого и второго типа. Функция полезности ЛПР в этом случае имеет вид, представленный на рисунке 4.12.

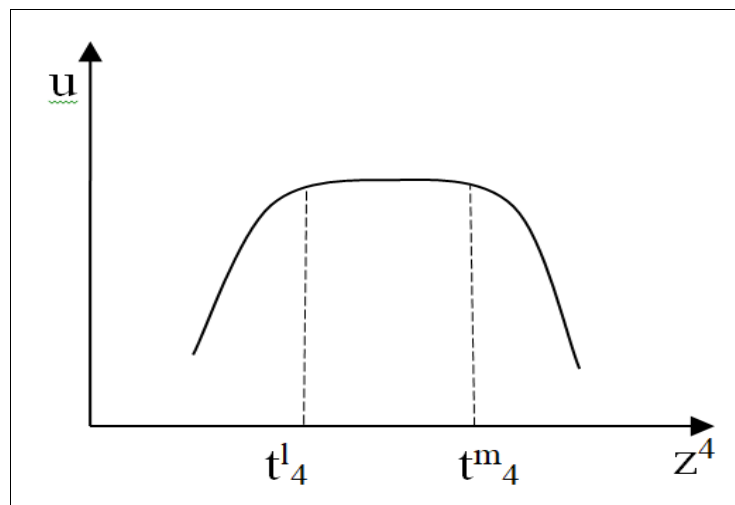


Рисунок 4.12 – Вид функции полезности ЛПР для третьего типа задания цели

Множество критериальных векторов, которые одновременно удовлетворяют всем целям будем называть утопическим множеством.

Будем искать допустимую точку, которая достигает всех целей как можно лучше. Методы отыскания таких точек составляют предмет целевого программирования.

В целевом программировании рассматривается два основных подхода к решению задачи:

- 1) архимедова модель;
- 2) модель с приоритетами.

Рассмотрим эти подходы подробнее.

4.6.2 Архимедова модель

Архимедова модель – координаты в решение генерируются путем определения тех точек из области допустимых решений, критериальные вектора для которых являются ближайшими в смысле взвешенной метрики l_1 утопическому множеству в пространстве критериев.

Расстояние между векторами x^k и x^l в архимедовой модели определяется по метрике l_1 :

$$d_{l_1}(x^k, x^l) = \sum_{i=1}^n (x_i^k - x_i^l). \quad (4.29)$$

В модели с приоритетами генерируются решения для которых критериальные вектора оказываются наиболее соответствующими в лексикографическом смысле точкам утопического множества.

Рассмотрим архимедову модель. Введем допущения модели.

1. Целевая функция представляет собой взвешенную сумму переменных нежелательных отклонений.

2. Каждая цель порождает одно целевое ограничение кроме случая с диапазоном, когда возникают два целевых ограничения.

3. Коэффициенты w_i^+ и w_i^- целевой функции представляют собой положительные штрафные веса. Они позволяют не штрафовать нежелательные отклонения от целей с разной степенью жесткости.

Предположим имеем 4 вида критериев (4.25-4.28), тогда архимедова модель имеет вид:

$$w_1^- d_1^- + w_2^+ d_2^+ + w_3^+ d_3^+ + w_3^- d_3^- + w_4^- d_4^- + w_4^+ d_4^+ \rightarrow \min, \quad (4.30)$$

$$\begin{cases} c^1x + d_1^- \geq t_1 \\ c^2x - d_2^+ \leq t_2 \\ c^3x - d_3^+ + d_3^- = t_3 \\ c^4x + d_4^- \geq t_4^d \\ c^4x - d_4^+ \leq t_4^u \end{cases} \quad (4.31)$$

$$x \in D, \quad (4.32)$$

$$d_1^-, d_2^+, d_3^+, d_3^-, d_4^+, d_4^- \geq 0. \quad (4.33)$$

Целевые ограничения являются нежесткими в том смысле, что они не ограничивают исходную область D в пространстве решений. В действительности они пополняют область допустимых решений, переводя D в пространство большей размерности и создавая таким образом расширенную (архимедову) область допустимых решений для задачи целевого программирования.

Будем изображать многокритериальную задачу в пространстве решений или в пространстве критериев.

Если используется пространство критериев, то демонстрируется достижимое множество в пространстве критериев z , которое определяется следующим образом $z = \{z \in \mathbb{R}^m \mid z = c \cdot x, x \in D\}$ достижимое множество, где m – количество критериев.

Пример 4.11

Пусть дана задача

$$f_1 = x_1 + \frac{1}{2}x_2,$$

$$f_2 = \frac{1}{2}x_1 + x_2,$$

$$\begin{cases} x_1 + x_2 \leq 5 \\ x_1 \leq 4 \\ x_{1,2} \geq 0. \end{cases}$$

Заданы следующие цели:

$$z_1 \geq 4 \text{ или } z_1 = x_1 + \frac{1}{2}x_2 \geq 4,$$

$$z_2 \in [4,5;6,5] \text{ или } \frac{1}{2}x_1 + x_2 \geq 4,5 \text{ и } \frac{1}{2}x_1 + x_2 \leq 6,5.$$

Отобразим на графике в пространстве x_1Ox_2 область допустимых решений (рисунок 4.13). Представим также утопическое множество, используя полученные неравенства.

Из рисунка 4.13 легко убедиться, что общих точек между утопическим множеством и ОДР нет.

В качестве решения необходимо найти точки расположенные ближе к утопическому множеству. Для этого представим задачу в пространстве критериев. Рассчитаем значение критериев во всех вершинах ОДР.

$$(0,0) = \begin{cases} 0 + \frac{1}{2} \cdot 0 \\ \frac{1}{2} \cdot 0 + 0 \end{cases} = (0,0);$$

$$(4,1) = \begin{cases} 4 + \frac{1}{2} \cdot 1 \\ \frac{1}{2} \cdot 4 + 1 \end{cases} = (4,5;3);$$

$$(0,5) = \begin{cases} 0 + \frac{1}{2} \cdot 5 \\ \frac{1}{2} \cdot 0 + 5 \end{cases} = (2,5;5);$$

$$(4,0) = \begin{cases} 4 + \frac{1}{2} \cdot 0 \\ \frac{1}{2} \cdot 4 + 0 \end{cases} = (4;2).$$

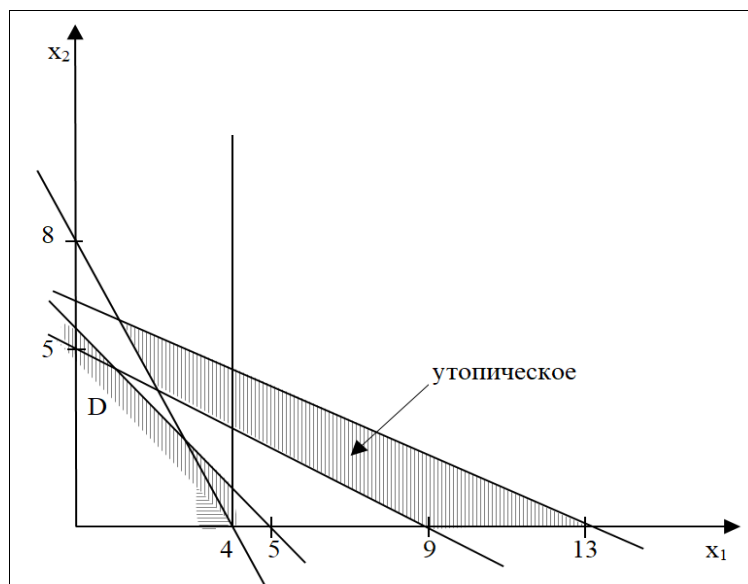


Рисунок 4.13 – ОДР и утопическое множество в пространстве переменных

На рисунке 4.14 представлена задача в пространстве критериальных векторов.

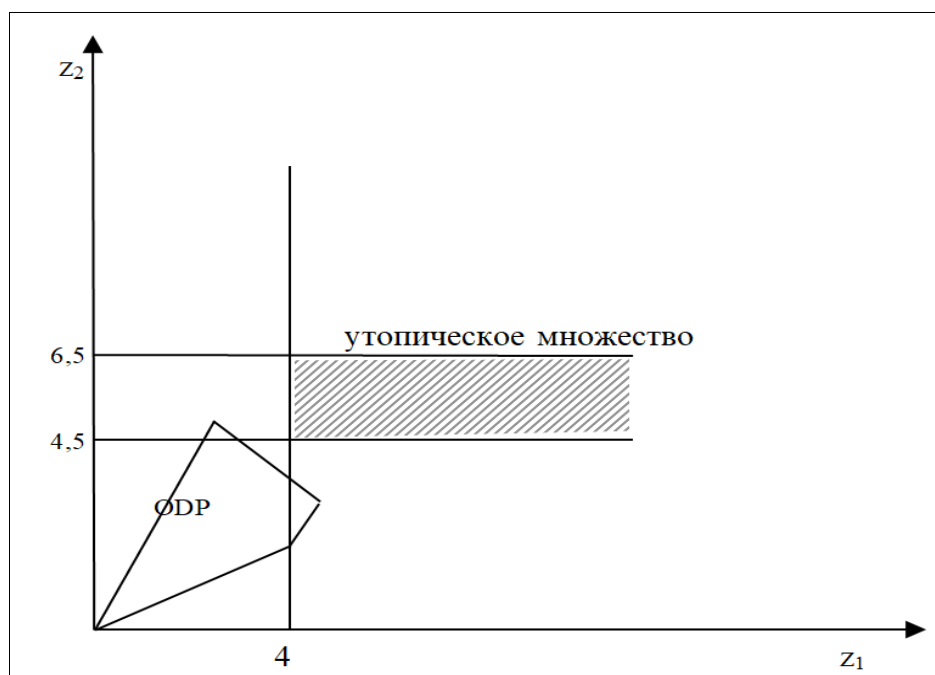


Рисунок 4.14 – ОДР и утопическое множество в пространстве критериев

Т.к. нет точек, которые были бы и допустимыми и удовлетворяли бы всем целям, то смысл целевого программирования – попытаться найти такую точку из области D, которая была бы наиболее близка к утопическому множеству, например, в смысле какой-нибудь метрики (рисунок 4.15).

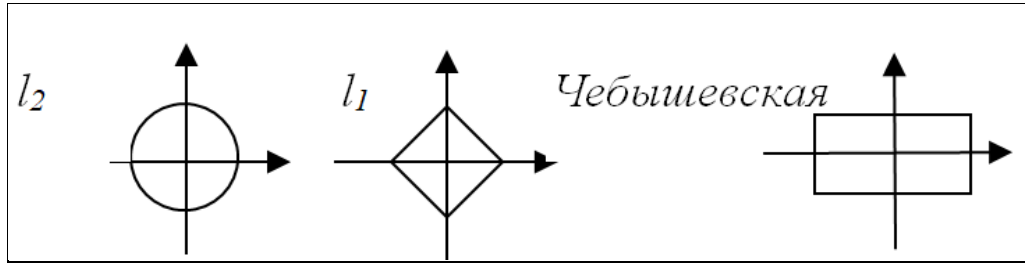


Рисунок 4.15 – Виды метрик

Архимедова задача целевого программирования относится к классу задач оптимизации в метрике l_1 , т.к. целевая функция представляет собой взвешенную сумму переменных нежелательных отклонений.

Пример 4.12

Рассмотрим следующие критерии:

$$z_1 = x_1,$$

$$z_2 = x_2.$$

Пусть заданы уровни достижения целей:

$$z_1 = t_1 = 8,$$

$$z_2 = t_2 = 7.$$

При ограничениях

$$\begin{cases} -x_1 + 2x_2 \leq 8 \\ x_1 + x_2 \leq 10 \\ x_{1,2} \geq 0. \end{cases}$$

Отобразим ОДР на рисунке 4.16. Утопическое множество это точка с координатами (7; 8). Архимедова модель задачи имеет вид:

$$\left\{ w_1^+ d_1^+ + w_1^- d_1^- + w_2^+ d_2^+ + w_2^- d_2^- \right\} \rightarrow \min,$$

$$\begin{aligned}
 x_1 + d_1^- - d_1^+ &= 8, \\
 x_2 + d_2^- - d_2^+ &= 7, \\
 \begin{cases} -x_1 + 2x_2 \leq 8 \\ x_1 + x_2 \leq 10 \end{cases} \\
 x_1, x_2 &\geq 0.
 \end{aligned}$$

Если $w_1^+ = w_1^-$, $w_2^+ = w_2^-$ и $w_1^+ = w_2^+$, то линии уровня целевой функции будут иметь вид, показанный на рисунке 4.16 (равносторонние ромбы).

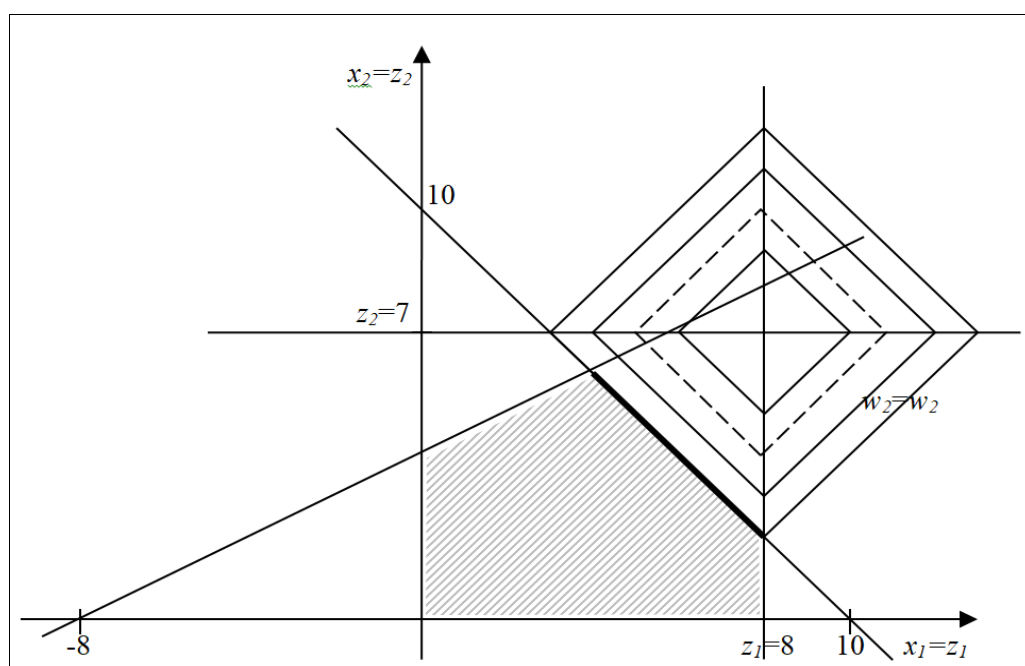


Рисунок 4.16 – ОДР, утопическое множество и линии уровня в архимедовой модели для случая равных весовых коэффициентов

В этом случае ответом будет отрезок заключенный между точками (4; 6) и (8; 2).

Если $w_1^+ = w_1^-$, $w_2^+ = w_2^-$ и $w_1^+ > w_2^+$, линии уровня примут вид аналогичный как на рисунке 4.17. В этом случае линии уровня целевой функции представляют собой горизонтально сжатые ромбы в точке (8,7). Наилучшим решением будет точка (8; 2).

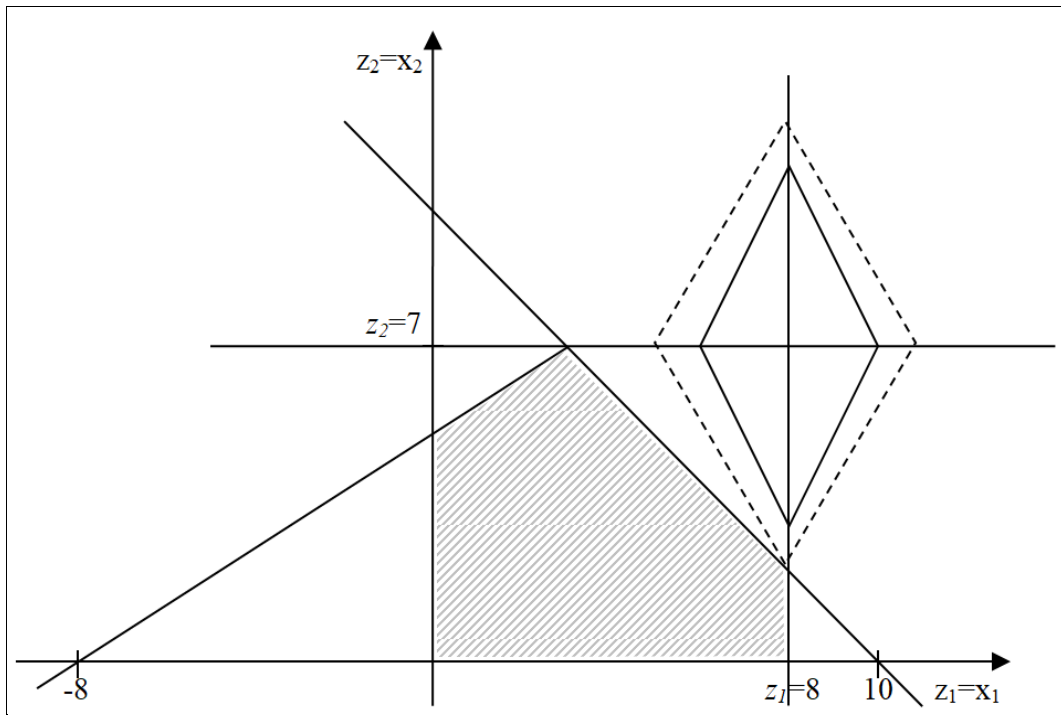


Рисунок 4.17 – ОДР, утопическое множество и линии уровня в архимедовой модели для случая $w_1^+ > w_2^+$

Если теперь $w_1^+ = w_1^-$, $w_2^+ = w_2^-$ и $w_1^+ < w_2^+$, получаем вертикально сжатый ромб (рисунок 4.18), лучшая точка (4,6).

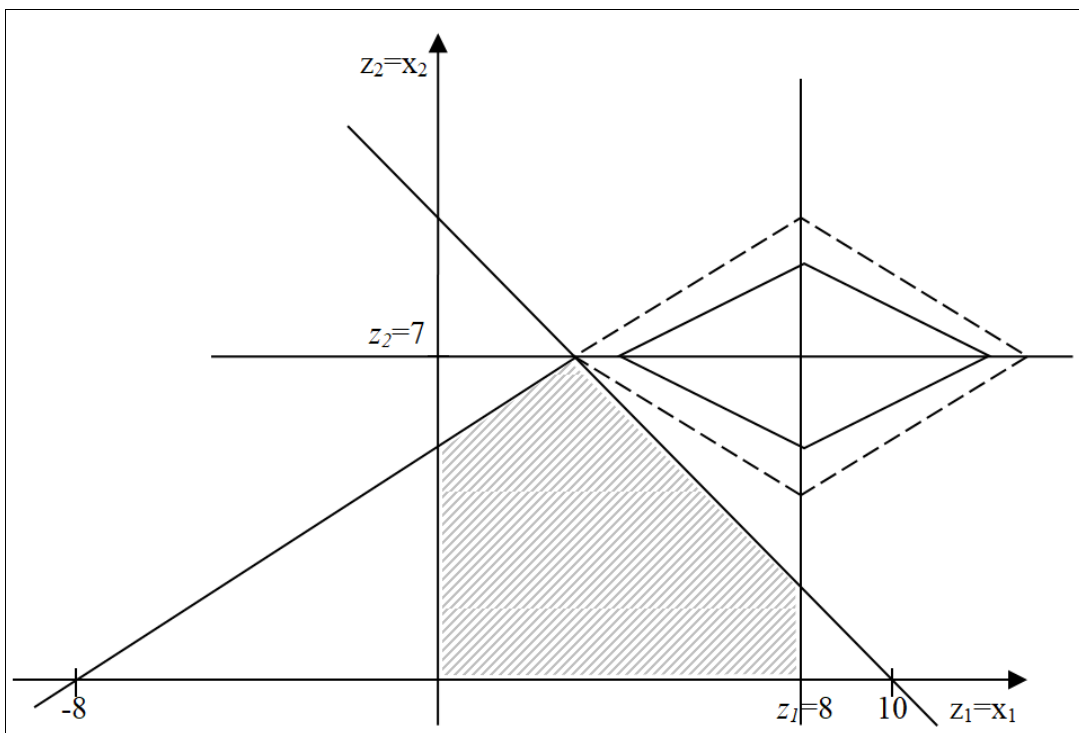


Рисунок 4.18 – ОДР, утопическое множество и линии уровня в архимедовой модели для случая $w_1^+ > w_2^+$

Пример 4.13

Рассмотрим линии уровня целевых функций, которые определяются двумя критериями, ограниченными по диапазону.

В случае одинаковых весов линии уровня будут иметь вид как на рисунке 4.19.

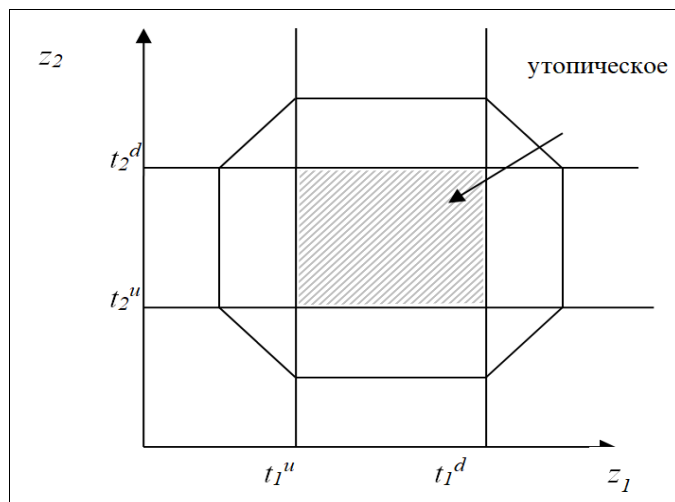


Рисунок 4.19 – ОДР, утопическое множество и линии уровня в архимедовой модели для примера 4.13

Пример 4.14

Рассмотрим линии уровня целевых функций, которые определяются двумя критериями $z_1 \geq t_1$ $z_2 = t_2$. В случае одинаковых весов линии уровня будут иметь вид как на рисунке 4.20.

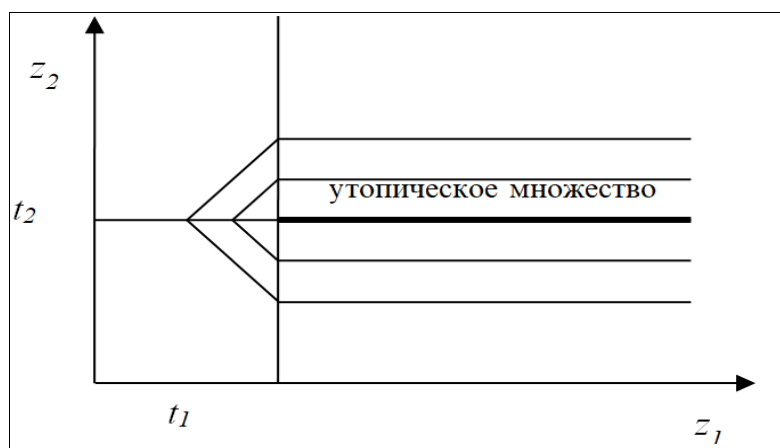


Рисунок 4.20 – ОДР, утопическое множество и линии уровня в архимедовой модели для примера 4.14

Пример 4.15

Рассмотрим линии уровня целевых функций, которые определяются двумя критериями $z_1 \geq t_1$ $z_2 \geq t_2$. В случае одинаковых весов линии уровня будут иметь вид как на рисунке 4.21.

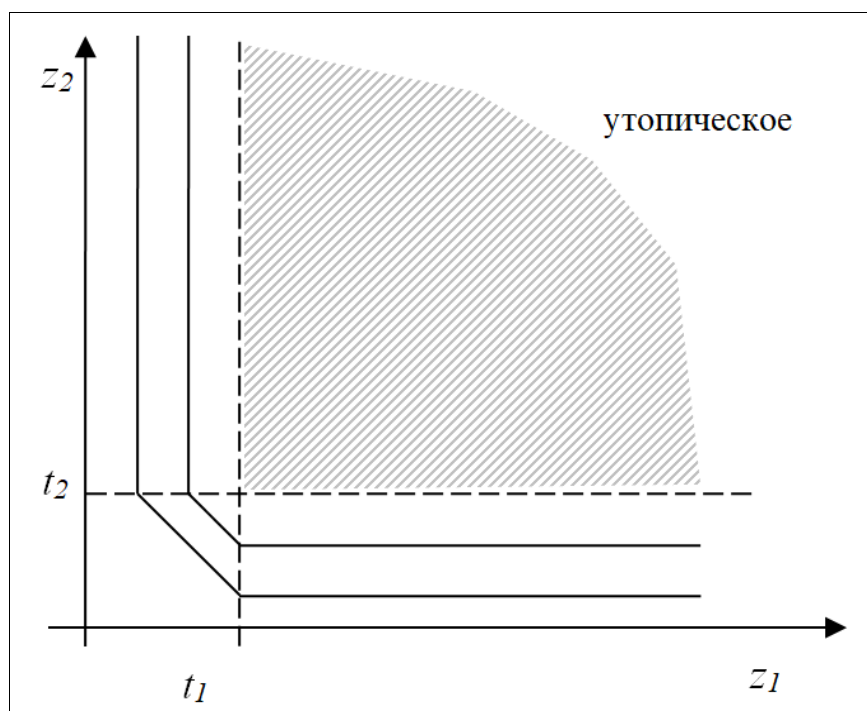


Рисунок 4.21 – ОДР, утопическое множество и линии уровня в архимедовой модели для примера 4.15

Проанализируем ещё одну задачу.

Пример 4.16

Проанализируем ещё одну задачу. Дано:

$$z_1 = x_1,$$

$$z_2 = x_2,$$

$$z_1 \leq t_1,$$

$$z_2 \geq t_2.$$

Составим архимедову модель:

$$w_1^+ d_1^+ + w_2^- d_2^- \rightarrow \min .$$

Утопическое множество в пространстве переменных отображено на рисунке 4.22. В зависимости от весов решением может быть либо x^1 , либо x^2 , либо $[x^1, x^2]$.

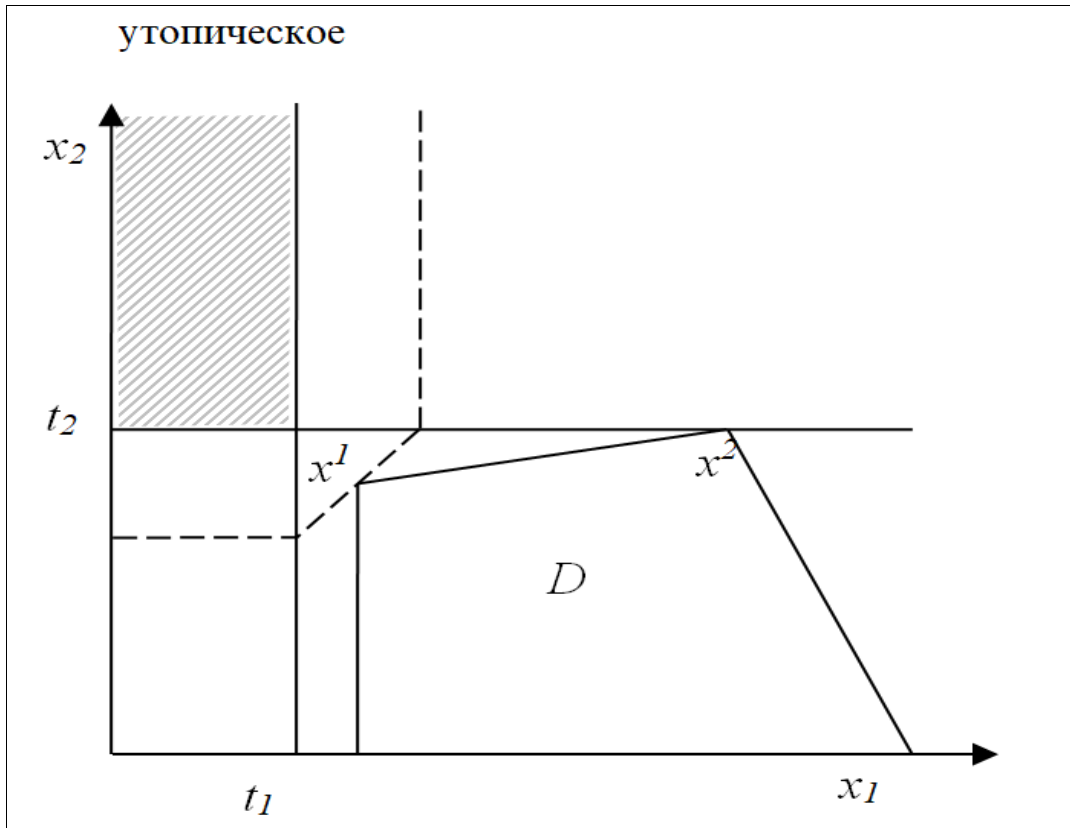


Рисунок 4.22 – ОДР, утопическое множество и линии уровня в архимедовой модели для примера 4.16

Пример 4.17

Имеется задача, геометрическая интерпретация которой приведена на рисунке 4.22.

Для критериев заданы следующие значения целей:

$$\begin{aligned} t_1^d \leq z_1 \leq t_1^4, \\ t_2^d \leq z_2 \leq t_2^4. \end{aligned}$$

Решением в этом случае является вся область четырехугольника $x^1 x^2 x^3 x^4$.

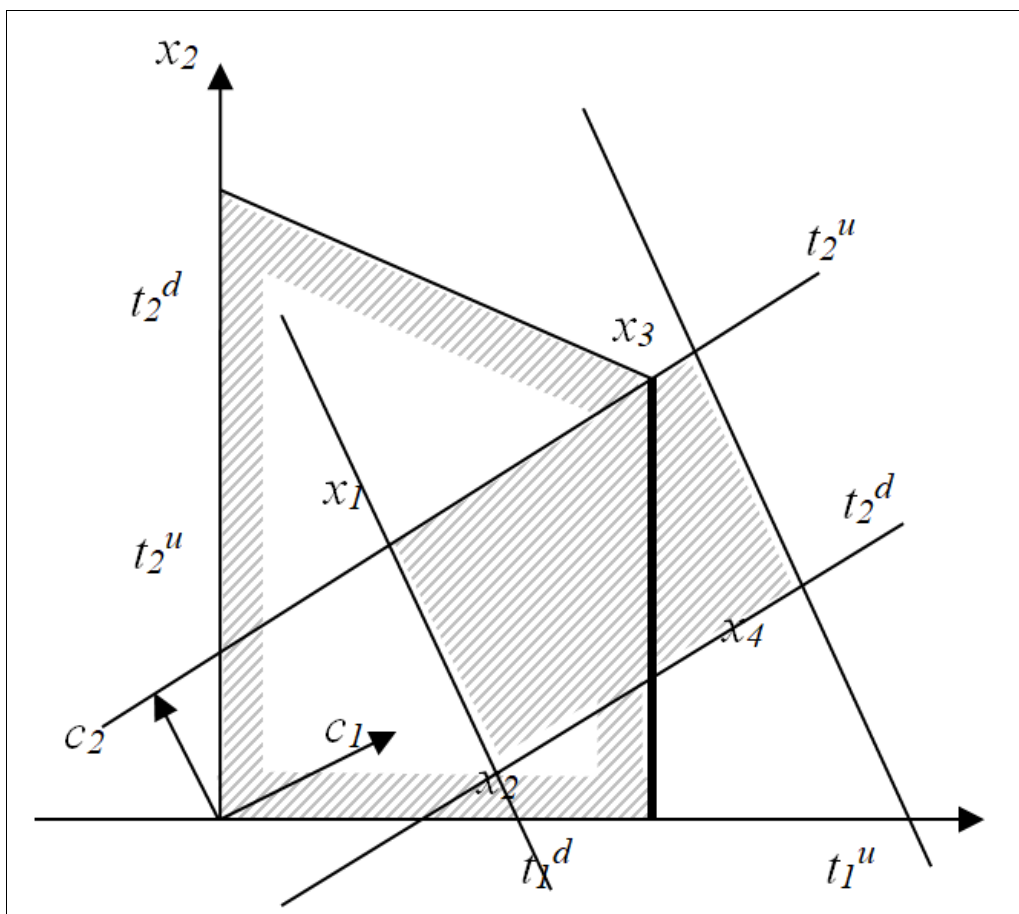


Рисунок 4.22 – Геометрическая интерпретация примера 4.16

ЗАМЕЧАНИЕ. Если такая задача будет решаться обычными методами решения задач линейного программирования, то в качестве решения может быть сгенерировано только конечное число различных точек и они могут представлять в пространстве решений следующее:

- 1) крайние точки области D , например, x_3 ;
- 2) точки границы области D , например, x_3, x_4 ;
- 3) внутренние точки области D , например, x_1, x_2 .

Если лицо принимающее решение предпочитает точку не являющейся крайней точкой допустимой области, то ее нельзя получить, не используя специальные процедуры, связанные с изменением целевых показателей t_i .

4.6.3 Задачи целевого программирования с приоритетами. Метод уступок

В задаче целевого программирования с приоритетами или лексикографическом целевом программировании, цели группируются по приоритетам. Цели с высшим первым уровнем P_1 приоритета считаются бесконечно важными по сравнению с целями со следующим (вторым уровнем) приоритета.

Цели со вторым уровнем приоритета P_2 – бесконечно важные по сравнению с третьим уровнем и т.д.

Рассмотрим этот подход на следующем примере. Пусть дана задача:

$$\begin{aligned} z_1 &= c^1 x \quad P_1; \\ z_2 &= c^2 x \quad P_2; \\ z_3 &= c^3 x \quad P_3. \end{aligned} \tag{4.34}$$

Требования:

$$\begin{aligned} z_1 &\leq t_1; \\ z_2 &\geq t_2; \\ z_3 &= t_3. \end{aligned} \tag{4.35}$$

Задача целевого программирования с приоритетами в лексикографической форме записывается в следующем виде:

$$\text{lex min } \{d_1^+, d_2^-, d_3^+ + d_3^-\} \tag{4.36}$$

$$\begin{cases} c^1 x - d_1^+ \leq t_1 \\ c^2 x + d_2^- \geq t_2 \\ c^3 x - d_3^+ + d_3^- \geq t_3 \\ x \in X, \quad d \geq 0. \end{cases} \tag{4.37}$$

Для решения этой задачи могут потребоваться три этапа оптимизации.

I Этап. Решается задача:

$$\begin{aligned}
 & d_1^+ \rightarrow \min, \\
 & \begin{cases} c^1 x - d_1^+ \leq t_1 \\ x \in X \\ d_1^+ \geq 0. \end{cases}
 \end{aligned} \tag{4.38}$$

Если в этой задаче есть альтернативные оптимумы, то формируем и решаем задачу второго этапа:

II Этап.

$$\begin{aligned}
 & d_2^- \rightarrow \min, \\
 & \begin{cases} c^1 x \leq t_1 + (d_1^+)^* \\ c^2 x + d_2^- \geq t_1 \\ x \in X \\ d_2^- \geq 0 \end{cases}
 \end{aligned} \tag{4.39}$$

где $(d_1^+)^*$ – оптимальное значение d_1^+ найденное на первом этапе.

Если в задаче есть альтернативные оптимумы, то решают задачу 3-го этапа.

III Этап.

$$\begin{aligned}
 & d_3^+ + d_3^- \rightarrow \min, \\
 & \begin{cases} c^1 x \leq t_1 + (d_1^+)^* \\ c^2 x \leq t_2 - (d_2^-)^* \\ c^3 x - d_3^+ + d_3^- = t_3 \\ x \in X \\ d_3^+, d_3^- \geq 0. \end{cases}
 \end{aligned} \tag{4.40}$$

Любое решение задачи 3-го этапа лексикографический минимум задачи с приоритетами.

Может случиться, что цели низших уровней не будут иметь шанса влиять на

решение, генерируемое в задаче целевого программирования. Это может произойти, если на каком-либо этапе мы получим единственное решение. Задачи различных этапов лексикографической оптимизации не могут решаться одновременно, т.к. для любого следующего этапа требуется информация об оптимальных решениях предыдущего этапа.

Этот недостаток преодолевается с помощью метода уступок, алгоритм которого аналогичен методу уступок решения задач многокритериальной оптимизации. Читателям предлагается сформулировать его самостоятельно.

4.6.4 ЦП эффективность и проблема чувствительности

Определим понятие ЦП эффективного решения.

Пусть \hat{d} – вектор нежелательных отклонений, соответствующий точке $\hat{x} \in D$ области D .

ОПРЕДЕЛЕНИЕ. Будем говорить, что точка \hat{x} ЦП – эффективна в смысле минимизации отклонений, если не существует другой точки $x \in D$, которой соответствует вектор отклонений d , удовлетворяющий условиям: $d \leq \hat{d}$ $d \neq \hat{d}$.

Пример 4.18

Дана задача:

$$\begin{aligned} z_1 &= c^1 x \\ z_2 &= c^2 x \end{aligned}$$

При следующих требованиях:

$$\begin{aligned} z_1 &= t_1, \\ z_2 &\geq t_2, \\ x &\in D. \end{aligned}$$

На рисунке представлена геометрическая интерпретация задачи.

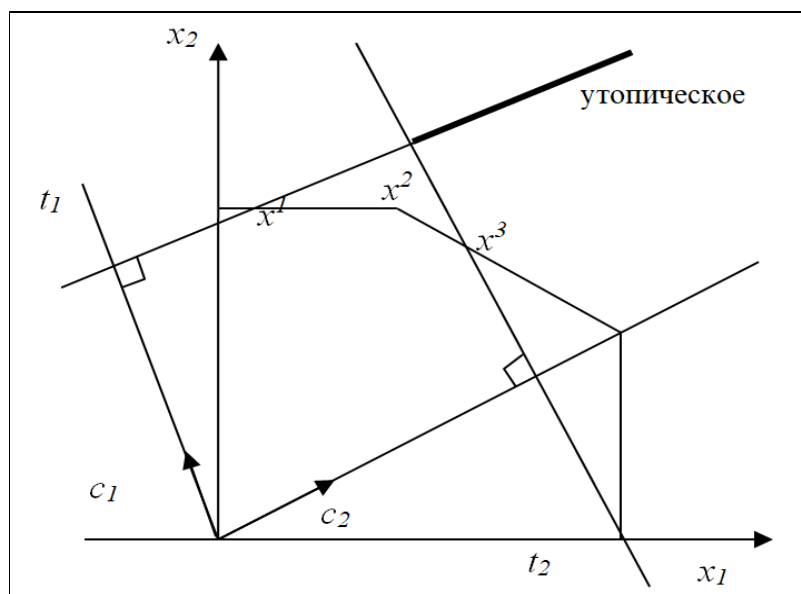


Рисунок 4.23 – Геометрическая интерпретация примера 4.18

Обозначим вектор нежелательных отклонений: $d = \begin{pmatrix} d_1^+ \\ d_1^- \\ d_2^- \end{pmatrix}$.

Для вектора нежелательных отклонений d ЦП эффективное множество представляет собой объединение отрезков $[x^1, x^2]$, $[x^2, x^3]$. Оптимальная с точки зрения лица принимающего решения точка должна принадлежать этому множеству.

Для определения ЦП эффективного множества можно использовать векторную оптимизацию, образуя минимизированные критерии из переменных нежелательных отклонений, т.е. решая задачу:

$$\begin{aligned}
 & d_1^+ \rightarrow \min, \\
 & d_1^- \rightarrow \min, \\
 & d_2^- \rightarrow \min \\
 & \begin{cases} c^1 x - d_1^+ + d_1^- = t_1 \\ c^2 x + d_2^- \geq t_2 \\ x \in X, \quad d_1^+, d_1^-, d_2^- \geq 0. \end{cases}
 \end{aligned} \tag{4.41}$$

Обычно на практике в задачах целевого программирования присутствует большое число переменных отклонений, т. к. они могут иметь разную размерность, то возникают трудности с назначением весов.

По этой причине в архимедовом ЦП мы сначала пытаемся использовать некоторые разумно назначенные штрафные веса, предполагая в дальнейшем провести анализ чувствительности с использованием других наборов весов и посмотреть нет ли лучших решений. Иногда такой подход работает удовлетворительно, иногда нет. В случае когда не работает, может произойти следующее.

Пусть набор весов приводит нас к неудовлетворительному решению, тогда некоторые веса изменяют и задачу решают снова.

Иногда новое решение будет тем же что и прежде, т.к. два набора весов соответствуют одной и той же вершине расширенного множества допустимых решений в задаче целевого программирования. Иногда, даже если мы чувствуем, что изменили веса разумным способом, получаем худшее решение.

Иногда, несмотря на то, что очень слабо изменили веса, получается совершенно другое решение, потому что мы смогли перепрыгнуть вершину, находящейся с противоположной стороны большой ЦП-эффективной грани. В результате, часто пользователь отказывается от архимедова подхода, т.к. не имеет возможности управлять изменением решения с помощью варьирования весов. В задаче целевого программирования с приоритетами, пользователь часто исследует чувствительность путем ротации приоритета. Если имеется r уровней приоритета, то имеется $r!$ вариантов их ротации. Обычно пользователь выбирает небольшое число вариантов приоритета и решает задачу для каждого из них. В результате получается несколько сильно отличающихся ЦП эффективных точек. Так же можно провести масштабирование целевых ограничений. В этом случае отклонение от цели измеряют в процентах.

Пример 4.19

Имеются ограничения вида:

$$c^i x + d_i^- \geq t_i,$$

$$c^i x = 20000,$$

$$t_i = 25000,$$

$$d_i^- = 5000.$$

Величина d_i^- составляет 20% уровня цели. Если умножим d_i^- на коэффициент $\frac{t_i}{100}$, то получим $d_i^- = 20$.

$$c^i x + \frac{t_i d_i^-}{100} \geq t_i, \quad 20000 + 250 \cdot 20 \geq 25000.$$

4.7 Вопросы для самопроверки

- 1) Приведите примеры задач принятия решений, в которых критерии конкурируют друг с другом.
- 2) Когда альтернатива является доминирующей по отношению к другой альтернативе?
- 3) Когда альтернатива является доминируемой по отношению к другой альтернативе?
- 4) Какое множество альтернатив называется множеством Парето?
- 5) В чем заключается принцип Парето?
- 6) Когда целесообразно применять метод субоптимизации?
- 7) Какие виды сверток критериев Вам известны? Укажите их преимущества и недостатки.
- 8) В чем заключается метод идеальной точки?
- 9) Чем отличается метод последовательных уступок от метода лексикографической оптимизации?
- 10) Из каких основных этапов состоит метод анализа иерархий?

- 11) Какие достоинства и недостатки метода анализа иерархий вам известны?
- 12) Какие виды задания целей Вы знаете?
- 13) В чем состоит идея архимедовой модели?
- 14) Какие виды метрик могут использоваться в целевом программировании?
- 15) Какие сложности возникают при решении задач целевого программирования?

4.8 Задания для практических (лабораторных) занятий

Задание 1 Организация выпускает 3 вида продукции видов P_1, P_2, P_3 . В производстве используется 4 вида основного сырья A_1, A_2, A_3, A_4 . В таблице даны: запасы сырья, затраты каждого вида сырья (усл.ед) на производство 1 усл.ед. каждого вида продукции, цены и себестоимость 1 усл.ед. каждого вида продукции. По одному из видов продукции были проведены маркетинговые исследования на предмет определения спроса на данный вид продукции (минимальный объем выпуска). Исходные данные представлены в таблице 4.4.

Найти оптимальные объемы выпуска каждого вида продукции, что бы прибыль и объем выпуска продукции были бы максимальны, а себестоимость. Необходимо:

- 1) составить экономико-математическую модель задачи;
- 2) найти решение по каждому из критериев в отдельности;
- 3) найти оптимальное решение по методу субоптимизации (основной критерий выбрать самостоятельно);
- 4) найти оптимальное решение с использованием аддитивной свертки критериев;
- 5) найти оптимальное решение методом уступок. Веса критериев, верхние границы и величину уступок задать самостоятельно.

Таблица 4.4 – Исходные данные

| Номер варианта | Вид сырья | Вид продукции Нормы расхода сырья | | | | Запасы сырья |
|----------------|---------------------|--------------------------------------|----------------|----------------|----------------|--------------|
| | | P ₁ | P ₂ | P ₃ | P ₄ | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | A ₁ | 3 | 1 | 2 | 10 | 25 |
| | A ₂ | 4 | 3 | 8 | 6 | 120 |
| | A ₃ | 2 | 3 | 7 | 9 | 155 |
| | A ₄ | 8 | 5 | 12 | 11 | 250 |
| | A ₅ | 2 | 3 | 4 | 1 | 100 |
| | Цена продукции | 49 | 33 | 76 | 109 | – |
| | Себестоимость прод. | 40 | 28 | 69 | 95 | – |
| | Объём выпуска прод. | Без огр. | 17 | Бз огр. | Без огр. | – |
| 2 | A ₁ | 2 | 1 | 1 | 9 | 105 |
| | A ₂ | 3 | 5 | 7 | 7 | 115 |
| | A ₃ | 1 | 2 | 6 | 8 | 100 |
| | A ₄ | 7 | 1 | 11 | 10 | 205 |
| | A ₅ | 1 | 2 | 3 | 0 | 195 |
| | Цена продукции | 44 | 20 | 63 | 92 | – |
| | Себестоимость прод. | 36 | 15 | 55 | 84 | – |
| | Объём выпуска прод. | Без огр | Без огр. | 20 | Без огр. | – |
| 3 | A ₁ | 4 | 3 | 3 | 11 | 302 |
| | A ₂ | 5 | 7 | 9 | 9 | 273 |
| | A ₃ | 3 | 5 | 8 | 10 | 501 |
| | A ₄ | 9 | 3 | 13 | 12 | 200 |
| | A ₅ | 3 | 5 | 5 | 2 | 255 |
| | Цена продукции | 62 | 57 | 76 | 118 | – |
| | Себестоимость прод. | 56 | 50 | 69 | 102 | – |
| | Объём выпуска прод. | Без огр | Без огр. | Без огр. | 10 | – |
| 4 | A ₁ | 8 | 6 | 6 | 22 | 605 |
| | A ₂ | 10 | 14 | 18 | 18 | 452 |
| | A ₃ | 5 | 10 | 16 | 20 | 953 |
| | A ₄ | 18 | 6 | 26 | 24 | 403 |
| | A ₅ | 6 | 10 | 10 | 4 | 500 |
| | Цена продукции | 124 | 114 | 152 | 236 | – |
| | Себестоимость прод. | 107 | 100 | 140 | 220 | – |
| | Объём выпуска прод. | 9 | Без огр | Без огр | Без огр. | – |
| 5 | A ₁ | 4 | 2 | 2 | 18 | 200 |
| | A ₂ | 6 | 10 | 14 | 14 | 252 |
| | A ₃ | 2 | 4 | 12 | 16 | 230 |
| | A ₄ | 14 | 2 | 22 | 20 | 415 |
| | A ₅ | 1 | 4 | 6 | 0 | 393 |
| | Цена продукции | 88 | 40 | 126 | 184 | – |
| | Себестоимость прод. | 80 | 32 | 116 | 170 | – |
| | Объём выпуска прод. | 8 | Без огр | Без огр | Без огр | – |

Продолжение таблицы 4.4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|---------------------|---------|---------|---------|---------|------|
| 6 | A ₁ | 6 | 5 | 4 | 20 | 503 |
| | A ₂ | 8 | 12 | 16 | 16 | 715 |
| | A ₃ | 4 | 6 | 14 | 18 | 400 |
| | A ₄ | 16 | 4 | 24 | 22 | 315 |
| | A ₅ | 3 | 6 | 8 | 2 | 281 |
| | Цена продукции | 101 | 100 | 162 | 264 | – |
| | Себестоимость прод. | 94 | 87 | 150 | 251 | – |
| | Объём выпуска прод. | Без огр | Без огр | 12 | Без огр | – |
| 7 | A ₁ | 7 | 6 | 5 | 21 | 1006 |
| | A ₂ | 9 | 13 | 17 | 16 | 1501 |
| | A ₃ | 5 | 7 | 15 | 19 | 833 |
| | A ₄ | 17 | 5 | 24 | 23 | 730 |
| | A ₅ | 4 | 7 | 9 | 2 | 250 |
| | Цена продукции | 124 | 125 | 195 | 274 | – |
| | Себестоимость прод. | 119 | 180 | 179 | 250 | – |
| | Объём выпуска прод. | 10 | Без огр | Без огр | Без огр | – |
| 8 | A ₁ | 7 | 6 | 5 | 21 | 500 |
| | A ₂ | 9 | 13 | 17 | 16 | 1402 |
| | A ₃ | 5 | 7 | 15 | 19 | 203 |
| | A ₄ | 17 | 5 | 24 | 23 | 600 |
| | A ₅ | 4 | 7 | 9 | 2 | 150 |
| | Цена продукции | 124 | 125 | 195 | 274 | – |
| | Себестоимость прод. | 115 | 114 | 180 | 261 | – |
| | Объём выпуска прод. | Без огр | 9 | Без огр | Без огр | – |
| 9 | A ₁ | 6 | 5 | 4 | 20 | 1010 |
| | A ₂ | 8 | 12 | 16 | 16 | 1553 |
| | A ₃ | 4 | 4 | 14 | 18 | 822 |
| | A ₄ | 16 | 6 | 24 | 22 | 735 |
| | A ₅ | 3 | 6 | 8 | 2 | 230 |
| | Цена продукции | 101 | 100 | 162 | 264 | – |
| | Себестоимость прод. | 88 | 76 | 142 | 234 | – |
| | Объём выпуска прод. | Без огр | 12 | Без огр | Без огр | – |
| 10 | A ₁ | 4 | 2 | 2 | 18 | 502 |
| | A ₂ | 6 | 10 | 14 | 14 | 718 |
| | A ₃ | 2 | 4 | 12 | 16 | 395 |
| | A ₄ | 14 | 2 | 22 | 20 | 320 |
| | A ₅ | 1 | 4 | 6 | 22 | 275 |
| | Цена продукции | 88 | 40 | 126 | 236 | – |
| | Себестоимость прод. | 79 | 35 | 118 | 223 | – |
| | Объём выпуска прод. | 20 | Без огр | Без огр | Без огр | – |

Продолжение таблицы 4.4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|---------------------|---------|---------|---------|---------|------|
| 11 | A ₁ | 1 | 10 | 7 | 4 | 1103 |
| | A ₂ | 3 | 5 | 6 | 8 | 830 |
| | A ₃ | 12 | 2 | 13 | 7 | 731 |
| | A ₄ | 2 | 6 | 11 | 9 | 822 |
| | A ₅ | 7 | 8 | 15 | 9 | 1450 |
| | Цена продукции | 64 | 92 | 135 | 86 | – |
| | Себестоимость прод | 56 | 81 | 123 | 72 | – |
| | Объём выпуска прод. | Без огр | Без огр | Без огр | 11 | – |
| 12 | A ₁ | 2 | 7 | 8 | 14 | 931 |
| | A ₂ | 15 | 9 | 6 | 1 | 725 |
| | A ₃ | 4 | 5 | 11 | 21 | 841 |
| | A ₄ | 8 | 3 | 13 | 12 | 1052 |
| | A ₅ | 9 | 7 | 5 | 1 | 635 |
| | Цена продукции | 107 | 99 | 128 | 162 | – |
| | Себестоимость прод | 96 | 87 | 119 | 153 | – |
| | Объём выпуска прод. | Без огр | Без огр | Без огр | 15 | – |
| 13 | A ₁ | 8 | 12 | 14 | 3 | 1023 |
| | A ₂ | 7 | 1 | 7 | 6 | 981 |
| | A ₃ | 3 | 4 | 8 | 10 | 725 |
| | A ₄ | 5 | 6 | 11 | 21 | 841 |
| | A ₅ | 4 | 4 | 8 | 1 | 900 |
| | Цена продукции | 91 | 89 | 152 | 106 | – |
| | Себестоимость прод | 80 | 74 | 138 | 87 | – |
| | Объём выпуска прод. | Без огр | Без огр | 12 | Без огр | – |
| 14 | A ₁ | 4 | 2 | 13 | 11 | 2011 |
| | A ₂ | 5 | 6 | 4 | 4 | 980 |
| | A ₃ | 7 | 14 | 7 | 13 | 1503 |
| | A ₄ | 8 | 21 | 12 | 11 | 2100 |
| | A ₅ | 1 | 5 | 10 | 8 | 800 |
| | Цена продукции | 78 | 125 | 136 | 140 | – |
| | Себестоимость прод | 67 | 113 | 124 | 131 | – |
| | Объём выпуска прод. | 17 | Без огр | Без огр | Без огр | – |
| 15 | A ₁ | 11 | 10 | 15 | 14 | 2631 |
| | A ₂ | 7 | 8 | 3 | 2 | 820 |
| | A ₃ | 4 | 5 | 12 | 10 | 1500 |
| | A ₄ | 3 | 4 | 8 | 9 | 953 |
| | A ₅ | 14 | 1 | 3 | 6 | 700 |
| | Цена продукции | 115 | 96 | 142 | 132 | – |
| | Себестоимость прод | 108 | 90 | 132 | 121 | – |
| | Объём выпуска прод. | Без огр | Без огр | 14 | Без огр | – |

Продолжение таблицы 4.4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|---------------------|---------|---------|---------|---------|------|
| 16 | A ₁ | 22 | 20 | 30 | 28 | 3200 |
| | A ₂ | 14 | 16 | 6 | 4 | 1500 |
| | A ₃ | 4 | 5 | 12 | 10 | 1532 |
| | A ₄ | 6 | 8 | 16 | 19 | 953 |
| | A ₅ | 14 | 1 | 3 | 6 | 700 |
| | Цена продукции | 202 | 196 | 145 | 230 | – |
| | Себестоимость прод | 193 | 179 | 132 | 118 | – |
| | Объём выпуска прод. | Без огр | 9 | Без огр | Без огр | – |
| 17 | A ₁ | 11 | 10 | 15 | 14 | 2353 |
| | A ₂ | 14 | 16 | 6 | 4 | 1700 |
| | A ₃ | 8 | 10 | 24 | 20 | 1531 |
| | A ₄ | 6 | 4 | 8 | 9 | 952 |
| | A ₅ | 28 | 2 | 6 | 12 | 708 |
| | Цена продукции | 175 | 154 | 193 | 176 | – |
| | Себестоимость прод | 161 | 144 | 182 | 161 | – |
| | Объём выпуска прод. | 13 | Без огр | Без огр | Без огр | – |
| 18 | A ₁ | 2 | 7 | 8 | 14 | 935 |
| | A ₂ | 30 | 18 | 12 | 2 | 1523 |
| | A ₃ | 4 | 5 | 11 | 21 | 844 |
| | A ₄ | 16 | 6 | 26 | 24 | 2131 |
| | A ₅ | 9 | 7 | 5 | 1 | 633 |
| | Цена продукции | 163 | 141 | 182 | 190 | – |
| | Себестоимость прод | 151 | 134 | 169 | 178 | – |
| | Объём выпуска прод. | Без огр | Без огр | Без огр | 12 | – |
| 19 | A ₁ | 4 | 14 | 16 | 28 | 1731 |
| | A ₂ | 15 | 9 | 6 | 1 | 725 |
| | A ₃ | 8 | 5 | 22 | 42 | 1603 |
| | A ₄ | 8 | 3 | 13 | 1 | 1055 |
| | A ₅ | 18 | 14 | 10 | 2 | 1270 |
| | Цена продукции | 138 | 135 | 206 | 190 | – |
| | Себестоимость прод | 123 | 127 | 187 | 176 | – |
| | Объём выпуска прод. | Без огр | 9 | Без огр | Без огр | – |
| 20 | A ₁ | 2 | 20 | 14 | 8 | 2232 |
| | A ₂ | 3 | 5 | 6 | 8 | 830 |
| | A ₃ | 4 | 4 | 26 | 14 | 1505 |
| | A ₄ | 2 | 6 | 11 | 2 | 831 |
| | A ₅ | 14 | 16 | 30 | 18 | 2533 |
| | Цена продукции | 112 | 160 | 224 | 132 | – |
| | Себестоимость прод | 102 | 143 | 209 | 118 | – |
| | Объём выпуска прод. | Без огр | 13 | Без огр | Без огр | – |

Продолжение таблицы 4.4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|---------------------|---------|---------|---------|---------|------|
| 21 | A ₁ | 1 | 10 | 7 | 4 | 1000 |
| | A ₂ | 6 | 10 | 12 | 16 | 1830 |
| | A ₃ | 12 | 2 | 13 | 7 | 751 |
| | A ₄ | 4 | 12 | 22 | 4 | 1743 |
| | A ₅ | 7 | 8 | 15 | 9 | 1452 |
| | Цена продукции | 80 | 128 | 181 | 122 | – |
| | Себестоимость прод | 67 | 109 | 168 | 112 | – |
| | Объём выпуска прод. | 15 | Без огр | Без огр | Без огр | – |
| 22 | A ₁ | 4 | 2 | 2 | 18 | 505 |
| | A ₂ | 12 | 20 | 28 | 28 | 1500 |
| | A ₃ | 2 | 4 | 12 | 16 | 394 |
| | A ₄ | 28 | 4 | 44 | 40 | 653 |
| | A ₅ | 1 | 4 | 6 | 22 | 840 |
| | Цена продукции | 103 | 114 | 240 | 344 | – |
| | Себестоимость прод | 93 | 101 | 222 | 321 | – |
| | Объём выпуска прод. | Без огр | Без огр | Без огр | 10 | – |
| 23 | A ₁ | 8 | 4 | 4 | 36 | 1000 |
| | A ₂ | 6 | 10 | 14 | 14 | 720 |
| | A ₃ | 4 | 8 | 24 | 32 | 810 |
| | A ₄ | 14 | 2 | 22 | 20 | 323 |
| | A ₅ | 2 | 8 | 12 | 22 | 551 |
| | Цена продукции | 106 | 96 | 184 | 394 | – |
| | Себестоимость прод | 89 | 77 | 157 | 381 | – |
| | Объём выпуска прод. | 12 | Без огр | Без огр | Без огр | – |
| 24 | A ₁ | 6 | 5 | 4 | 20 | 1103 |
| | A ₂ | 16 | 24 | 32 | 32 | 3210 |
| | A ₃ | 4 | 4 | 14 | 18 | 831 |
| | A ₄ | 32 | 12 | 48 | 44 | 1535 |
| | A ₅ | 3 | 6 | 8 | 2 | 250 |
| | Цена продукции | 165 | 163 | 296 | 376 | – |
| | Себестоимость прод | 152 | 147 | 276 | 354 | – |
| | Объём выпуска прод. | Без огр | 14 | Без огр | Без огр | – |
| 25 | A ₁ | 12 | 10 | 8 | 40 | 2225 |
| | A ₂ | 8 | 2 | 16 | 16 | 1631 |
| | A ₃ | 4 | 4 | 14 | 18 | 1733 |
| | A ₄ | 32 | 12 | 48 | 44 | 1520 |
| | A ₅ | 6 | 12 | 16 | 4 | 450 |
| | Цена продукции | 174 | 106 | 238 | 410 | – |
| | Себестоимость прод | 152 | 92 | 222 | 395 | – |
| | Объём выпуска прод. | Без огр | Без огр | 11 | Без огр | – |

Продолжение таблицы 4.4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|----|---------------------|---------|---------|---------|---------|------|
| 26 | A ₁ | 19 | 10 | 8 | 41 | 2224 |
| | A ₂ | 7 | 3 | 16 | 16 | 1630 |
| | A ₃ | 4 | 4 | 15 | 20 | 1778 |
| | A ₄ | 30 | 14 | 48 | 45 | 1532 |
| | A ₅ | 6 | 121 | 17 | 4 | 350 |
| | Цена продукции | 170 | 126 | 251 | 412 | – |
| | Себестоимость прод | 156 | 112 | 243 | 391 | – |
| | Объём выпуска прод. | Без огр | Без огр | 15 | Без огр | – |
| 27 | A ₁ | 29 | 10 | 8 | 41 | 2384 |
| | A ₂ | 8 | 6 | 18 | 17 | 630 |
| | A ₃ | 15 | 8 | 15 | 30 | 1788 |
| | A ₄ | 36 | 14 | 48 | 43 | 1632 |
| | A ₅ | 7 | 12 | 21 | 5 | 351 |
| | Цена продукции | 260 | 456 | 351 | 722 | – |
| | Себестоимость прод | 246 | 432 | 333 | 689 | – |
| | Объём выпуска прод. | Без огр | Без огр | Без огр | 9 | – |
| 28 | A ₁ | 32 | 8 | 8 | 41 | 2304 |
| | A ₂ | 9 | 6 | 18 | 17 | 630 |
| | A ₃ | 19 | 15 | 16 | 30 | 1588 |
| | A ₄ | 36 | 15 | 48 | 43 | 1642 |
| | A ₅ | 7 | 15 | 21 | 5 | 349 |
| | Цена продукции | 560 | 156 | 357 | 822 | – |
| | Себестоимость прод | 542 | 140 | 335 | 786 | – |
| | Объём выпуска прод. | Без огр | 15 | Без огр | Без огр | – |
| 29 | A ₁ | 32 | 8 | 8 | 41 | 3304 |
| | A ₂ | 9 | 11 | 18 | 17 | 6322 |
| | A ₃ | 21 | 15 | 16 | 24 | 458 |
| | A ₄ | 36 | 15 | 13 | 43 | 1646 |
| | A ₅ | 5 | 15 | 21 | 17 | 365 |
| | Цена продукции | 160 | 156 | 357 | 822 | – |
| | Себестоимость прод | 147 | 141 | 337 | 788 | – |
| | Объём выпуска прод. | Без огр | Без огр | 17 | Без огр | – |
| 30 | A ₁ | 22 | 9 | 11 | 42 | 3354 |
| | A ₂ | 29 | 11 | 19 | 17 | 8322 |
| | A ₃ | 32 | 18 | 16 | 25 | 4158 |
| | A ₄ | 31 | 15 | 13 | 43 | 1641 |
| | A ₅ | 7 | 8 | 22 | 56 | 366 |
| | Цена продукции | 120 | 256 | 359 | 922 | – |
| | Себестоимость прод | 112 | 243 | 329 | 877 | – |
| | Объём выпуска прод. | Без огр | 15 | Без огр | Без огр | – |

Задача 2 Для каждой из данных задач многокритериальной оптимизации найдите множество решений, оптимальных по Парето, определите решения, получаемые с помощью субоптимизации, лексикографической оптимизации, метода свертки критериев, метода идеальной точки и метода последовательных уступок (необходимые параметры методов выбрать, исходя из здравого смысла):

$$\begin{array}{ll}
 f_1(x) = x_2 \rightarrow \max, & f_1(x) = x_1 \rightarrow \max, \\
 f_2(x) = -x_1 - 2x_2 \rightarrow \max, & f_2(x) = 2x_1 - x_2 \rightarrow \max, \\
 f_3(x) = x_1 + 3x_2 \rightarrow \max, & f_3(x) = x_2 \rightarrow \max, \\
 \text{а) } \begin{cases} x_1 + x_2 \leq 6 \\ x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0. \end{cases} & \text{б) } \begin{cases} -x_1 + 3x_2 \leq 2 \\ 2x_1 - x_2 \leq 5 \\ 1 \leq x_1 \leq 4 \\ 1 \leq x_2 \leq 7. \end{cases}
 \end{array}$$

$$\begin{array}{ll}
 f_1(x) = x_1 \rightarrow \max, & f_1(x) = 3x_1 + 4x_2 \rightarrow \max, \\
 f_2(x) = x_1 + 2x_2 \rightarrow \max, & f_2(x) = -x_2 \rightarrow \max, \\
 f_3(x) = x_1 - 4x_2 \rightarrow \max, & f_3(x) = x_1 + 2x_2 \rightarrow \max, \\
 \text{в) } \begin{cases} x_1 + x_2 \leq 6 \\ x_2 \leq 3 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases} & \text{г) } \begin{cases} -x_1 + 3x_2 \leq 2 \\ 2x_1 - x_2 \leq 5 \\ 1 \leq x_1 \leq 4 \\ 0 \leq x_2 \leq 5. \end{cases}
 \end{array}$$

Задача 3 Решить задачу при заданных ограничениях, используя «архимедову модель» целевого программирования.

$$\begin{array}{l}
 c11 \cdot x_1 + c12 \cdot x_2 \in [a1, b1] \\
 c21 \cdot x_1 + c22 \cdot x_2 \in [a2, b2] \\
 \begin{cases} d11 \cdot x_1 + d12 \cdot x_2 \leq d13 \\ d21 \cdot x_1 + d22 \cdot x_2 \leq d23 \end{cases} \\
 x_1, x_2 \geq 0
 \end{array}$$

Для этого:

- 1) изобразить графически множество допустимых решений;
- 2) изобразить множество допустимых решений и «утопическое множество» в 8-пространстве критериев;
- 3) построить «архимедову модель», веса задать самостоятельно;
- 4) найти аналитическое решение, используя программу «DVSimp» и надстройку MS Excel «Поиск решения».

Таблица 4.5 – Варианты заданий

| № варианта | c11 | c12 | [a1, b1] | c21 | c22 | [a1, b1] | d11 | d12 | d13 | d21 | d22 | d23 |
|------------|-----|-----|----------|-----|-----|-----------|-----|-----|-----|-----|-----|-----|
| 1 | 1 | 2 | ≥ 7 | -2 | 4 | [12, 15] | 3 | 3 | 9 | -2 | 5 | 10 |
| 2 | 2 | 1 | [6, 8] | -2 | 4 | ≥ 14 | 3 | 3 | 9 | -2 | 5 | 10 |
| 3 | 2 | 1 | ≥ 6 | 4 | -2 | [11, 12] | 4 | 3 | 10 | -5 | 5 | 3 |
| 4 | 2 | 1 | ≥ 8 | 4 | -2 | ≤ 4 | 4 | 3 | 10 | -5 | 2 | 10 |
| 5 | 3 | 1 | [7, 8] | 3 | -2 | ≥ 8 | 5 | 3 | 11 | -5 | 3 | 5 |
| 6 | 2 | 2 | [5, 6] | 4 | -3 | ≥ 8 | 4 | 5 | 8 | 5 | -2 | 6 |
| 7 | 3 | -1 | [7, 9] | -3 | 2 | ≥ 8 | 5 | 3 | 11 | -5 | 1 | 5 |
| 8 | -2 | 2 | ≥ 4 | 4 | -3 | [4, 6] | -4 | 5 | 8 | 5 | -2 | 6 |
| 9 | -3 | 1 | ≥ 4 | 3 | 2 | [8, 12] | 5 | 3 | 11 | -5 | 3 | 8 |
| 10 | 2 | 2 | ≥ 8 | 4 | 3 | [9, 14] | 4 | 2 | 8 | -5 | 3 | 8 |
| 11 | 1 | 2 | [10, 12] | -2 | 4 | ≥ 7 | 3 | 3 | 9 | -2 | 5 | 10 |
| 12 | 1 | 2 | ≥ 7 | -2 | 4 | [11, 14] | 3 | 3 | 9 | -2 | 5 | 10 |
| 13 | 2 | 1 | [8, 10] | 4 | -2 | ≥ 6 | 4 | 3 | 10 | -5 | 5 | 3 |
| 14 | 2 | 1 | [10, 11] | 4 | -2 | ≥ 6 | 4 | 3 | 10 | -5 | 2 | 10 |
| 15 | 3 | 1 | [11, 14] | 3 | -2 | ≥ 12 | 5 | 3 | 11 | -5 | 3 | 5 |
| 16 | 2 | 2 | [9, 10] | 4 | -3 | ≥ 5 | 4 | 5 | 8 | 5 | -2 | 6 |
| 17 | 3 | -1 | ≥ 7 | -3 | 2 | [11, 13] | 5 | -3 | 11 | -5 | 3 | 5 |
| 18 | -2 | 2 | [10, 12] | 4 | -3 | ≥ 11 | -4 | 5 | 8 | 5 | -2 | 6 |
| 19 | -3 | 1 | [7, 9] | 3 | 2 | ≥ 7 | 5 | 3 | 11 | -5 | 3 | 8 |
| 20 | 2 | -2 | [8, 10] | 4 | 3 | ≥ 12 | 4 | 5 | 8 | 5 | -2 | 9 |

4.9 Задания для самостоятельной проработки и подготовке к контрольным работам

Задача 4.1 Предположим, что предстоит выбрать место работы из девяти

вариантов, представленных в табл.1. В качестве основных критериев взяты: зарплата Z , длительность отпуска D , время поездки на работу B . Из смысла задачи следует, что критерии Z и D следует максимизировать, а критерий B – минимизировать. Какой вариант является оптимальным?

Таблица 4.6 – Исходные данные к задаче 4.1

| Варианты | Критерий | | |
|----------|------------------|-----------------------------|----------------------|
| | Зарплата, (руб.) | Длительность отпуска, (дни) | Время поездки, (мин) |
| 1 | 900 | 20 | 60 |
| 2 | 500 | 30 | 20 |
| 3 | 700 | 36 | 40 |
| 4 | 800 | 40 | 50 |
| 5 | 400 | 60 | 15 |
| 6 | 600 | 30 | 10 |
| 7 | 900 | 35 | 60 |
| 8 | 600 | 24 | 10 |
| 9 | 650 | 35 | 40 |

Задача 4.2 Для участия в конкурсе представлено пять вариантов строительства предприятий различного типа (это могут быть машиностроительный завод, текстильная фабрика, молочный завод и т.п.) на территории, непосредственно прилегающей к жилому району. Оценивание качества проекта производится по четырем критериям: f_1 – стоимость реализации проекта, f_2 – величина прибыли проектируемого предприятия, f_3 – величина экологического ущерба от строительства и f_4 – заинтересованность жителей района в строительстве данного предприятия. Для оценки всех критериев была использована пятибалльная шкала в 1, 2, 3, 4 и 5 баллов. Определите наиболее предпочтительные варианты для строительства.

Таблица 4.7 – Оценивание качества проектов

| Варианты строительства | f_1 | f_2 | f_3 | f_4 |
|------------------------|-------|-------|-------|-------|
| A1 | 4 | 3 | 4 | 3 |
| A2 | 5 | 3 | 3 | 3 |
| A3 | 2 | 4 | 2 | 4 |
| A4 | 5 | 3 | 2 | 3 |
| A5 | 3 | 4 | 3 | 4 |

Задача 4.3 Имеется три объекта для инвестирования средств. Для оценки выгодности инвестирования используются два критерия: величина прироста прибыли от вложения, измеряемая в процентах по отношению к исходной сумме инвестирования, и надежность вложенных средств, измеряемая в пятибалльной шкале от 1 до 5. Определите наиболее предпочтительные объекты для инвестирования средств.

Таблица 4.8 – Оценивание выгодности объектов инвестирования

| Объекты инвестирования | Величина прироста прибыли от вложения | Надежность вложенных средств |
|------------------------|---------------------------------------|------------------------------|
| A1 | 40 | 1 |
| A2 | 30 | 2 |
| A3 | 10 | 3 |

Задача 4.4 Из железного листа, имеющего форму квадрата со стороной 1, требуется скроить коробку максимально возможного объема при минимальном расходе материала. При вырезке из листа четырех квадратов с неизвестной стороной x и сгибании по линиям, обозначенным пунктиром (рисунок 4.24), получается коробка, объем которой равен $x(1-2x)^2$. При этом теряется железо общей площадью $4x^2$. Как надо скроить коробку, чтобы затраты металла на ее производство были минимальным, а сама коробка была максимального объема?

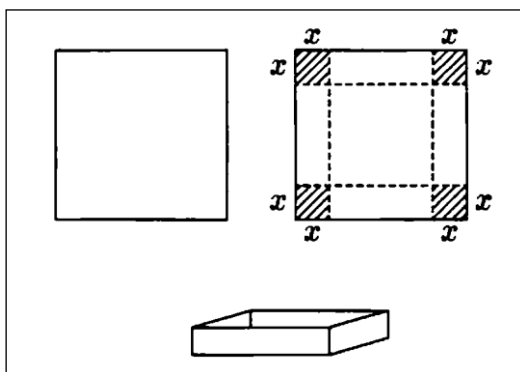


Рисунок 4.24 – Коробка

Задача 4.5 Переносной автомат для забивания стальных дюбелей в бетонные

стены состоит из корпуса с магазином, содержащим запас дубелей; подающего-спускового механизма с зарядами и ствола. Требуется определить основные конструктивные параметры автомата – длину ствола L и число дубелей N при следующих исходных данных: число дубелей $N \geq 12$; масса одного дубеля с зарядом равна $m=50$ г; масса ствола 1.6 кг/м; масса корпуса 2 кг.

При фиксированной величине заряда и заданной массе дубеля скорость V выбрасывания связана с длиной ствола L соотношением $V = k \cdot \sqrt{L}$; где $k=150$ м^{0.5}/с. Минимально допустимая скорость $V_{\min}=100$ м/с. Масса автомата не должна превышать 6 кг. Частными критериями являются скорость выбрасывания и число дубелей, помещающихся в магазине. Выбор этих критериев объясняется тем, что чем выше V , тем надёжнее дубеля проникают в бетонные стены любой марки, а чем больше N , тем удобнее работать. По мнению экспертов оба критерия V и N имеют одинаковую важность.

Задача 4.6 На множестве $\begin{cases} 0 \leq x_1 \leq 6 \\ 0 \leq x_2 \leq 4 \\ 2x_1 - x_2 \geq 7 \end{cases}$ заданы две линейные функции:

$f_1(x) = x_1$ и $f_2(x) = x_1 - x_2 - 1$. Требуется найти решение задачи $f_1(x) \rightarrow \max$ и $f_2(x) \rightarrow \min$ с помощью указания нижних границ критериев, если имеется дополнительная информация о значениях нижней границы по каждому критерию: $\gamma_1 = 3, \gamma_2 = 3$.

Задача 4.7 Решите задачу методом субоптимизации:

$$\begin{aligned} f_1(x) &= x_2 \rightarrow \max, \\ f_2(x) &= -x_1 - 2x_2 \rightarrow \max, \\ f_3(x) &= x_1 + 3x_2 \rightarrow \max, \\ \begin{cases} x_1 + x_2 \leq 6 \\ x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0. \end{cases} \end{aligned}$$

Наиболее приоритетный критерий f_1 , для критериев f_2 и f_3 заданы нижние

границы $\gamma_2 = 4$, $\gamma_3 = 2$.

Задача 4.8 Решите задачу методом лексикографической оптимизации:

$$\begin{aligned} f_1(x) &= x_2 \rightarrow \max, \\ f_2(x) &= -x_1 - 2x_2 \rightarrow \max, \\ f_3(x) &= x_1 + 3x_2 \rightarrow \max, \\ \begin{cases} x_1 + x_2 \leq 6 \\ x_2 \leq 4 \\ x_1 \geq 0 \\ x_2 \geq 0. \end{cases} \end{aligned}$$

Критерии упорядочены согласно их нумерации. Наиболее приоритетный критерий f_1 , для критериев f_2 и f_3 заданы нижние границы $\gamma_2 = 4$, $\gamma_3 = 2$.

Задача 4.9 Решите задачу методом аддитивной свертки критериев, если известны коэффициенты важности критериев: $\alpha_1 = 0,7$, $\alpha_2 = 0,2$ и $\alpha_3 = 0,1$.

$$\begin{aligned} f_1(x) &= x_2 \rightarrow \max, \\ f_2(x) &= x_1 - x_2 \rightarrow \max, \\ f_3(x) &= x_1 + 3x_2 \rightarrow \max, \\ \begin{cases} x_1 + x_2 \leq 6 \\ x_2 \leq 4. \end{cases} \end{aligned}$$

Задача 4.10 Решите задачу методом мультипликативной свертки критериев, если известны коэффициенты важности критериев: $\alpha_1 = 0,7$, $\alpha_2 = 0,2$ и $\alpha_3 = 0,1$.

$$\begin{aligned} f_1(x) &= x_2 \rightarrow \max, \\ f_2(x) &= x_1 - x_2 \rightarrow \max, \\ f_3(x) &= x_1 + 3x_2 \rightarrow \max, \\ \begin{cases} x_1 + x_2 \leq 6 \\ x_2 \leq 4. \end{cases} \end{aligned}$$

Задача 4.11 Решите задачу методом последовательных уступок, если критерии упорядочены согласно их нумерации и величины уступок по первому и второму критериям равны $\Delta_1 = 3$ и $\Delta_2 = \frac{5}{3}$ соответственно.

$$f_1(x) = x_2 \rightarrow \max,$$

$$f_2(x) = x_1 - x_2 \rightarrow \max,$$

$$f_3(x) = x_1 + 3x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + x_2 \leq 6 \\ x_2 \leq 4. \end{cases}$$

Задача 4.12 На множестве, определяемой системой неравенств
$$\begin{cases} 0 \leq x \leq 2 \\ 0 \leq y \leq 4 \\ 2x + y \leq 6 \end{cases}$$

заданы линейные функции: $U = x + y + 2$ и $V = x - y + 6$. Требуется найти решение задачи $U \rightarrow \max$ и $V \rightarrow \max$ при условии, что точка утопии M^* имеет координаты (7; 8).

5 Модели сетевого планирования

5.1 Основные понятия и определения

Методы сетевого планирования и управления получили развитие в конце 50-х годов XX века и стали применяться при планировании сложных процессов в строительстве, научных разработках, в областях народного хозяйства и т.д. [19].

Приведем примеры задач математического программирования, представимых в виде сетевых моделей:

1) проектирование газопровода, соединяющего буровые скважины (целевая функция соответствующей модели должна минимизировать стоимость строительства газопровода);

2) поиск кратчайшего маршрута между двумя городами по существующей сети дорог;

3) определение максимальной пропускной способности трубопровода для транспортировки угольной пульпы от угольных шахт к электростанциям;

4) определение схемы транспортировки нефти от пунктов нефтедобычи к нефтеперерабатывающим заводам с минимальной стоимостью транспортировки;

5) составление временного графика строительных работ (определение дат начала и завершения отдельных этапов работ).

Решение перечисленных задач требует применения различных сетевых оптимизационных алгоритмов [36]:

1) алгоритм нахождения минимального остовного дерева;

2) алгоритм поиска кратчайшего пути;

3) алгоритм определения максимального потока;

4) алгоритм минимизации стоимости потока в сети с ограниченной пропускной способностью;

5) алгоритм определения критического пути.

ОПРЕДЕЛЕНИЕ. Сетевая модель представляет собой план выполнения

некоторого комплекса работ (операций), заданного в форме сети, графическое изображение которой называется сетевым графиком [19].

ОПРЕДЕЛЕНИЕ. Сетевой график представляет собой схему, на которой в определенном порядке показаны все операции по созданию сначала промежуточной продукции с определенной степенью готовности, а под конец – полное завершение разработки, т.е. достижение конечной цели [30].

ОПРЕДЕЛЕНИЕ. Сетевой график представляет собой сеть, состоящую из отдельных нитей и узлов, отражающих логическую взаимосвязь и взаимообусловленность всех операций, входящих в общий комплекс [30].

В основу построения сети закладываются три основных понятия: работа, событие и путь.

ОПРЕДЕЛЕНИЕ. Работа – любой трудовой процесс, сопровождающийся затратами времени и ресурсов (например, устройство фундамента под стены здания, монтаж двигателя, проектирование какого-либо узла машины, испытание установки и т.д.) [30].

В понятие «работа» также входит и ожидание, т.е. пассивный процесс, не требующий затрат труда и материальных ресурсов, но отнимающий время. К таким процессам можно отнести технологические перерывы, например, твердение бетона при строительстве. Кроме того, под работой понимают простую зависимость, т.е. логическую связь между двумя или большим числом операций. Иногда эту зависимость называют фиктивной (холостой), так как она не требует никаких затрат – ни времени, ни труда, ни средств [30].

ОПРЕДЕЛЕНИЕ. Событие представляет собой итог какой-либо деятельности, промежуточный или окончательный результат выполнения одной или нескольких предшествующих работ, позволяющий приступить к выполнению последующих работ. Событие, в отличие от работы, не является процессом. Оно не имеет длительности, так как совершается мгновенно и не сопровождается затратами времени и средств [30].

На сетевом графике события изображаются кружками с порядковым номером, действительные работы и ожидания – сплошными стрелками, а фиктивные работы –

пунктирными стрелками (рисунок 5.1).

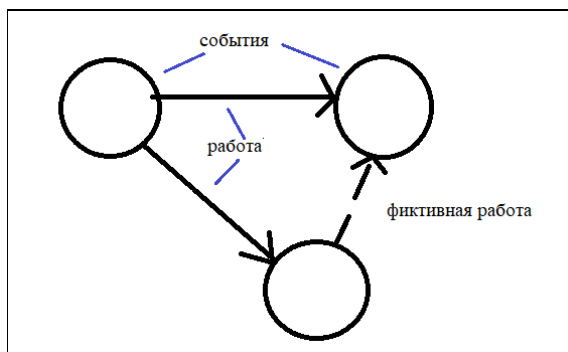


Рисунок 5.1 – Обозначения в сетевых графиках

Пример 5.1

Предприятие по производству офисной мебели имеет:

- 1) завод, на котором осуществляется производство товаров;
- 2) склад C_1 на территории завода для хранения готовой продукции (размещение произведенных товаров на складе занимает 1 день);
- 3) склад C_2 в черте города, с которого осуществляется отгрузка товаров в магазины;
- 4) склад C_3 в соседнем городе, с которого так же осуществляется отгрузка товаров в магазин.

Произведенная на заводе продукция поступает на склад C_1 и развозится по складам C_2 и C_3 (доставка занимает 3 дня и 7 дней соответственно). Продукция со склада C_2 отгружается в два магазина – M_1 и M_2 , находящихся в черте города (сроки доставки составляют 2 и 3 дня соответственно). Продукция со склада C_3 поступает в магазин M_3 (срок доставки – 1 день).

Магазины M_2 и M_3 осуществляют доставку приобретенной продукции до потребителей (в сроки 5 дней и 4 дня соответственно), а в магазине M_1 вывоз товаров осуществляется самими клиентами.

Составим сетевой график процесса перемещения товара. Выделим события сетевого графика (таблица 5.1).

Таблица 5.1 – Обозначение и описание событий сетевого графика

| Обозначение | Описание события |
|-------------|--|
| 1 | Товар произведен и готов к отгрузке на склад |
| 2 | Товар отгружен на склад C_1 |
| 3 | Товар поступил на склад C_2 |
| 4 | Товар поступил на склад C_3 |
| 5 | Товар поступил в магазин M_1 |
| 6 | Товар поступил в магазин M_2 |
| 7 | Товар поступил в магазин M_3 |
| 8 | Товар доставлен потребителю |

Таким образом, сетевой график будет включать 8 событий.

Рассмотрим список работ, связывающих между собой выделенные события (таблица 5.2).

Таблица 5.2 – Обозначение и описание работ сетевого графика

| Обозначение | Описание работы | Время выполнения |
|-------------|---|------------------|
| (1;2) | Произведенный товар поступает на склад C_1 | 1 день |
| (2;3) | Товар со склада C_1 поступает на склад C_2 | 3 дня |
| (2;4) | Товар со склада C_1 поступает на склад C_3 | 7 дней |
| (3;5) | Товар со склада C_2 поступает в магазин M_1 | 2 дня |
| (3;6) | Товар со склада C_2 поступает в магазин M_2 | 3 дня |
| (4;7) | Товар со склада C_3 поступает в магазин M_3 | 1 день |
| (5;8) | Товар отгружается клиенту | мгновенно |
| (6;8) | Магазин M_2 доставляет товар клиенту | 5 дней |
| (7;8) | Магазин M_3 доставляет товар клиенту | 4 дня |

Из таблицы 5.2 следует, что сетевой график включает 9 работ, одна из которых (работа (5;8)) является фиктивной. Поскольку клиенты самостоятельно осуществляют вывоз товара из магазина после его приобретения, эта работа не требует временных, трудовых и материальных затрат самого предприятия.

На рисунке 5.2 представим сетевой график выполнения работ по реализации продукции предприятия.

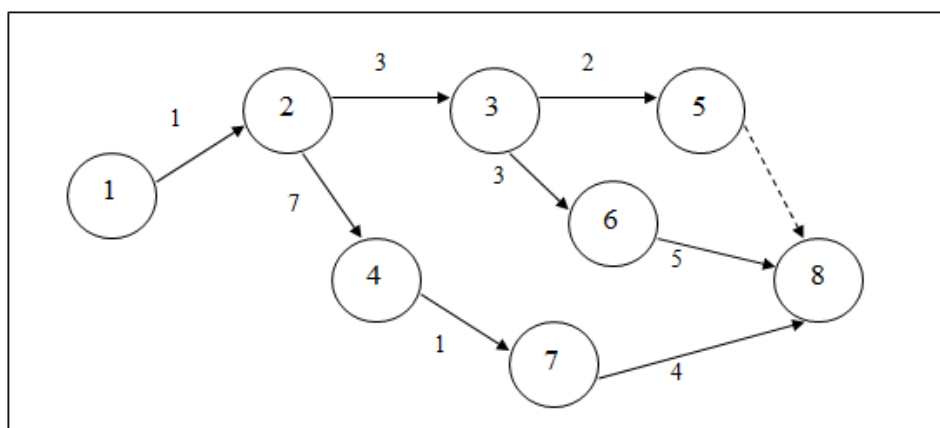


Рисунок 5.2 – Сетевой график выполнения работ

Стрелки вычерчиваются без масштаба, их длина произвольна, но на чертеже они располагаются в таком порядке, который указывает на определенную последовательность выполнения операций [30].

Любая работа-стрелка соединяет только два события и отражает процесс перехода от одного события к другому. Событие, из которого стрелка выходит, называется начальным или предшествующим по отношению к данной работе. Событие, в которое стрелка входит, является конечным или последующим. Одно и тоже событие, кроме начального и конечного, одновременно является и предшествующим, и последующим [30].

Любую работу можно обозначить номерами двух событий (рисунок 5.2). Переход от одной работы к другой возможен лишь после получения какого-то результата, т.е. свершения события, стоящего у начала работы [30].

Все события в свою очередь связываются между собой в соответствии с последовательностью выполнения работ. В одно и то же событие может входить несколько работ, так же как и выходить из него. Таким образом, свершение события может зависеть от одной, двух или более работ, в него входящих. Любое событие, фиксирующее момент окончания одной или нескольких работ, одновременно является началом другой работы или группы работ. С этой точки зрения все события

разделяются на простые и сложные (узловые). Первые имеют только одну входящую и одну выходящую работы. Вторые могут иметь несколько входящих и выходящих работ. Первое событие, которому не предшествуют никакие события, называется исходным, оно определяет начало выполнения программы. Последнее событие, обуславливающее достижение конечной цели (например, ввод здания в эксплуатацию), называется завершающим, оно не имеет последующих работ. Все остальные события считаются промежуточными [30].

Все работы, входящие в сетевой график, имеют временные оценки. Событие считается свершившимся только тогда, когда будет закончена самая длительная из всех входящих в него работ. Продолжительность работ подписывают над стрелками или под ними, а наименование работ записывают или в специальной таблице в соответствии с принятым кодом, или проставляют непосредственно на самом графике над стрелкой [30].

ОПРЕДЕЛЕНИЕ. Путь представляет собой непрерывную технологическую последовательность работ (цепь) от исходного события до завершающего.

Длина пути определяется суммой продолжительности лежащих на нем работ.

ОПРЕДЕЛЕНИЕ. Путь, на котором суммарная продолжительность работ имеет максимальное значение, называется критическим путем. Он определяет время, необходимое для выполнения всех работ, включенных в сетевой график.

Работы, лежащие на критическом пути, являются критическими и от их продолжительности зависит конечный срок выполнения программы.

В сетях возможно существование нескольких критических путей. Пути, имеющие продолжительность меньше критической, но близкие к ней, называются подкритическими или субкритическими. Все остальные пути принято называть ненапряженными или некритическими [30].

Наличие критического пути позволяет использовать его в качестве основы для оптимизации плана. Например, чтобы сократить срок создания нового объекта, необходимо принять меры, позволяющие уменьшить продолжительность выполнения не всех работ, а только работ, находящихся на критическом пути [36].

Все работы, не лежащие на критическом пути, обладают некоторыми

резервами времени, которые являются важными показателями работы сети. Под резервами времени понимаются допустимые сдвиги сроков свершения событий и выполнения работ, не меняющие срока завершающего события. Наличие запаса времени имеет большое практическое значение, поскольку позволяет руководителям свободно маневрировать внутренними ресурсами. В процессе выполнения программы в пределах имеющегося запаса времени можно увеличивать продолжительность некритических работ и передавать освободившиеся людские ресурсы и технические средства на работы, находящиеся на критическом пути, от которых зависит выполнение плана [30].

Отметим, что сокращение сроков выполнения критических работ может привести к возникновению нового критического пути.

5.2 Правила построения сетевого графика

Процесс построения сетевого графика включает следующие этапы [19]:

- 1) выделение совокупности работ;
- 2) разработка перечня работ и событий;
- 3) анализ логических связей и формирование последовательности выполнения работ;
- 4) закрепление работ за исполнителями.

При построении сетевого графика необходимо соблюдать ряд правил:

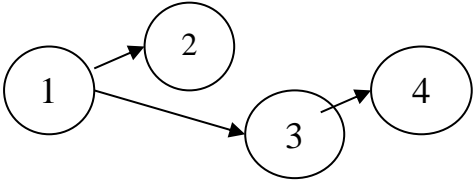
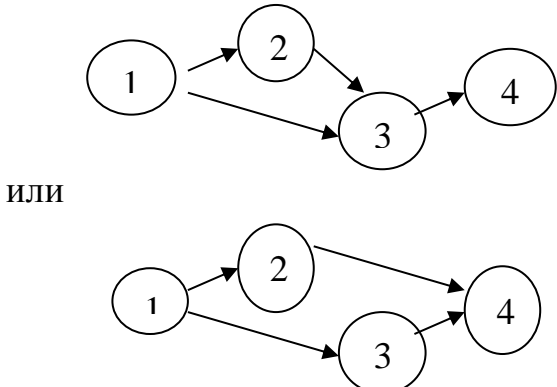
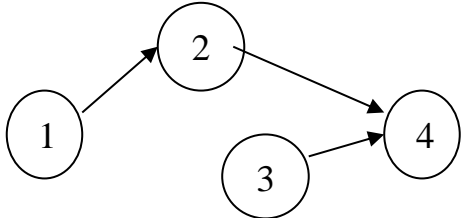
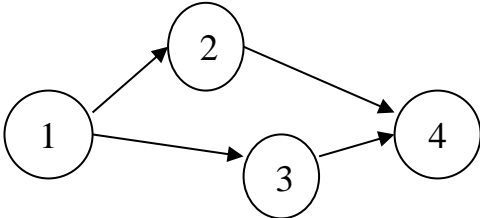
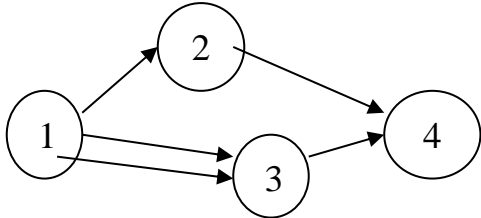
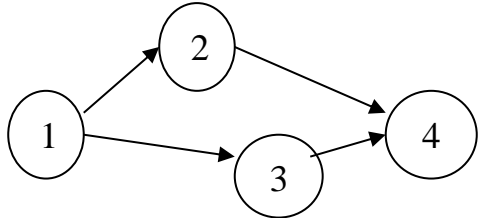
- 1) в сетевом графике должны отсутствовать «тупиковые» события – события, из которых не выходит ни одна работа (за исключением завершающего события);
- 2) в сетевом графике не должно быть «хвостовых» событий, которым не предшествует хотя бы одна работа (помимо исходного события);
- 3) в сети не должно быть замкнутых контуров и петель, т.е. путей, соединяющих события с ними же самими;
- 4) любые два события должны быть непосредственно связаны не более чем

одной работой;

5) в сети рекомендуется иметь одно исходное и одно завершающее события [18].

В таблице 5.3 представлены примеры вариантов не правильного и правильного построения сетевого графика.

Таблица 5.3 – Варианты верного и не верного представления сетевого графика

| Не правильно | Правильно |
|---|--|
|  |  |
|  |  |
|  |  |

В итоге строится сетевой график, после упорядочения которого рассчитываются параметры работ и событий, определяются резервы времени и критический путь. В завершении проводится анализ и оптимизация сетевого графика, который при необходимости перестраивается заново с пересчетом параметров работ и событий [19].

Упорядочение сетевого графика заключается в таком расположении событий и

работ, при котором для любой работы, предшествующее ей событие расположено левее и имеет меньший номер по сравнению с завершающим эту работу событием. Для построения упорядоченного сетевого графика введем понятие ранг события.

Рангом события z_i назовем максимальное число работ, соединяющих данное событие с начальным событием сетевого графика.

Алгоритм построения правильной нумерации в сетевом графике (упорядочения сетевого графика):

- 1) фиксируем начальное событие z_1 ;
- 2) берем все события первого ранга и нумеруем их от 2 до r_1 ;
- 3) берем все события второго ранга и нумеруем их от r_1+1 до r_2 ;
- 4) и так далее;
- 5) завершающее событие нумеруем n .

Пример 5.2

По данным примера 5.1, рассмотренного в предыдущем параграфе, проведем упорядочение сетевого графика.

Как видно из рисунка 5.2 построенный сетевой график не является упорядоченным, поскольку события с большими номерами расположены левее событий с меньшими номерами.

Упорядоченный сетевой график представлен на рисунке 5.3.

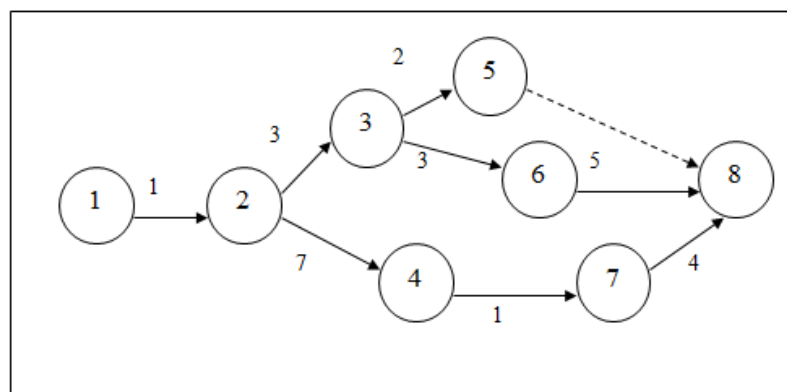


Рисунок 5.3 – Упорядоченный сетевой график выполнения работ по реализации продукции предприятия

Классический вид сетевого графика – это сеть, вычерченная без масштаба времени. Поэтому сетевой график, хотя и дает четкое представление о порядке следования работ, но недостаточно нагляден для определения тех работ, которые должны выполняться в каждый момент времени. В связи с этим рекомендуется дополнять сетевой график линейной диаграммой проекта. При построении линейной диаграммы каждая работа изображается отрезком, параллельным оси времени, длина которого равна продолжительности этой работы. Фиктивная работа нулевой продолжительности изображается точкой. События i и j , являющиеся началом и концом работы, помещают соответственно в начале и конце отрезка. Отрезки располагаются один над другим, снизу вверх в порядке возрастания индекса i , а при одном и том же i – в порядке возрастания индекса j . По линейной диаграмме можно определить критическое время, критический путь, а также резервы времени всех работ [19]. Линейная диаграмма также называется диаграммой Ганта.

Пример 5.3

По данным примера 5.1 построим линейную диаграмму проекта по реализации продукции предприятия (рисунок 5.4).

Из линейной диаграммы следует, что длина критического пути составляет 13 дней.

5.3 Параметры сетевого графика

В число основных параметров сетевого графика входят: продолжительность критического пути, наиболее ранние и наиболее поздние допустимые сроки начала и окончания работ, различные виды резервов времени [30].

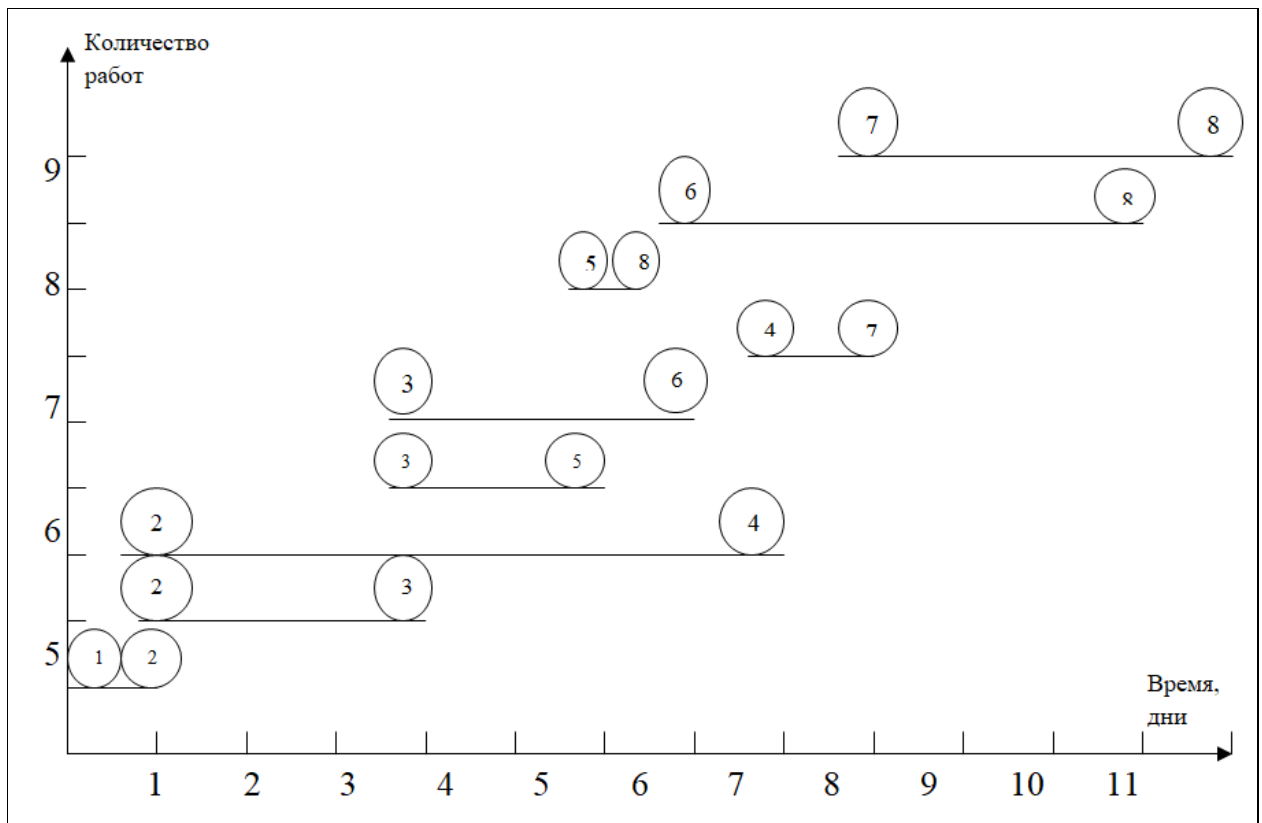


Рисунок 5.4 – Линейная диаграмма проекта

Введем обозначения:

i – номер предшествующего события;

j – номер последующего события;

(i, j) – номер работы;

t_{ij} – продолжительность работы;

$t(L)$ – продолжительность пути;

$t_{кр}$ – продолжительность критического пути;

t_i – время свершения предшествующего события;

t_j – время свершения последующего события;

t_i^p – ранний срок свершения события;

t_i^n – поздний срок свершения события;

$t_{i,j}^{p,n}$ – ранний срок начала работы;

$t_{i,j}^{n,n}$ – поздний срок начала работы;

$t_{i,j}^{p,o}$ – ранний срок окончания работы;

$t_{i,j}^{n,o}$ – поздний срок окончания работы;

R_i – резерв времени на свершение i -го события;

$R(L)$ – полный резерв времени пути;

$R_{i,j}^p$ – общий (полный) резерв времени работы;

$R_{i,j}^{(1)}$ – частный резерв времени работы первого вида;

$R_{i,j}^{(2)}$ – частный резерв времени работы второго вида (свободный резерв времени);

$R_{i,j}^n$ – независимый резерв времени работы.

В первую очередь определяются ранние сроки начала и окончания работ, начиная от исходного события и заканчивая завершающим. На основе вычисленных ранних сроков определяется критический путь. Далее находят поздние сроки начала и окончания работ и рассчитывают резервы времени [30].

Приведем формулы для расчета параметров сетевого графика.

1. Параметры событий:

1) ранний (ожидаемый) срок t_i^p свершения i -го события определяется продолжительностью максимального пути, предшествующего этому событию:

$$t_i^p = \max_{L_{ni}} t(L_{ni}), \quad (5.1)$$

где L_{ni} – любой путь, предшествующий i -му событию (путь от исходного до i -го события сети).

Ранний срок свершения первого события t_1^p равен нулю. Если событие j имеет несколько предшествующих путей, а, следовательно, и несколько предшествующих событий i , то ранний срок свершения события j определяется по формуле:

$$t_j^p = \begin{cases} 0, & j=1 \\ \max_{i < j} (t_i^p + t_{i,j}), & j = \overline{2, n}; \end{cases} \quad (5.2)$$

2) поздний (предельный) срок свершения последнего n -го события t_n^p равен его раннему сроку t_n^p . Если событие i имеет несколько последующих путей, то есть несколько последующих событий j , то поздний срок свершения события i определяется по формуле:

$$t_i^p = \begin{cases} t_n^p, & i = n \\ \min_{j > i} (t_j^p - t_{i,j}), & i = \overline{1, n-1}. \end{cases} \quad (5.3)$$

3) резерв времени R_i i -го события определяется как разность между поздним и ранним сроками его свершения:

$$R_i = t_i^p - t_i^p. \quad (5.4)$$

ОПРЕДЕЛЕНИЕ. Резерв времени события показывает, на какой допустимый период времени можно задержать наступление этого события, не вызывая при этом увеличения срока выполнения комплекса работ.

Критические события резервов времени не имеют, так как любая задержка в свершении события, лежащего на критическом пути, вызовет такую же задержку в свершении завершающего события. Таким образом, чтобы определить длину и топологию критического пути, не нужно перебирать все полные пути сетевого графика и определять их длины [19]. Определив ранний срок наступления завершающего события сети, находится длина критического пути, а путем выявления событий с нулевыми резервами времени, определяется его топология.

Пример 5.4

По данным примера 5.1 рассчитаем параметры событий сетевого графика. Результаты представлены в таблице 5.4 и на рисунке 5.5.

Таблица 5.4 – Результаты расчета параметров событий

| № события | Ранний срок свершения события | Поздний срок свершения события | Резерв времени события |
|-----------|--|--|---------------------------|
| 1 | $t_1^p = 0$ | $t_1^n = t_2^n - t_{1,2} = 1 - 1 = 0$ | $R_1 = t_1^n - t_1^p = 0$ |
| 2 | $t_2^p = t_1^p + t_{1,2} = 0 + 1 = 1$ | $t_2^n = \min \left\{ \begin{array}{l} t_3^n - t_{2,3} \\ t_4^n - t_{2,4} \end{array} \right\} = \min \left\{ \begin{array}{l} 5 - 3 \\ 8 - 7 \end{array} \right\} = 1$ | $R_2 = t_2^n - t_2^p = 0$ |
| 3 | $t_3^p = t_2^p + t_{2,3} = 1 + 3 = 4$ | $t_3^n = \min \left\{ \begin{array}{l} t_5^n - t_{3,5} \\ t_6^n - t_{3,6} \end{array} \right\} = \min \left\{ \begin{array}{l} 13 - 2 \\ 8 - 3 \end{array} \right\} = 5$ | $R_3 = t_3^n - t_3^p = 1$ |
| 4 | $t_4^p = t_2^p + t_{2,4} = 1 + 7 = 8$ | $t_4^n = t_7^n - t_{4,7} = 9 - 1 = 8$ | $R_4 = t_4^n - t_4^p = 0$ |
| 5 | $t_5^p = t_3^p + t_{3,5} = 4 + 2 = 6$ | $t_5^n = t_8^n - t_{5,8} = 13 - 0 = 13$ | $R_5 = t_5^n - t_5^p = 7$ |
| 6 | $t_6^p = t_3^p + t_{3,6} = 4 + 3 = 7$ | $t_6^n = t_8^n - t_{6,8} = 13 - 5 = 8$ | $R_6 = t_6^n - t_6^p = 1$ |
| 7 | $t_7^p = t_4^p + t_{4,7} = 8 + 1 = 9$ | $t_7^n = t_8^n - t_{7,8} = 13 - 4 = 9$ | $R_7 = t_7^n - t_7^p = 0$ |
| 8 | $t_8^p = \max \left\{ \begin{array}{l} t_5^p + t_{5,8} \\ t_6^p + t_{6,8} \\ t_7^p + t_{7,8} \end{array} \right\} = \max \left\{ \begin{array}{l} 6 + 0 \\ 7 + 5 \\ 9 + 4 \end{array} \right\} = 13$ | $t_8^n = t_8^p = 13$ | $R_8 = t_8^n - t_8^p = 0$ |

Из таблицы 5.4 следует, что события 1, 2, 4, 7 и 8 обладают нулевым резервом

времени, следовательно, относятся к критическому пути. Представим результаты расчета параметров событий на рисунке 5.5.

На рисунке 5.5 выделен критический путь, включающий работы (1;2), (2;4), (4;7), (7;8). Рассчитаем длину критического пути:

$$t_{кр} = t_{1,2} + t_{2,4} + t_{4,7} + t_{7,8} = 1 + 7 + 1 + 4 = 13 \text{ дней.}$$

2. Параметры работ [19].

Отдельная работа может начаться (и окончиться) в ранние, поздние или другие промежуточные сроки. В дальнейшем при оптимизации сетевого графика возможно любое размещение работы в заданном интервале.

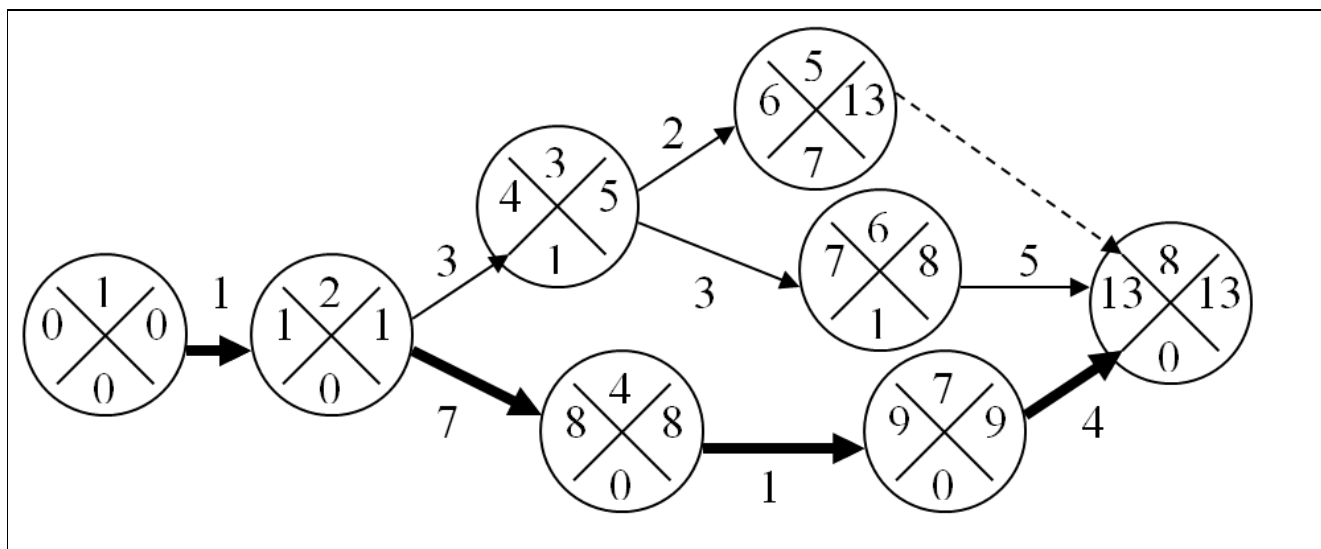


Рисунок 5.5 – Сетевой график с рассчитанными параметрами событий

1) ранний срок $t_{i,j}^{p,h}$ начала работы (i, j) совпадает с ранним сроком наступления начального (предшествующего) события i:

$$t_{i,j}^{p,h} = t_i^p; \quad (5.5)$$

2) ранний срок $t_{i,j}^{p,o}$ окончания работы (i, j) определяется по формуле:

$$t_{i,j}^{p,o} = t_i^p + t_{ij}; \quad (5.6)$$

3) ни одна работа не может закончиться позже допустимого позднего срока своего конечного события i . Поэтому поздний срок $t_{i,j}^{n,o}$ окончания работы (i,j) определяется соотношением:

$$t_{i,j}^{n,o} = t_j^n, \quad (5.7)$$

4) поздний срок $t_{i,j}^{n,h}$ начала работы определяется по формуле:

$$t_{i,j}^{n,h} = t_j^n - t_{ij}. \quad (5.8)$$

Такими образом, в рамках сетевой модели, моменты начала и окончания работ тесно связаны с соседними событиями [19].

Пример 5.5

Рассчитаем параметры работ по данным рассмотренного выше примера. В таблице 5.5 представлены результаты расчетов.

Таблица 5.5 – Результаты расчета параметров работ

| Обозначение работы | Ранний срок начала работы | Ранний срок окончания работы | Поздний срок окончания работы | Поздний срок начала работы |
|--------------------|-----------------------------|---|-------------------------------|---|
| (1;2) | $t_{1,2}^{p,h} = t_1^p = 0$ | $t_{1,2}^{p,o} = t_1^p + t_{1,2} = 0 + 1 = 1$ | $t_{1,2}^{n,o} = t_2^n = 1$ | $t_{1,2}^{n,h} = t_2^n - t_{1,2} = 1 - 1 = 0$ |
| (2;3) | $t_{2,3}^{p,h} = t_2^p = 1$ | $t_{2,3}^{p,o} = t_2^p + t_{2,3} = 1 + 3 = 4$ | $t_{2,3}^{n,o} = t_3^n = 5$ | $t_{2,3}^{n,h} = t_3^n - t_{2,3} = 5 - 3 = 2$ |
| (2;4) | $t_{2,4}^{p,h} = t_2^p = 1$ | $t_{2,4}^{p,o} = t_2^p + t_{2,4} = 1 + 7 = 8$ | $t_{2,4}^{n,o} = t_4^n = 8$ | $t_{2,4}^{n,h} = t_4^n - t_{2,4} = 8 - 7 = 1$ |
| (3;5) | $t_{3,5}^{p,h} = t_3^p = 4$ | $t_{3,5}^{p,o} = t_3^p + t_{3,5} = 4 + 2 = 6$ | $t_{3,5}^{n,o} = t_5^n = 13$ | $t_{3,5}^{n,h} = t_5^n - t_{3,5} = 13 - 2 = 11$ |

Продолжение таблицы 5.5

| | | | | |
|-------|-----------------------------|--|------------------------------|---|
| (3;6) | $t_{3,6}^{p,H} = t_3^p = 4$ | $t_{3,6}^{p,o} = t_3^p + t_{3,6} = 4 + 3 = 7$ | $t_{3,6}^{n,o} = t_6^n = 8$ | $t_{3,6}^{n,H} = t_6^n - t_{3,6} = 8 - 3 = 5$ |
| (4;7) | $t_{4,7}^{p,H} = t_4^p = 8$ | $t_{4,7}^{p,o} = t_4^p + t_{4,7} = 8 + 1 = 9$ | $t_{4,7}^{n,o} = t_7^n = 9$ | $t_{4,7}^{n,H} = t_7^n - t_{4,7} = 9 - 1 = 8$ |
| (5;8) | $t_{5,8}^{p,H} = t_5^p = 6$ | $t_{5,8}^{p,o} = t_5^p + t_{5,8} = 6 + 0 = 6$ | $t_{5,8}^{n,o} = t_8^n = 13$ | $t_{5,8}^{n,H} = t_8^n - t_{5,8} = 13 - 0 = 13$ |
| (6;8) | $t_{6,8}^{p,H} = t_6^p = 7$ | $t_{6,8}^{p,o} = t_6^p + t_{6,8} = 7 + 5 = 12$ | $t_{6,8}^{n,o} = t_8^n = 13$ | $t_{6,8}^{n,H} = t_8^n - t_{6,8} = 13 - 5 = 8$ |
| (7;8) | $t_{7,8}^{p,H} = t_7^p = 9$ | $t_{7,8}^{p,o} = t_7^p + t_{7,8} = 9 + 4 = 13$ | $t_{7,8}^{n,o} = t_8^n = 13$ | $t_{7,8}^{n,H} = t_8^n - t_{7,8} = 13 - 4 = 9$ |

Для работ, лежащих на критическом пути, ранний срок начала работы совпадает с поздним сроком окончания работы и ранний срок окончания работы равен позднему сроку ее начала.

3. Резерв времени пути $R(L)$ определяется как разность между длиной критического и рассматриваемого пути:

$$R(L) = t_{кр} - t(L). \quad (5.9)$$

Резерв времени пути показывает на сколько в сумме могут быть увеличены продолжительности всех работ, принадлежащих этому пути. Если выполнение работ, лежащих на этом пути, запоздает на время большее, чем $R(L)$, то критический путь переместится на путь L . Следовательно, любая из работ пути L , лежащая на участке, не совпадающем с критическим путем, обладает резервом времени [19].

Пример 5.6

Рассчитаем величину резерва пути $1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 8$.

Продолжительность рассматриваемого пути будет равна:

$$t(L) = t_{1,2} + t_{2,3} + t_{3,6} + t_{6,8} = 1 + 3 + 3 + 5 = 12 \text{ дней.}$$

Тогда резерв времени рассматриваемого пути составляет:

$$R(L) = t_{кр} - t(L) = 13 - 12 = 1 \text{ день.}$$

4. Резервы времени работ [18]:

1) полный резерв времени $R_{i,j}^{\Pi}$ работы (i, j) показывает, на сколько можно увеличить время выполнения данной работы при условии, что срок выполнения комплекса работ не изменится. Полный резерв времени работы определяется по формуле:

$$R_{i,j}^{\Pi} = t_j^{\Pi} - t_i^{\Pi} - t_{ij}. \quad (5.10)$$

Полный резерв времени работы равен резерву максимального из путей, проходящих через данную работу. Важным свойством полного резерва времени работы является то, что он принадлежит не только этой работе, но и всем полным путям, проходящим через нее;

2) частный резерв времени первого вида $R_{i,j}^{(1)}$ работы (i, j) является частью полного резерва времени, на которую можно увеличить продолжительность работы, не изменив при этом позднего срока ее начального события:

$$R_{i,j}^{(1)} = t_j^{\Pi} - t_i^{\Pi} - t_{ij} = R_{ij}^{\Pi} - R_i; \quad (5.11)$$

3) частный резерв времени второго вида $R_{i,j}^{(2)}$ (свободный резерв времени) работы (i, j) представляет часть полного резерва времени, на которую можно увеличить продолжительность работы, не изменив при этом раннего срока ее конечного события:

$$R_{i,j}^{(2)} = t_j^p - t_i^p - t_{ij}; \quad (5.12)$$

Свободным резервом времени можно пользоваться для предотвращения случайностей, которые могут возникнуть в ходе выполнения работ. Если планировать выполнение работ по ранним срокам их начала и окончания, то всегда будет возможность при необходимости перейти на поздние сроки начала и окончания работ [19];

4) независимый резерв времени $R_{i,j}^H$ работы (i, j) – это часть полного резерва времени, получаемая для случая, когда все предшествующие работы заканчиваются в поздние сроки, а все последующие работы начинаются в свои ранние сроки:

$$R_{i,j}^H = t_j^p - t_i^H - t_{ij}. \quad (5.13)$$

Использование независимого резерва времени не влияет на величину резервов времени других работ. Независимые резервы стремятся использовать тогда, когда окончание предыдущей работы произошло в поздний допустимый срок, а последующие работы необходимо выполнить в ранние сроки. Если величина независимого резерва не отрицательна, то такая возможность есть [19].

Работы, лежащие на критическом пути, так же как и критические события, резервов времени не имеют.

Если на критическом пути лежит начальное событие i , то

$$R_{i,j}^H = R_{i,j}^{(1)}. \quad (5.14)$$

Если на критическом пути лежит конечное событие j , то

$$R_{i,j}^H = R_{i,j}^{(2)}. \quad (5.15)$$

Если на критическом пути лежат начальное и конечное события i и j , но сама

работа не принадлежит этому пути, то

$$R_{i,j}^{\Pi} = R_{i,j}^{(1)} = R_{i,j}^{(2)} = R_{i,j}^H. \quad (5.16)$$

Соотношения (5.14)-(5.16) можно использовать при проверке правильности расчетов резервов времени отдельных работ.

Пример 5.7

Проведем расчет резервов времени выполнения работ по данным примера 1. Результаты представлены в таблицах 5.6-5.7.

Таблица 5.6 – Результаты расчета полного и независимого резервов времени работ

| Обозначение работы | Полный резерв времени | Независимый резерв времени |
|--------------------|---|--|
| (1;2) | $R_{1,2}^{\Pi} = t_2^{\Pi} - t_1^P - t_{1,2} =$ $= 1 - 0 - 1 = 0$ | $R_{1,2}^H = t_2^P - t_1^{\Pi} - t_{1,2} =$ $= 1 - 0 - 1 = 0$ |
| (2;3) | $R_{2,3}^{\Pi} = t_3^{\Pi} - t_2^P - t_{2,3} =$ $= 5 - 1 - 3 = 1$ | $R_{2,3}^H = t_3^P - t_2^{\Pi} - t_{2,3} =$ $= 4 - 1 - 3 = 0$ |
| (2;4) | $R_{2,4}^{\Pi} = t_4^{\Pi} - t_2^P - t_{2,4} =$ $= 8 - 1 - 7 = 0$ | $R_{2,4}^H = t_4^P - t_2^{\Pi} - t_{2,4} =$ $= 8 - 1 - 7 = 0$ |
| (3;5) | $R_{3,5}^{\Pi} = t_5^{\Pi} - t_3^P - t_{3,5} =$ $= 13 - 4 - 2 = 7$ | $R_{3,5}^H = t_5^P - t_3^{\Pi} - t_{3,5} =$ $= 6 - 5 - 2 = -1$ |
| (3;6) | $R_{3,6}^{\Pi} = t_6^{\Pi} - t_3^P - t_{3,6} =$ $= 8 - 4 - 3 = 1$ | $R_{3,6}^H = t_6^P - t_3^{\Pi} - t_{3,6} =$ $= 7 - 5 - 3 = -1$ |
| (4;7) | $R_{4,7}^{\Pi} = t_7^{\Pi} - t_4^P - t_{4,7} =$ $= 9 - 8 - 1 = 0$ | $R_{4,7}^H = t_7^P - t_4^{\Pi} - t_{4,7} =$ $= 9 - 8 - 1 = 0$ |
| (5;8) | $R_{5,8}^{\Pi} = t_8^{\Pi} - t_5^P - t_{5,8} =$ $= 13 - 6 - 0 = 7$ | $R_{5,8}^H = t_8^P - t_5^{\Pi} - t_{5,8} =$ $= 13 - 13 - 0 = 0$ |
| (6;8) | $R_{6,8}^{\Pi} = t_8^{\Pi} - t_6^P - t_{6,8} =$ $= 13 - 7 - 5 = 1$ | $R_{6,8}^H = t_8^P - t_6^{\Pi} - t_{6,8} =$ $= 13 - 8 - 5 = 0$ |
| (7;8) | $R_{7,8}^{\Pi} = t_8^{\Pi} - t_7^P - t_{7,8} =$ $= 13 - 9 - 4 = 0$ | $R_{7,8}^H = t_8^P - t_7^{\Pi} - t_{7,8} =$ $= 13 - 9 - 4 = 0$ |

Итак, полные резервы времени критических работ равны нулю. Также из таблицы 5.6 видно, что независимые резервы времени для работ (3;5) и (3;6) являются отрицательными, следовательно, если предыдущая работа (2;3) закончится в поздний допустимый срок, то выделенные работы не смогут начаться в ранний срок.

В таблице 5.7 сведены результаты расчета частных резервов времени работ:

- 1) частного резерва времени первого вида;
- 2) свободного резерва времени работы.

Таблица 5.7 – Результаты расчета частных резервов времени работ

| Обозначение работы | Частный резерв времени первого вида | Частный резерв времени второго вида |
|--------------------|--|---|
| (1;2) | $R_{1,2}^{(1)} = t_2^{\Pi} - t_1^{\Pi} - t_{1,2} =$ $= 1 - 0 - 1 = 0$ | $R_{1,2}^{(2)} = t_2^{\text{P}} - t_1^{\text{P}} - t_{1,2} =$ $= 1 - 0 - 1 = 0$ |
| (2;3) | $R_{2,3}^{(1)} = t_3^{\Pi} - t_2^{\Pi} - t_{2,3} =$ $= 5 - 1 - 3 = 1$ | $R_{2,3}^{(2)} = t_3^{\text{P}} - t_2^{\text{P}} - t_{2,3} =$ $= 4 - 1 - 3 = 0$ |
| (2;4) | $R_{2,4}^{(1)} = t_4^{\Pi} - t_2^{\Pi} - t_{2,4} =$ $= 8 - 1 - 7 = 0$ | $R_{2,4}^{(2)} = t_4^{\text{P}} - t_2^{\text{P}} - t_{2,4} =$ $= 8 - 1 - 7 = 0$ |
| (3;5) | $R_{3,5}^{(1)} = t_5^{\Pi} - t_3^{\Pi} - t_{3,5} =$ $= 13 - 5 - 2 = 6$ | $R_{3,5}^{(2)} = t_5^{\text{P}} - t_3^{\text{P}} - t_{3,5} =$ $= 6 - 4 - 2 = 0$ |
| (3;6) | $R_{3,6}^{(1)} = t_6^{\Pi} - t_3^{\Pi} - t_{3,6} =$ $= 8 - 5 - 3 = 0$ | $R_{3,6}^{(2)} = t_6^{\text{P}} - t_3^{\text{P}} - t_{3,6} =$ $= 7 - 4 - 3 = 0$ |
| (4;7) | $R_{4,7}^{(1)} = t_7^{\Pi} - t_4^{\Pi} - t_{4,7} =$ $= 9 - 8 - 1 = 0$ | $R_{4,7}^{(2)} = t_7^{\text{P}} - t_4^{\text{P}} - t_{4,7} =$ $= 9 - 8 - 1 = 0$ |
| (5;8) | $R_{5,8}^{(1)} = t_8^{\Pi} - t_5^{\Pi} - t_{5,8} =$ $= 13 - 13 - 0 = 0$ | $R_{5,8}^{(2)} = t_8^{\text{P}} - t_5^{\text{P}} - t_{5,8} =$ $= 13 - 6 - 0 = 7$ |
| (6;8) | $R_{6,8}^{(1)} = t_8^{\Pi} - t_6^{\Pi} - t_{6,8} =$ $= 13 - 8 - 5 = 0$ | $R_{6,8}^{(2)} = t_8^{\text{P}} - t_6^{\text{P}} - t_{6,8} =$ $= 13 - 7 - 5 = 1$ |
| (7;8) | $R_{7,8}^{(1)} = t_8^{\Pi} - t_7^{\Pi} - t_{7,8} =$ $= 13 - 9 - 4 = 0$ | $R_{7,8}^{(2)} = t_8^{\text{P}} - t_7^{\text{P}} - t_{7,8} =$ $= 13 - 9 - 4 = 0$ |

На основе результатов расчета частного резерва времени первого вида можно сделать вывод, что время выполнения работ (2;3) и (3;5) может быть увеличено на 1

день и 6 дней соответственно без изменения позднего срока их начальных событий.

Поскольку частный резерв времени второго вида для работы (6;8) равен 1 дню, то продолжительность ее выполнения также можно увеличить, не изменив при этом раннего срока конечного события.

5.4 Оценка времени выполнения работ в условиях неопределенности

При определении временных параметров сетевого графика в предыдущем параграфе предполагалось, что время выполнения каждой работы известно. В реальности такое предположение выполняется достаточно редко. Чаще всего продолжительность времени выполнения работ t_{ij} является случайной величиной, распределенной по некоторому закону распределения с математическим ожиданием $\overline{t_{ij}}$ и дисперсией σ_{ij}^2 .

Практически во всех системах сетевого планирования предполагается, что плотность распределения продолжительности работ обладает свойствами: непрерывностью, унимодальностью и отлична от нуля на некотором промежутке, за пределами которого тождественно равна нулю. Кроме того, установлено, что распределение продолжительности выполнения работ обладает положительной асимметрией (максимум кривой смещен влево относительно медианы). Распределение, как правило, более круто поднимается при удалении от минимального значения t и полого опускается при приближении к максимальному значению t (рисунок 5.6) [19]. Простейшим распределением с подобными свойствами является β -распределение.

Случайная величина $\beta(\alpha_1, \alpha_2)$, подчиняющаяся закону β -распределения с параметрами α_1 и α_2 ($0 < \alpha_1 < \infty$, $0 < \alpha_2 < \infty$), имеет плотность вероятности [2]:

$$f_{\beta(\alpha_1, \alpha_2)}(x) = \begin{cases} \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1) \cdot \Gamma(\alpha_2)} \cdot x^{\alpha_1 - 1} \cdot (1 - x)^{\alpha_2 - 1}, & 0 \leq x \leq 1; \\ 0. & \end{cases} \quad (5.17)$$

Отметим, что случайная величина $\beta(1,1)$ распределена равномерно на отрезке $[0;1]$.

Основные числовые характеристики β -распределенной (с параметрами α_1 и α_2) случайной величины $\beta(\alpha_1, \alpha_2)$ определяются по формулам [2]:

$$M\beta(\alpha_1, \alpha_2) = \frac{\alpha_1}{\alpha_1 + \alpha_2}, \quad (5.18)$$

$$D\beta(\alpha_1, \alpha_2) = \frac{\alpha_1 \cdot \alpha_2}{(\alpha_1 + \alpha_2)^2 \cdot (\alpha_1 + \alpha_2 + 1)}. \quad (5.19)$$

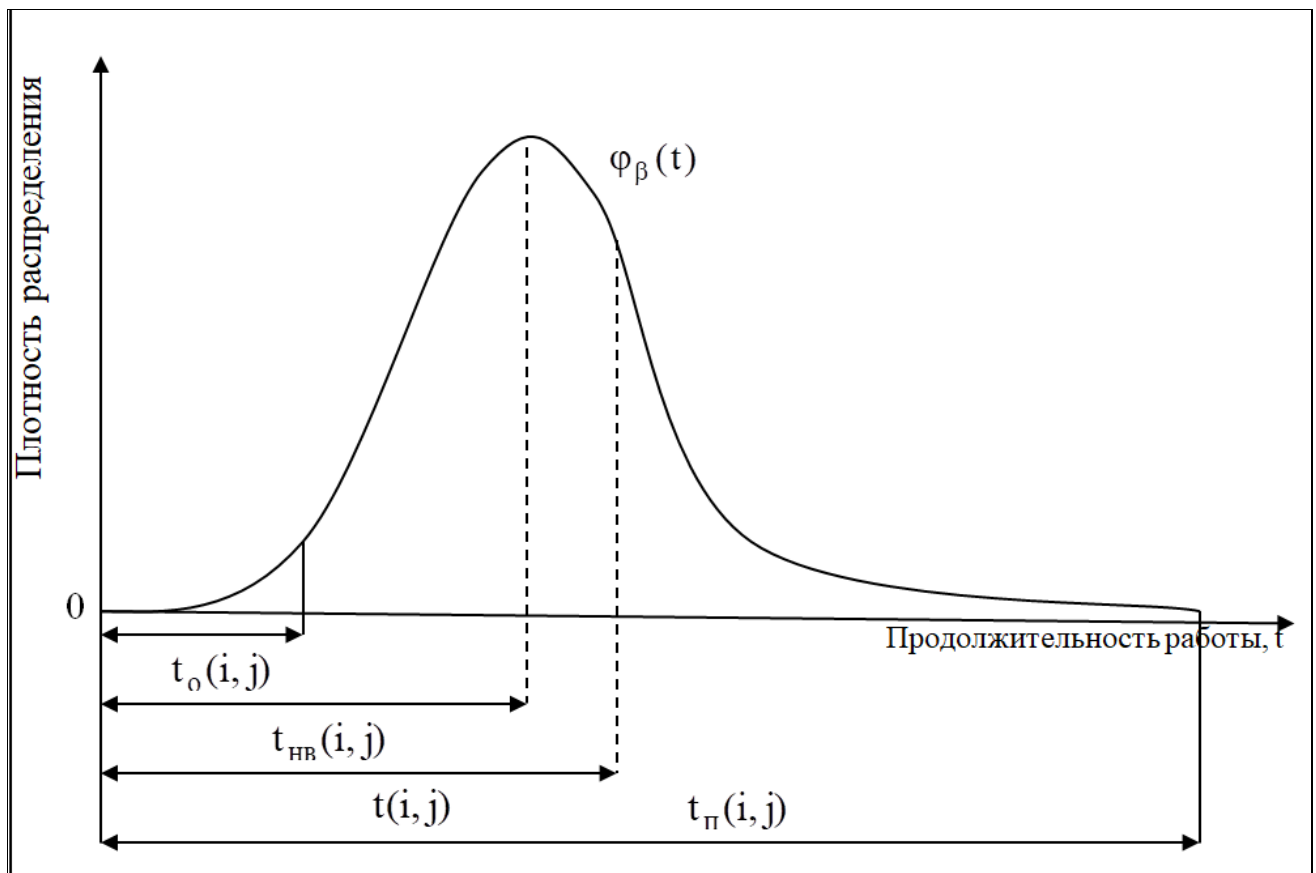


Рисунок 5.6 – Плотность β -распределения времени выполнения работ

Для определения числовых характеристик \bar{t}_{ij} и $\hat{\sigma}_{ij}^2$ этого распределения на основе опроса ответственных исполнителей проекта и экспертов определяют три временные оценки (рисунок 5.6) [18]:

1) оптимистическую оценку $t_o(i, j)$ – продолжительность работы при самых благоприятных условиях;

2) пессимистическую оценку $t_{\pi}(i, j)$ – продолжительность работы при самых неблагоприятных условиях;

3) наиболее вероятную оценку $t_{\text{нв}}(i, j)$.

Предположение о β -распределении продолжительности работы позволяет получить следующие оценки ее числовых характеристик [19]:

$$\bar{t}_{ij} = \frac{t_o(i, j) + 4 \cdot t_{\text{нв}}(i, j) + t_{\pi}(i, j)}{6}, \quad (5.20)$$

$$\hat{\sigma}_{ij}^2 = \left[\frac{t_{\pi}(i, j) - t_o(i, j)}{6} \right]^2. \quad (5.21)$$

Отметим, что в случае невозможности получения оценки наиболее вероятного времени выполнения работы, в реальных проектах используется упрощенная формула оценки средней продолжительности работы на основе только двух задаваемых временных оценок $t_o(i, j)$ и $t_{\pi}(i, j)$ [19]:

$$\bar{t}_{ij} = \frac{2 \cdot t_o(i, j) + 3 \cdot t_{\pi}(i, j)}{5}. \quad (5.22)$$

При достаточно большом количестве работ, принадлежащих пути L , можно применить центральную предельную теорему Ляпунова, на основании которой предполагается, что общая продолжительность пути L имеет нормальный закон распределения со средним значением $\bar{t}(L)$, равным сумме средних значений продолжительности составляющих его работ \bar{t}_{ij} и дисперсией $\hat{\sigma}^2(L)$, равной сумме

соответствующих дисперсий $\hat{\sigma}_{ij}^2$ [19]:

$$\bar{t}(L) = \sum_{i,j} \bar{t}_{ij}, \quad (5.23)$$

$$\hat{\sigma}^2(L) = \sum_{i,j} \hat{\sigma}_{ij}^2. \quad (5.24)$$

Отметим, что временные параметры сетевого графика – длина критического пути, ранние и поздние сроки свершения событий, резервы времени – будут определяться по формулам, представленным в параграфе 5.3. Но, при этом, рассчитанные параметры будут являться средними значениями соответствующих случайных величин. Поэтому предварительный анализ сетей со случайными продолжительностями работ, не ограничивается расчетом временных параметров сети. Важным этапом анализа является оценка вероятности того, что срок выполнения проекта $t_{кр}$ не превысит заданного срока T [18]. Полагая, что $t_{кр}$ – случайная величина, распределенная по нормальному закону распределения, получим:

$$P(t_{кр} \leq T) = \frac{1}{2} + \frac{1}{2} \cdot \Phi\left(\frac{T - \bar{t}_{кр}}{\hat{\sigma}_{кр}}\right), \quad (5.25)$$

где $\Phi(x)$ – функция нормального распределения.

В некоторых задачах представляет интерес решение обратной задачи – определение максимального срока выполнения проекта T , который возможен с заданной надежностью α . В этом случае:

$$T = \bar{t}_{кр} + t_{\alpha} \cdot \hat{\sigma}_{кр}, \quad (5.26)$$

где из уравнения

$$\Phi(t_{\alpha}) = \alpha, \quad (5.27)$$

находим квантиль уровня α .

Таким образом, рассмотрели общий подход к анализу времени выполнения работ с учетом неопределенности.

5.5 Анализ и оптимизация сетевого графика

Анализ сетевого графика начинается с определения степени возможности выполнения в срок каждой группы работ не критического пути с помощью коэффициента напряженности работ [19].

ОПРЕДЕЛЕНИЕ. Коэффициентом напряженности K_n работы (i, j) называется отношение продолжительности несовпадающих (заклученных между одними и теми же событиями) отрезков пути, одним из которых является путь максимальной продолжительности, проходящий через данную работу, а другим – критический путь [18]:

$$K_n(i, j) = \frac{t(L_{\max}) - t'_{кр}}{t_{кр} - t'_{кр}}, \quad (5.27)$$

где $t(L_{\max})$ – продолжительность максимального пути, проходящего через работу (i, j) ;

$t_{кр}$ – длина критического пути;

$t'_{кр}$ – продолжительность отрезка рассматриваемого пути, совпадающего с критическим путем.

Коэффициент напряженности изменяется в интервале $K_n \in [0;1]$.

Для определения коэффициента напряженности работы также можно применить формулу [19]:

$$K_H(i, j) = 1 - \frac{R_{i,j}^{\Pi}}{t_{кр} - t'_{кр}}, \quad (5.28)$$

где $R_{i,j}^{\Pi}$ – полный резерв времени работы (i, j) .

Чем ближе коэффициент напряженности к 1, тем сложнее выполнить работу в установленные сроки. Чем ближе K_H к нулю, тем большим относительным резервом обладает максимальный путь, проходящий через работу. Коэффициент напряженности фиктивных работ равен 0, а работ, лежащих на критическом пути, – 1. Коэффициент напряженности позволяет классифицировать работы по зонам [19]:

- 1) критическая зона – $K_H(i, j) > 0,8$;
- 2) подкритическая зона – $0,6 \leq K_H(i, j) \leq 0,8$;
- 3) резервная зона – $K_H(i, j) < 0,6$.

Оптимизация сетевого графика представляет собой процесс улучшения организации выполнения комплекса работ с учетом срока его выполнения. Оптимизация проводится с целью сокращения длительности критического пути, выравнивания коэффициентов напряженности работ, рационального использования ресурсов [18].

В первую очередь принимаются меры по сокращению продолжительности работ, находящихся на критическом пути. Это достигается:

- перераспределением всех видов ресурсов, как временных (за счет использования резервов времени некритических путей), так и трудовых, и материальных. При этом перераспределение ресурсов должно идти из менее напряженных зон в области, объединяющие наиболее напряженные работы;

- сокращением трудоемкости критических работ в следствие передачи части работ на другие пути, имеющие резервы времени;

- параллельным выполнением работ критического пути;

- пересмотром топологии сети, изменением состава работ и структуры сети [19].

В процессе сокращения продолжительности работ критический путь может измениться и дальнейший процесс оптимизации будет направлен на сокращение продолжительности работ нового критического пути.

Рассмотрим задачу оптимизации сетевого графика с точки зрения трудовых ресурсов. Ставится задача, используя резервы времени работ, определить оптимальную численность рабочих при заданном сроке выполнения работы. Критерием будет служить минимизация функции суммы квадратов отклонений требуемого количества работников от их среднего числа:

$$I = \frac{1}{T} \sum_{t=1}^T (R(t) - R_{cp.})^2 + [\max R(t) - \min R(t)]^2 \rightarrow \min, \quad (5.29)$$

где $R(t)$ – потребность в рабочих в каждый момент времени;

R_{cp} – средняя потребность в рабочих;

T – общая продолжительность проекта.

Так как функция $R(t)$ является ступенчатой, то минимизация данного функционала традиционными методами невозможна. В связи с этим требуется разработать собственный алгоритм.

Для решения поставленной задачи на основе нами в работе [33] были использованы следующие алгоритмы.

I. Приближённый метод оптимизации сетевого графика по используемым трудовым ресурсам.

1. В качестве начального выбираем план, где все работы начинаются в свои самые ранние сроки. Вычисляем значение критерия:

$$I_0 = \sum_i (R_i - R_{cp})^2, \quad (5.30)$$

где R_i – количество рабочих в i -й день.

1. Счетчик итераций $q:=nr$.
2. Задаем начальные значения для величины сдвига m текущей работы и величины оптимального приращения функции затрат $I \Delta I$: $m:=0, \Delta I:=0$.
3. Вычисление резерва для сдвига работы q :

$$R_{\text{сдвига}} = t^{\text{п.о.}}(q) - T_{\text{конца}} - t(q). \quad (5.31)$$

4. Если $R_{\text{сдвига}} > 0$, то определяем оптимальную величину сдвига m текущей работы и величину оптимального приращения ΔI функции затрат I .
5. Если $\Delta I \leq 0$, то вычисляем новое значение функции I : $I := I + \Delta I$.
6. Проводим сдвиг текущей работы на величину m :

$$\begin{aligned} T_{\text{нач}} &:= T_{\text{нач}}(q) + m, \\ T_{\text{кон}} &:= T_{\text{кон}}(q) + m; \end{aligned}$$
7. Выбор следующей работы для сдвига: $q:=q-1$.
8. Если $q=0$, то конец алгоритма, иначе выбираем следующую работу, идем на шаг 3.

Этот алгоритма имеет следующие недостатки:

1. текущее оптимальное расположение некоторой работы при фиксированном положении остальных работ могут дать решение весьма далекое от оптимального;
2. результаты данного алгоритма зависят от порядка перебора работ.

II Метод Монте-Карло (имитационного моделирования) оптимизации сетевого графика по используемым трудовым ресурсам.

1. В качестве начального выбираем план, где все работы начинаются в свои самые ранние сроки. Вычисляем значение критерия (5.29). Вводится N – количество итераций.

2. Счетчик итераций $k:=0$.

3. Случайным образом формируем новый план, определяем для него ежедневную потребность в рабочих. Вычисляем для него значение критерия:

$$I_1 = \sum_i (R'_i - R_{\text{ср}})^2 + (\max(R') - \min(R'))^2. \text{ Увеличиваем счетчик итераций } k:=k+1.$$

4. Сравниваем значения критериев I_0 и I_1 .

Если $I_0 > I_1$, тогда в качестве текущего выбираем полученный план, $I_0 = I_1$ $I_0 := I_1$.

5. Если $k < N$, то идем на шаг 3.

6. Демонстрация ЛПР текущего решения. Если данное решение устраивает ЛПР, выбираем его в качестве оптимального, иначе идем на шаг 2.

III Полный перебор

Также для оптимизации может быть использован алгоритм полного перебора всех возможных планов. Такой алгоритм достаточно эффективный, но он требует очень больших вычислительных затрат. Поэтому он может быть использован только для сетевых графиков небольшой размерности. Либо возможно его применение для групп работ, но такой подход не гарантирует получение оптимального решение.

5.6 Программные средства для сетевого моделирования

Для управления проектами существуют такие готовые решения как Microsoft Enterprise Project Management Solution (EPM), Oracle Primavera P6, Deltek Open Plan, облачный сервис Wrike и другие. Среди российских разработок отметим Spider Project компании «Технологии управления спайдер», Project Kaiser компании «Triniforce». Среди функционала таких программ следует отметить: планирование задач и установка приоритетов, назначение исполнителей, диаграмма Ганта, распределение и анализ ресурсов, возможность генерировать отчеты и считать сметы и ряд других специфичных функций (ментальные карты, kanban-доски).

Рассмотрим возможности MS Visio для построения линейной диаграммы проекта. При создании нового проекта необходимо выбрать в разделе «Расписания» шаблон «Диаграмма Ганта» (рисунок 5.7).

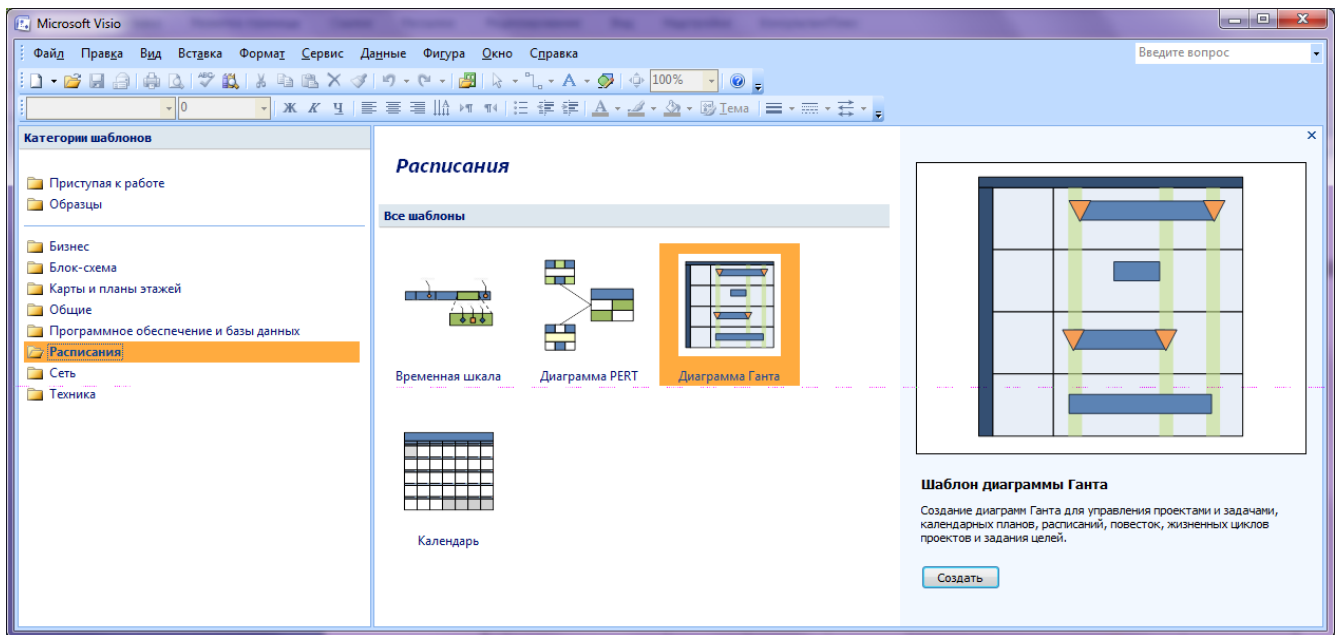


Рисунок 5.7 – Выбор шаблона для построения диаграммы

Далее мы можем задать единицы измерения для времени, количество задач и другие параметры диаграммы (рисунок 5.8).

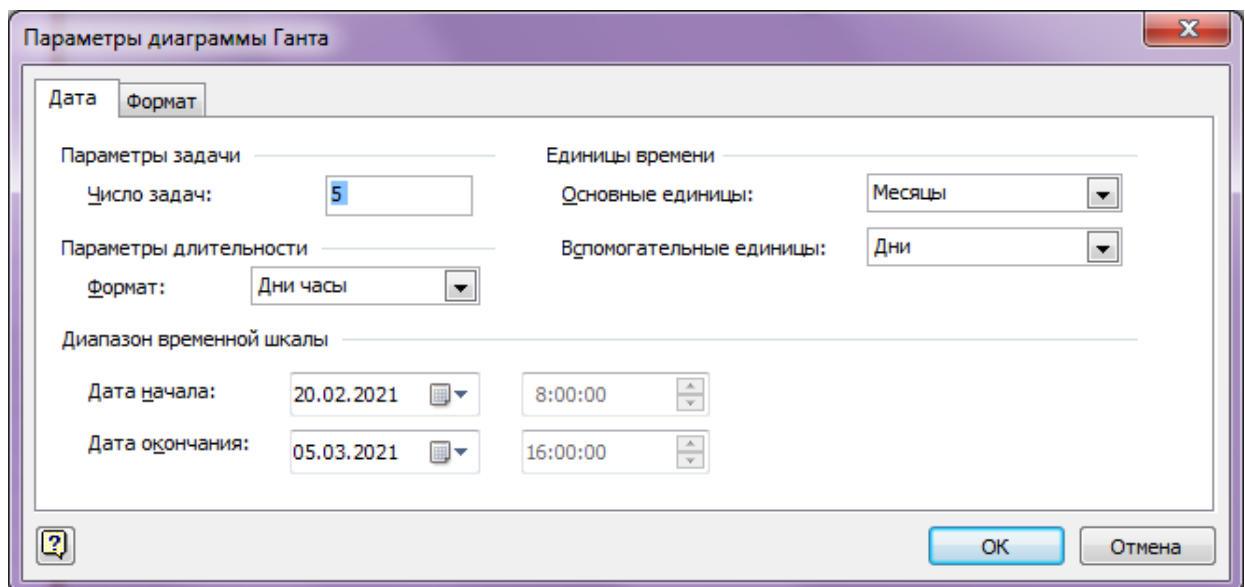


Рисунок 5.8 – Форма задания параметров линейной диаграммы Ганта

После указания параметров появится шаблон, в котором в столбец «Название задачи» вводятся содержательное условие, указывается начало, конец и длительность задачи (рисунок 5.9). Отметим, что поля «Начало», «Конец» и «Длительность»

взаимосвязаны и вычисляемы.

В поле справа будет изображаться сама диаграмма по заданным данным. Используя фигуры для диаграммы можно добавить: столбцы, задачи, подзадачи, связи между задачами, выноски, название диаграммы (рисунок 5.10).

Пример 5.8

Построим линейную диаграммы для задачи реализации продукции мебельного предприятия. Результаты приведены на рисунке 5.11. Отметим взаимосвязь работ, используя динамическое соединение или линии связи с панели рисунок 5.10. К диаграмме можно добавлять комментарии (горизонтальные выноски) в которых делать необходимые пометки, например ответственный исполнитель или статус задачи (рисунок 5.11).

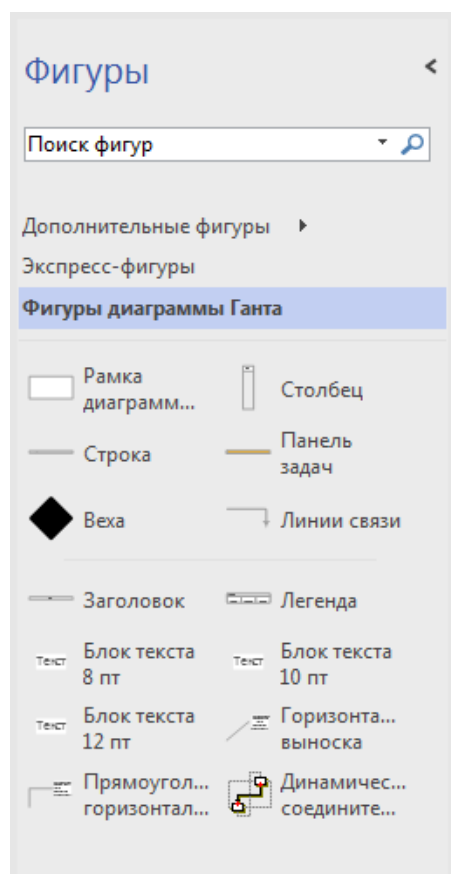


Рисунок 5.10 – Элементы диаграммы Ганта

Продemonстрируем работы автоматизированного программного комплекса

«Исследование и оптимизация сетевых графиков» (АПК «Netgraf») [1] для анализа и оптимизации сетевого графика по критерию (5.29).

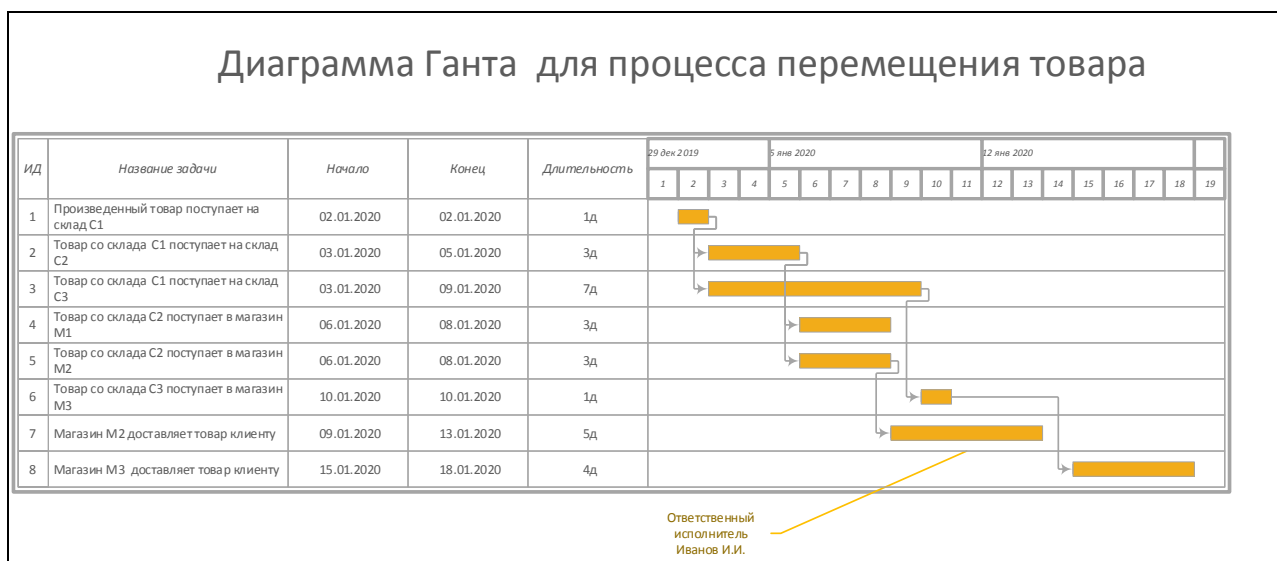


Рисунок 5.11 – Диаграмма Ганта в MS Visio

Пример 5.9

Рассмотрим задачу реализации продукции мебельного предприятия. В таблице 5.8 приведем информацию о трудовых затратах, связанных с доставкой товаров на склады, в магазины и до клиентов.

Рассмотрим линейную диаграмму проекта с указанием общего числа рабочих, необходимых в каждый момент времени (рисунок 5.12).

Таблица 5.8 – Информация о трудовых затратах на выполнение работ

| Обозначение работы | Время выполнения | Число рабочих |
|--------------------|------------------|---------------|
| (1;2) | 1 день | 3 |
| (2;3) | 3 дня | 6 |
| (2;4) | 7 дней | 3 |
| (3;5) | 2 дня | 6 |
| (3;6) | 3 дня | 6 |
| (4;7) | 1 день | 3 |
| (5;8) | мгновенно | 0 |
| (6;8) | 5 дней | 9 |
| (7;8) | 4 дня | 6 |

Например, в седьмой день осуществляется выполнение работ (2;4) и (3;6). Из

таблицы 5.8 видим, что для выполнения этих двух работ необходимо задействовать 9 рабочих.

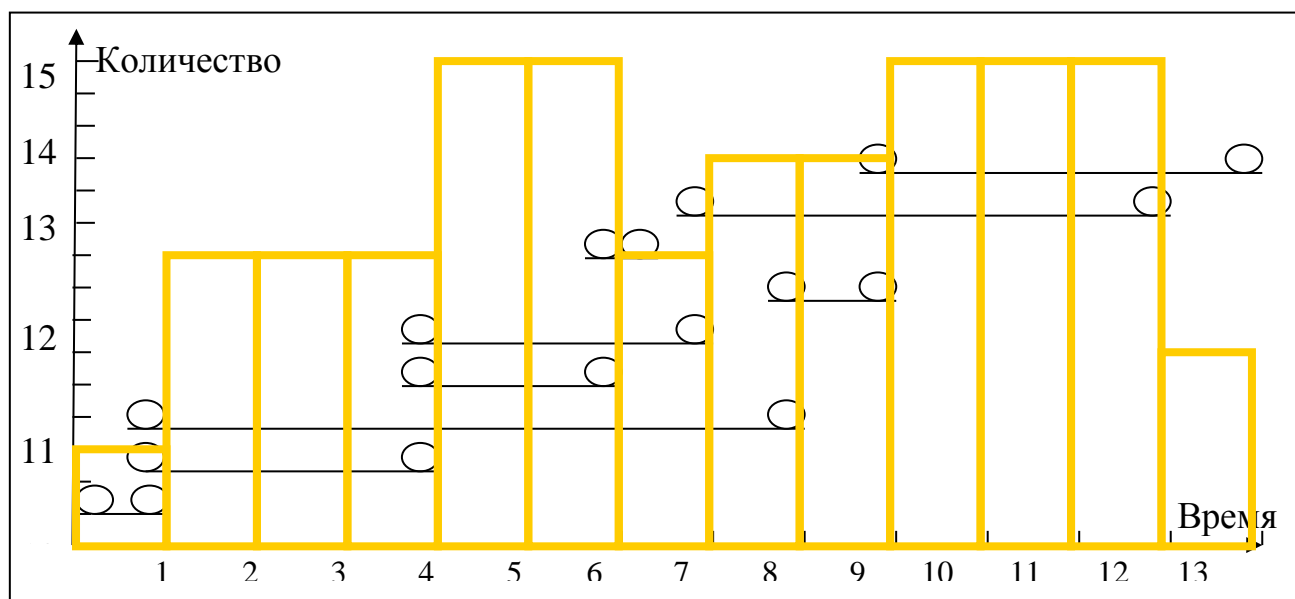


Рисунок 5.12 – Линейная диаграмма проекта с указанием числа рабочих

Проведем оптимизацию трудовых ресурсов в программном средстве АПК «Netgraf». Исходную информацию о работах сетевого графика, времени их выполнения и затратах трудовых ресурсов запишем в блокнот (рисунок 5.13).

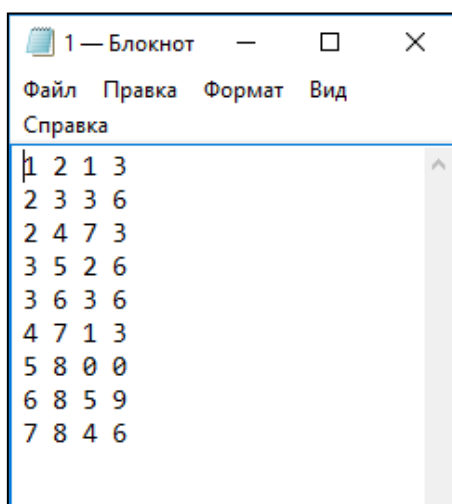


Рисунок 5.13 – Ввод исходной информации в блокнот

На рисунке 5.14 представлен интерфейс программного средства.

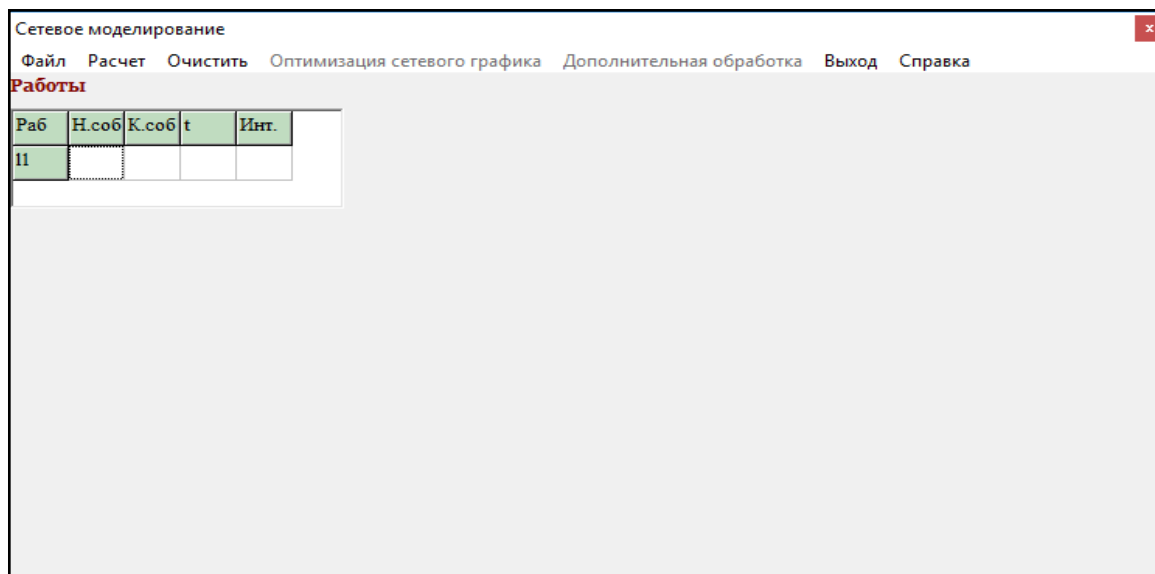


Рисунок 5.14 – Интерфейс АПК «Netgraf»

Выберем «Файл»→«Открыть» и укажем путь к блокноту, в котором введены исходные данные. Результат представлен на рисунке 5.15.

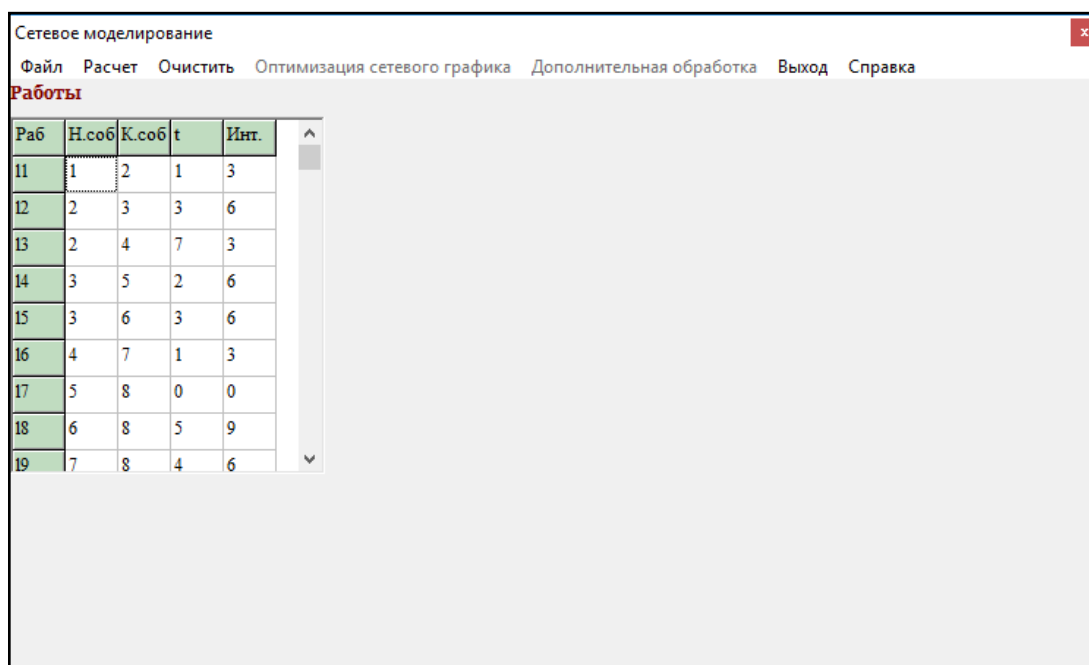


Рисунок 5.15 – Ввод исходных данных в программное средство

Выберем пункт меню «Расчет» и определим сначала характеристики событий (рисунок 5.16), а затем характеристики работ (рисунок 5.17).

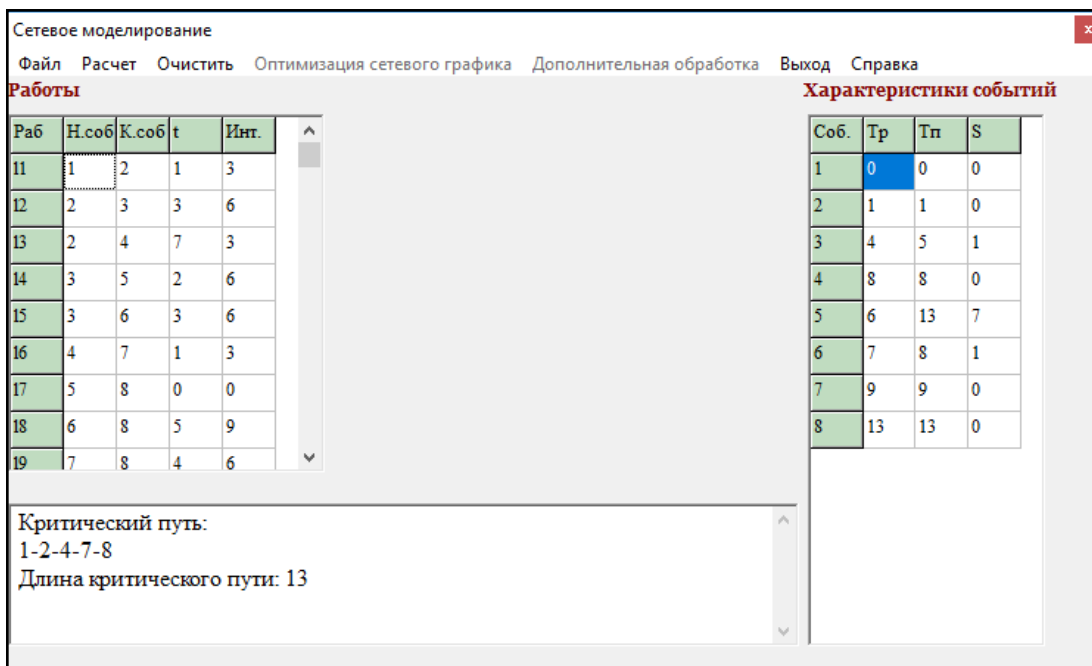


Рисунок 5.16 – Результаты расчета параметров событий

Полученный в программном средстве критический путь совпадает с результатами решения задачи, приведенными в параграфе 5.3.

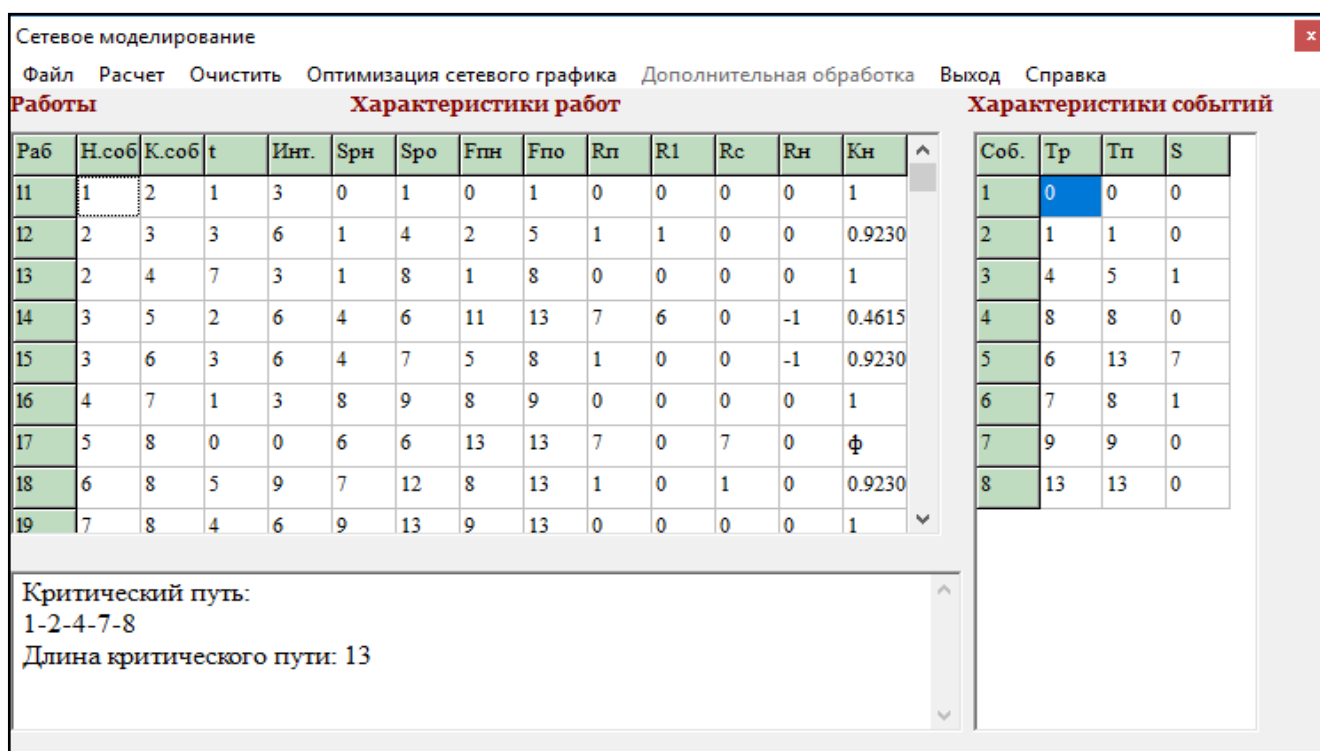


Рисунок 5.17 – Результаты расчета параметров работ

На рисунке 5.17 представлены результаты расчета коэффициентов напряженности работ. Работа (3;5) относится к резервной зоне, следовательно, может быть использована при оптимизации сетевого графика. Остальные работы входят в критическую зону.

Для оптимизации сетевого графика в программном средстве реализовано 3 метода: приближенный алгоритм, имитационное моделирование, полный перебор (рисунок 5.18).

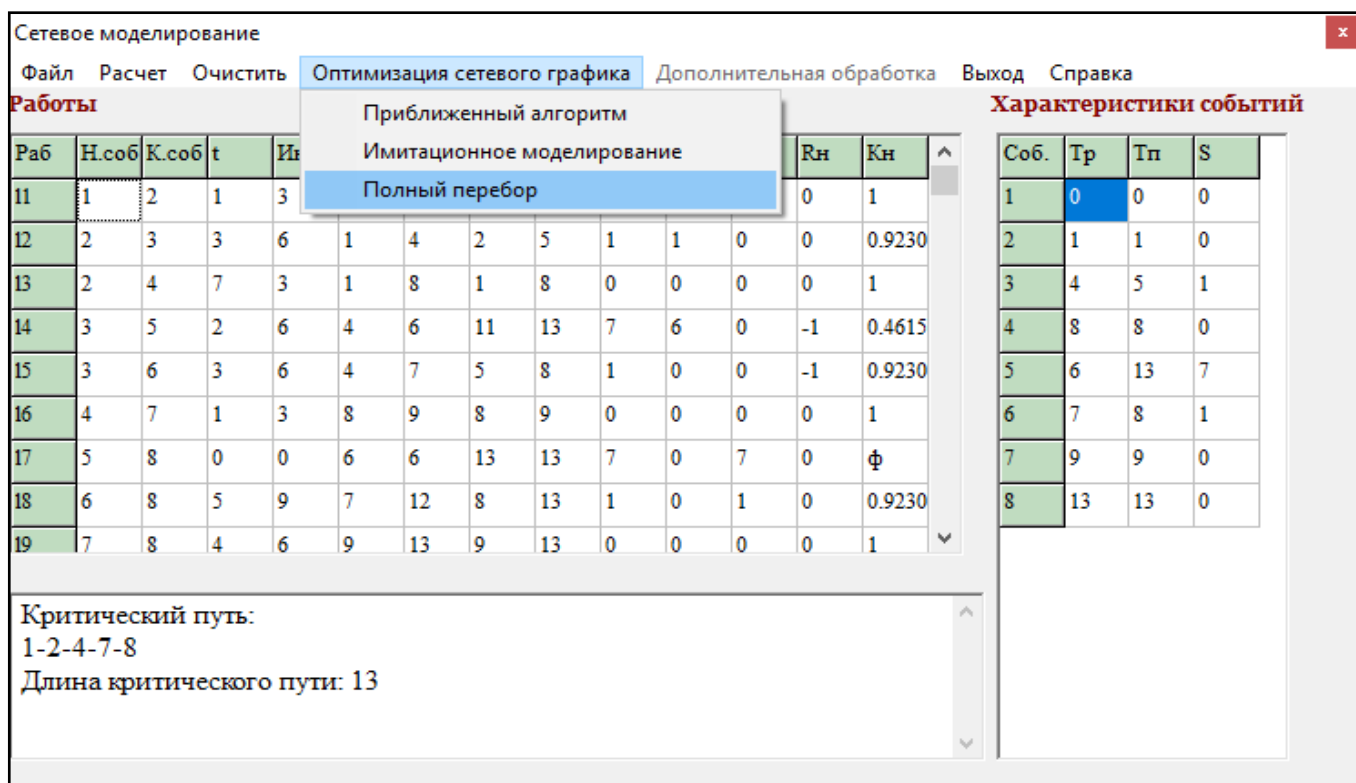


Рисунок 5.18 – Выбор метода оптимизации

Для реализации приближенного алгоритма необходимо будет нажать кнопку «Оптимизация» (рисунок 5.19), в методе имитационного моделирования необходимо будет задать количество имитаций и при необходимости процедуру можно повторить (кнопка «Продолжить?»).

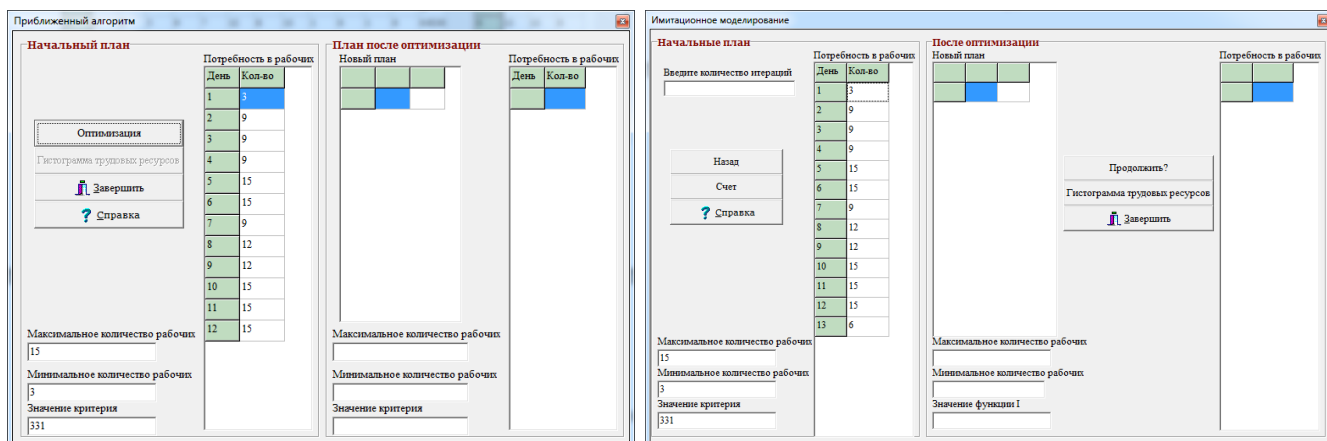


Рисунок 5.19 – окно для проведения оптимизации по приближенному алгоритму (слева) и методу имитационного моделирования

Если какой-то из алгоритмов не дал результатов пользователь увидит сообщение об ошибке (рисунок 5.20), в этой ситуации можно попробовать другой метод.

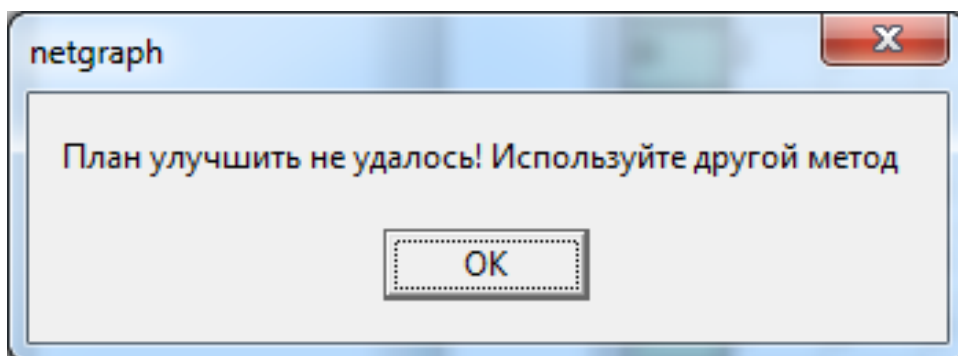


Рисунок 5.20 – сообщение об ошибке

Отметим, что полный перебор работает только для задач небольшой размерности и пользователь увидит сообщение, если для его задачи этот метод не подходит (рисунок 5.19).

Для нашего примера проведем оптимизацию трудовых ресурсов методом полного перебора, форма представлена на рисунке 5.21. Для этого необходимо нажать кнопку «Оптимизация».

На рисунке 5.21 представлены результаты решения задачи оптимизации.

В результате решения задачи оптимизации удалось минимизировать сумму

квадратов отклонений требуемого количества работников от их среднего числа на 5,4% (с 331 до 313). Это достигается за счет сдвига времени выполнения двух работ: работы (3;5) на два дня вперед и работы (6;8) на один день вперед.

Таким образом, предлагается:

- 1) задержать на два дня отгрузку товара со склада C_2 в магазин M_1 ;
- 2) планировать доставку товара клиентам из магазина M_2 , начиная со следующего дня после покупки.

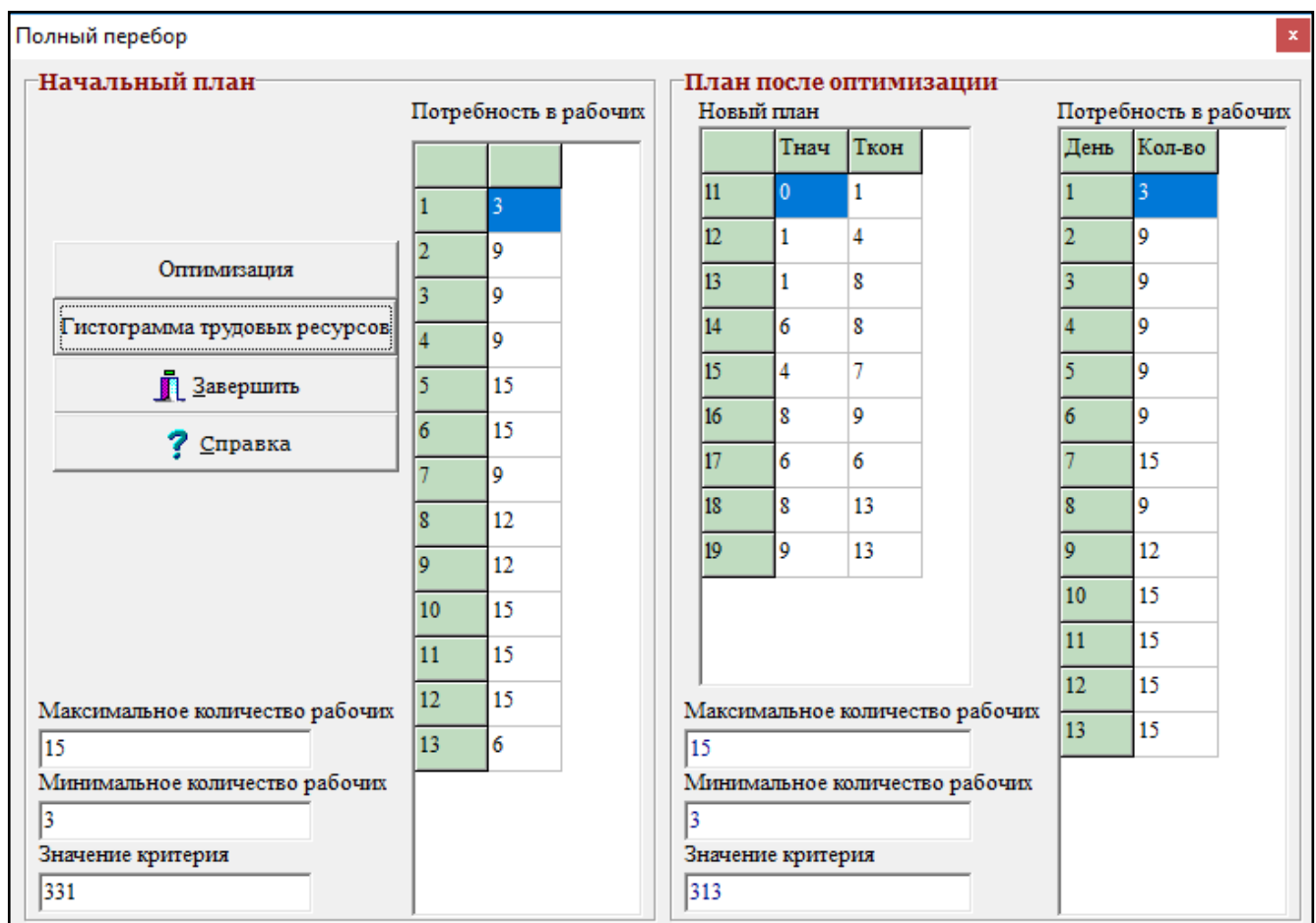


Рисунок 5.21 – Результаты оптимизации трудовых ресурсов

На рисунке 5.22 представим линейную диаграмму проекта после оптимизации.

На рисунке 5.23 представлены графики численности занятых рабочих в каждый момент времени до и после оптимизации, он строится в программе нажав на кнопку «Гистограмма трудовых ресурсов».

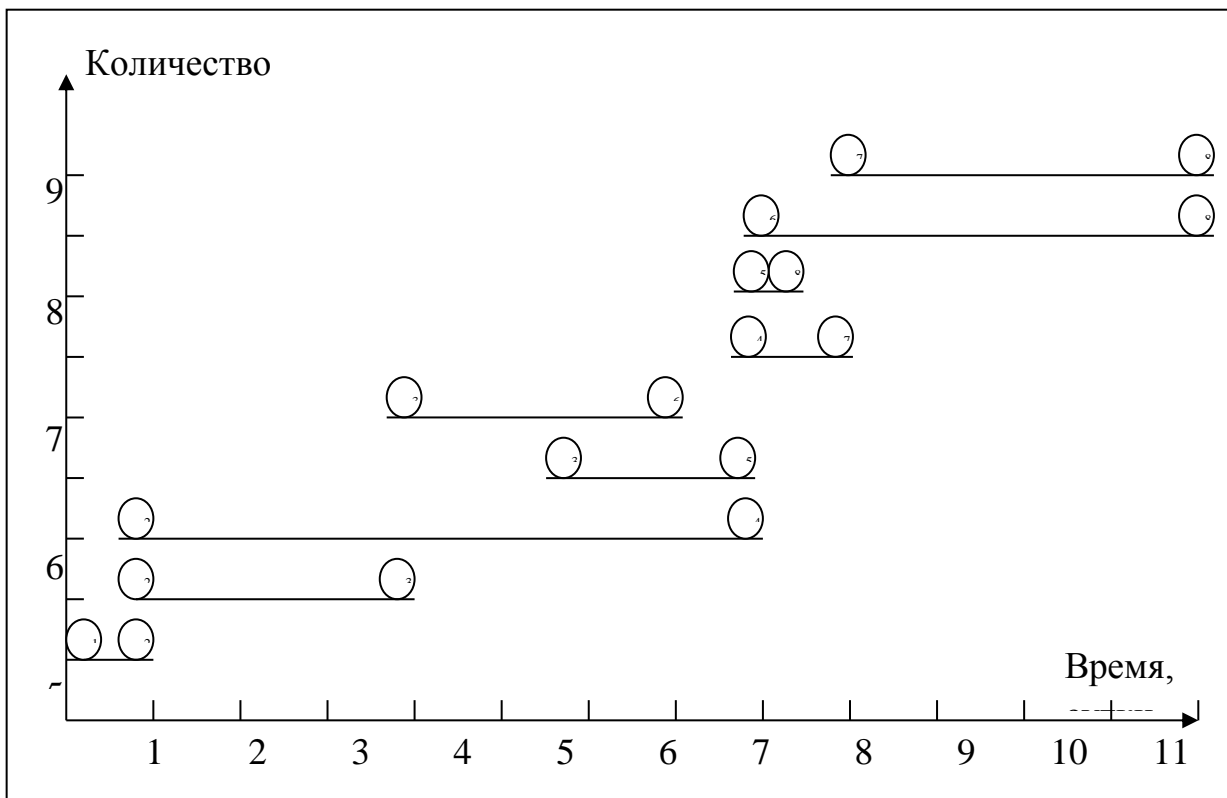


Рисунок 5.22 – Линейная диаграмма проекта после оптимизации

Аналогично может быть проведена оптимизация сетевого графика по материальным ресурсам.

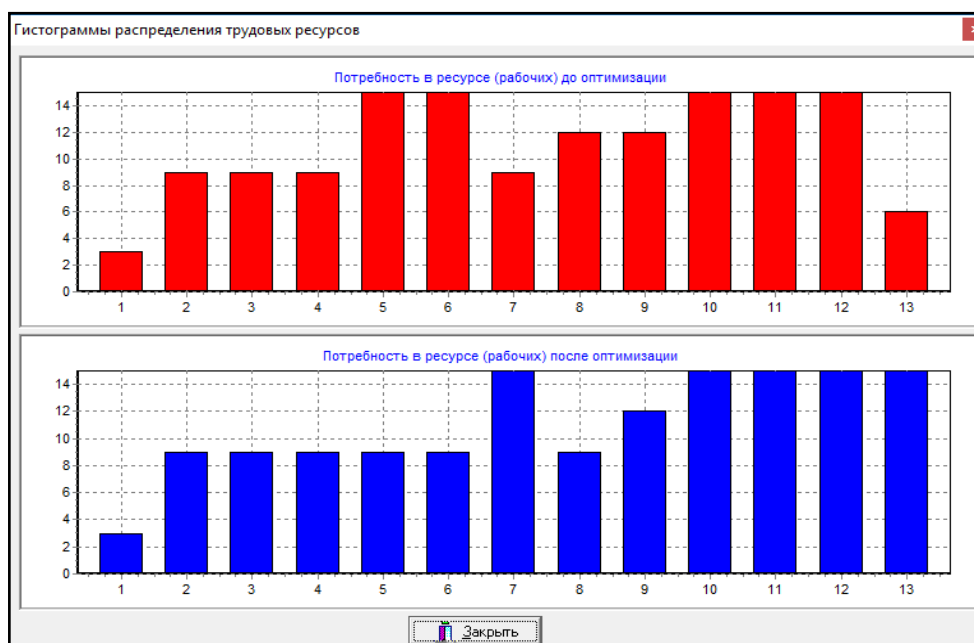


Рисунок 5.23 – Численность занятых рабочих до и после оптимизации

5.7 Вопросы для самопроверки

1. Дайте определение сетевого графика. Из каких компонент он состоит?
2. Какие работы сетевого графика называются фиктивными?
3. Сформулируйте правила построения сетевого графика.
4. Как провести упорядочение сетевого графика?
5. Сформулируйте цель построения линейной диаграммы.
6. Дайте определение критического пути.
7. Приведите формулы расчета параметров событий.
8. Чему равен резерв времени события, входящего в критический путь?
9. Приведите формулы расчета параметров работ.
10. Приведите формулы расчета резервов времени работ.
11. Какие значения может принимать независимый резерв времени выполнения работ?
12. Как определить среднюю продолжительность работ, если известен закон распределения времени выполнения работ?
13. Запишите формулу расчета коэффициента напряженности работ.
14. При каких значениях коэффициентов напряженности работ возможно проведение оптимизации сетевого графика?
15. Сформулируйте задачу оптимизации сетевого графика по трудовым ресурсам.

5.8 Задания для практических (лабораторных работ)

Задание 5.1 Туристическая компания организует экскурсии по достопримечательностям края. Компания предлагает клиентам несколько маршрутов, длительность каждого из которых известна. За каждым участком

маршрута закреплены определенные группы экскурсоводов и инструкторов. Информация о маршрутах, их длительности, а также о затратах трудовых и материальных ресурсов представлена в таблице 5.9.

Требуется:

- 1) построить сетевой график маршрутов туристической компании, провести его упорядочение;
- 2) рассчитать параметры событий, работ и резервы времени;
- 3) определить критический путь;
- 4) рассчитать коэффициенты напряженности работ и провести оптимизацию по трудовым и материальным ресурсам;
- 5) сделать выводы по полученным результатам.

Таблица 5.9 – Исходные данные к задаче 5.1

| Номер работы | Длительность работы (пути), в днях | Затраты трудовых ресурсов, чел. | Материальные затраты, руб. в день. |
|--------------|------------------------------------|---------------------------------|------------------------------------|
| 1 | 2 | 3 | 4 |
| Вариант 1 | | | |
| (1;2) | 5 | 3 | 6500 |
| (1;4) | 8 | 5 | 10500 |
| (2;3) | 3 | 2 | 4000 |
| (2;4) | 4 | 2 | 5000 |
| (3;5) | 10 | 5 | 12500 |
| (3;7) | 12 | 5 | 14500 |
| (4;5) | 2 | 2 | 3000 |
| (4;6) | 1 | 2 | 2000 |
| (5;6) | 1 | 2 | 2000 |
| (6;7) | 3 | 2 | 4000 |
| Вариант 2 | | | |
| (1;2) | 1 | 2 | 2000 |
| (1;3) | 3 | 2 | 4000 |
| (2;4) | 2 | 2 | 3000 |
| (3;4) | 1 | 2 | 2000 |
| (3;5) | 2 | 2 | 3000 |
| (4;5) | 2 | 2 | 3000 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 4 | 3 | 5000 |
| (5;8) | 5 | 3 | 6500 |
| (6;8) | 2 | 2 | 3000 |
| (7;8) | 3 | 2 | 4000 |

Продолжение таблицы 5.9

| 1 | 2 | 3 | 4 |
|-----------|----|---|-------|
| Вариант 3 | | | |
| (1;2) | 2 | 2 | 3000 |
| (1;3) | 3 | 2 | 4000 |
| (1;4) | 7 | 5 | 9000 |
| (2;5) | 10 | 5 | 12500 |
| (3;6) | 8 | 5 | 10500 |
| (4;5) | 1 | 2 | 2000 |
| (4;6) | 2 | 2 | 3000 |
| (5;7) | 3 | 2 | 4000 |
| (6;7) | 3 | 2 | 4000 |
| Вариант 4 | | | |
| (1;2) | 1 | 2 | 2000 |
| (1;4) | 4 | 3 | 5000 |
| (2;3) | 2 | 2 | 3000 |
| (3;4) | 1 | 2 | 2000 |
| (3;5) | 5 | 3 | 6500 |
| (3;6) | 7 | 5 | 9000 |
| (4;5) | 2 | 2 | 3000 |
| (5;6) | 1 | 2 | 2000 |
| (6;7) | 2 | 2 | 3000 |
| Вариант 5 | | | |
| (1;2) | 3 | 2 | 4000 |
| (2;3) | 1 | 2 | 2000 |
| (2;4) | 3 | 2 | 4000 |
| (3;5) | 5 | 5 | 6500 |
| (4;5) | 2 | 2 | 3000 |
| (4;6) | 1 | 2 | 2000 |
| (5;7) | 4 | 3 | 5000 |
| (6;7) | 3 | 2 | 4000 |
| (6;8) | 7 | 5 | 9000 |
| (7;8) | 2 | 2 | 3000 |
| Вариант 6 | | | |
| (1;2) | 2 | 2 | 3000 |
| (1;3) | 3 | 2 | 4000 |
| (1;4) | 5 | 3 | 6500 |
| (2;3) | 1 | 2 | 2000 |
| (2;5) | 7 | 5 | 9000 |
| (3;4) | 2 | 2 | 3000 |
| (3;5) | 5 | 3 | 6500 |
| (4;5) | 3 | 2 | 4000 |
| (4;6) | 7 | 5 | 9000 |
| (5;6) | 2 | 2 | 3000 |

Продолжение таблицы 5.9

| 1 | 2 | 3 | 4 |
|------------|----|---|-------|
| Вариант 7 | | | |
| (1;2) | 2 | 2 | 3000 |
| (1;3) | 4 | 3 | 5000 |
| (2;3) | 1 | 2 | 2000 |
| (2;4) | 5 | 3 | 6500 |
| (3;4) | 3 | 2 | 4000 |
| (4;5) | 1 | 2 | 2000 |
| (4;6) | 4 | 3 | 5000 |
| (5;6) | 2 | 2 | 3000 |
| (5;7) | 5 | 3 | 6500 |
| (6;7) | 3 | 2 | 4000 |
| Вариант 8 | | | |
| (1;2) | 3 | 2 | 4000 |
| (2;3) | 1 | 2 | 2000 |
| (3;4) | 2 | 2 | 3000 |
| (3;5) | 2 | 2 | 3000 |
| (3;6) | 5 | 3 | 6500 |
| (4;5) | 1 | 2 | 2000 |
| (4;7) | 5 | 3 | 6500 |
| (4;8) | 8 | 5 | 10500 |
| (5;7) | 3 | 2 | 4000 |
| (6;7) | 3 | 2 | 4000 |
| (7;8) | 2 | 2 | 3000 |
| Вариант 9 | | | |
| (1;2) | 5 | 3 | 6500 |
| (1;3) | 5 | 3 | 6500 |
| (1;4) | 7 | 5 | 9000 |
| (2;5) | 3 | 2 | 4000 |
| (3;5) | 1 | 2 | 2000 |
| (4;6) | 4 | 3 | 5000 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 3 | 2 | 4000 |
| (6;7) | 5 | 3 | 6500 |
| Вариант 10 | | | |
| (1;2) | 4 | 3 | 5000 |
| (1;3) | 3 | 2 | 4000 |
| (2;4) | 2 | 2 | 3000 |
| (2;5) | 4 | 3 | 5000 |
| (3;5) | 3 | 2 | 4000 |
| (3;6) | 6 | 3 | 8000 |
| (4;7) | 8 | 5 | 10500 |
| (5;7) | 5 | 3 | 6500 |
| (6;8) | 10 | 5 | 12500 |
| (7;8) | 5 | 3 | 6500 |

Продолжение таблицы 5.9

| 1 | 2 | 3 | 4 |
|------------|----|---|-------|
| Вариант 11 | | | |
| (1;2) | 5 | 3 | 6500 |
| (1;4) | 10 | 5 | 12500 |
| (2;3) | 2 | 2 | 3000 |
| (3;4) | 3 | 2 | 4000 |
| (3;5) | 4 | 3 | 5000 |
| (3;6) | 7 | 5 | 9000 |
| (4;5) | 2 | 2 | 3000 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 3 | 2 | 4000 |
| (6;7) | 1 | 2 | 2000 |
| Вариант 12 | | | |
| (1;2) | 2 | 2 | 3000 |
| (1;5) | 8 | 5 | 10500 |
| (2;3) | 2 | 2 | 3000 |
| (3;4) | 2 | 2 | 3000 |
| (3;5) | 3 | 2 | 4000 |
| (4;5) | 1 | 2 | 2000 |
| (4;6) | 3 | 2 | 4000 |
| (5;6) | 2 | 2 | 3000 |
| (5;7) | 4 | 3 | 5000 |
| (6;7) | 2 | 2 | 3000 |
| Вариант 13 | | | |
| (1;2) | 3 | 2 | 4000 |
| (1;3) | 4 | 3 | 5000 |
| (1;4) | 5 | 3 | 6500 |
| (1;5) | 6 | 5 | 8000 |
| (2;3) | 1 | 2 | 2000 |
| (3;6) | 4 | 3 | 5000 |
| (3;7) | 5 | 3 | 6500 |
| (4;7) | 5 | 3 | 6500 |
| (5;8) | 8 | 5 | 10500 |
| (6;7) | 1 | 2 | 2000 |
| (7;8) | 2 | 2 | 3000 |
| Вариант 14 | | | |
| (1;2) | 5 | 3 | 6500 |
| (2;3) | 3 | 2 | 4000 |
| (2;4) | 2 | 2 | 3000 |
| (3;5) | 4 | 3 | 5000 |
| (4;5) | 3 | 2 | 4000 |
| (4;6) | 5 | 3 | 6500 |
| (5;7) | 1 | 2 | 2000 |
| (6;7) | 3 | 2 | 4000 |
| (7;8) | 4 | 3 | 5000 |

Продолжение таблицы 5.9

| 1 | 2 | 3 | 4 |
|------------|----|---|-------|
| Вариант 15 | | | |
| (1;2) | 5 | 3 | 6500 |
| (1;3) | 8 | 5 | 10500 |
| (2;3) | 3 | 2 | 4000 |
| (2;4) | 2 | 2 | 3000 |
| (3;5) | 4 | 3 | 5000 |
| (4;5) | 1 | 2 | 2000 |
| (4;6) | 3 | 2 | 4000 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 2 | 2 | 3000 |
| (6;7) | 1 | 2 | 2000 |
| Вариант 16 | | | |
| (1;2) | 4 | 3 | 5000 |
| (1;3) | 4 | 3 | 5000 |
| (2;4) | 3 | 2 | 4000 |
| (3;4) | 2 | 2 | 3000 |
| (4;5) | 5 | 3 | 6500 |
| (4;6) | 3 | 2 | 4000 |
| (5;7) | 2 | 2 | 3000 |
| (6;7) | 1 | 2 | 2000 |
| Вариант 17 | | | |
| (1;2) | 1 | 2 | 2000 |
| (1;3) | 3 | 2 | 4000 |
| (1;4) | 5 | 3 | 6500 |
| (2;5) | 6 | 5 | 8000 |
| (3;5) | 2 | 2 | 3000 |
| (3;6) | 4 | 3 | 5000 |
| (4;6) | 5 | 3 | 6500 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 2 | 2 | 3000 |
| (6;7) | 3 | 2 | 4000 |
| Вариант 18 | | | |
| (1;2) | 3 | 2 | 4000 |
| (1;3) | 4 | 3 | 5000 |
| (1;4) | 8 | 5 | 10500 |
| (2;3) | 2 | 2 | 3000 |
| (2;5) | 10 | 5 | 12500 |
| (3;5) | 3 | 2 | 4000 |
| (3;6) | 4 | 3 | 5000 |
| (4;6) | 5 | 3 | 6500 |
| (5;6) | 2 | 2 | 3000 |

Продолжение таблицы 5.9

| 1 | 2 | 3 | 4 |
|------------|---|---|------|
| Вариант 19 | | | |
| (1;2) | 1 | 2 | 2000 |
| (1;5) | 5 | 3 | 6500 |
| (2;3) | 1 | 2 | 2000 |
| (3;4) | 2 | 2 | 3000 |
| (4;5) | 1 | 2 | 2000 |
| (4;6) | 3 | 2 | 4000 |
| (4;7) | 5 | 3 | 6500 |
| (5;7) | 4 | 3 | 5000 |
| (6;7) | 1 | 2 | 2000 |
| Вариант 20 | | | |
| (1;2) | 5 | 3 | 6500 |
| (1;3) | 4 | 3 | 5000 |
| (2;3) | 1 | 2 | 2000 |
| (2;4) | 2 | 2 | 3000 |
| (3;4) | 3 | 2 | 4000 |
| (3;5) | 6 | 5 | 8000 |
| (4;5) | 1 | 2 | 2000 |
| (5;6) | 4 | 3 | 5000 |
| (5;7) | 3 | 2 | 4000 |
| (6;8) | 2 | 2 | 3000 |
| (7;8) | 2 | 2 | 3000 |
| Вариант 21 | | | |
| (1;2) | 2 | 2 | 3000 |
| (1;3) | 3 | 2 | 4000 |
| (1;4) | 5 | 3 | 6500 |
| (1;5) | 7 | 5 | 9000 |
| (2;6) | 4 | 3 | 5000 |
| (3;6) | 3 | 2 | 4000 |
| (4;7) | 5 | 3 | 6500 |
| (5;7) | 2 | 2 | 3000 |
| (5;8) | 4 | 3 | 5000 |
| (6;7) | 1 | 2 | 2000 |
| (7;8) | 1 | 2 | 2000 |
| Вариант 22 | | | |
| (1;2) | 4 | 3 | 5000 |
| (2;3) | 2 | 2 | 3000 |
| (3;4) | 5 | 3 | 6500 |
| (3;5) | 3 | 2 | 4000 |
| (4;5) | 1 | 2 | 2000 |
| (4;7) | 6 | 5 | 8000 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 4 | 3 | 5000 |
| (6;7) | 2 | 2 | 3000 |
| (6;8) | 4 | 3 | 5000 |
| (7;8) | 2 | 2 | 3000 |

Продолжение таблицы 5.9

| 1 | 2 | 3 | 4 |
|------------|---|---|-------|
| Вариант 23 | | | |
| (1;2) | 2 | 2 | 3000 |
| (1;3) | 4 | 3 | 5000 |
| (2;4) | 5 | 3 | 6500 |
| (3;6) | 8 | 5 | 10500 |
| (4;5) | 1 | 2 | 2000 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 3 | 2 | 4000 |
| (5;8) | 5 | 3 | 6500 |
| (6;8) | 7 | 5 | 9000 |
| (7;8) | 2 | 2 | 3000 |
| Вариант 24 | | | |
| (1;2) | 2 | 2 | 3000 |
| (1;4) | 4 | 3 | 5000 |
| (2;3) | 1 | 2 | 2000 |
| (3;4) | 3 | 2 | 4000 |
| (3;5) | 5 | 3 | 6500 |
| (4;5) | 2 | 2 | 3000 |
| (4;6) | 7 | 5 | 9000 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 5 | 3 | 6500 |
| (6;7) | 4 | 3 | 5000 |
| Вариант 25 | | | |
| (1;2) | 1 | 2 | 2000 |
| (1;3) | 1 | 2 | 2000 |
| (2;4) | 3 | 2 | 4000 |
| (3;4) | 4 | 3 | 5000 |
| (4;5) | 7 | 5 | 9000 |
| (4;6) | 5 | 3 | 6500 |
| (5;6) | 2 | 2 | 3000 |
| (6;7) | 4 | 3 | 5000 |
| Вариант 26 | | | |
| (1;2) | 2 | 2 | 3000 |
| (2;3) | 4 | 3 | 5000 |
| (3;4) | 3 | 2 | 4000 |
| (3;5) | 3 | 2 | 4000 |
| (4;6) | 2 | 2 | 3000 |
| (4;7) | 5 | 3 | 6500 |
| (5;7) | 4 | 3 | 5000 |
| (6;7) | 1 | 2 | 2000 |
| (6;8) | 3 | 2 | 4000 |
| (7;8) | 5 | 3 | 6500 |

Продолжение таблицы 5.9

| 1 | 2 | 3 | 4 |
|------------|---|---|-------|
| Вариант 27 | | | |
| (1;2) | 3 | 2 | 4000 |
| (1;5) | 7 | 5 | 9000 |
| (2;3) | 1 | 2 | 2000 |
| (2;5) | 4 | 3 | 5000 |
| (3;4) | 1 | 2 | 2000 |
| (4;6) | 5 | 3 | 6500 |
| (5;6) | 3 | 2 | 4000 |
| (5;7) | 4 | 3 | 5000 |
| (6;7) | 2 | 2 | 3000 |
| Вариант 28 | | | |
| (1;2) | 3 | 2 | 4000 |
| (1;3) | 1 | 2 | 2000 |
| (2;4) | 5 | 3 | 6500 |
| (3;5) | 8 | 5 | 10500 |
| (4;5) | 2 | 2 | 3000 |
| (4;6) | 4 | 3 | 5000 |
| (5;6) | 3 | 2 | 4000 |
| (5;7) | 5 | 3 | 6500 |
| (6;7) | 1 | 2 | 2000 |
| (6;8) | 3 | 2 | 4000 |
| (7;8) | 3 | 2 | 4000 |
| Вариант 29 | | | |
| (1;2) | 2 | 2 | 3000 |
| (1;4) | 7 | 5 | 9000 |
| (2;3) | 1 | 2 | 2000 |
| (3;5) | 6 | 5 | 8000 |
| (4;5) | 2 | 2 | 3000 |
| (4;6) | 4 | 3 | 5000 |
| (5;6) | 1 | 2 | 2000 |
| (5;7) | 3 | 2 | 4000 |
| (6;7) | 2 | 2 | 3000 |
| Вариант 30 | | | |
| (1;2) | 1 | 2 | 2000 |
| (1;3) | 4 | 3 | 5000 |
| (1;4) | 3 | 2 | 4000 |
| (1;5) | 5 | 3 | 6500 |
| (2;3) | 1 | 2 | 2000 |
| (3;6) | 6 | 5 | 8000 |
| (4;5) | 1 | 2 | 2000 |
| (5;7) | 3 | 2 | 4000 |
| (6;8) | 5 | 3 | 6500 |
| (7;8) | 4 | 3 | 5000 |

Задание 5.2 Для сетевой модели, представленной на рисунке 5. необходимо:

- 1) упорядочить сетевой график (ввести правильную нумерацию событий);
- 2) построить линейную диаграмму, определить критический путь, длину критического пути;
- 3) рассчитать параметры событий и работ, резервы времени;
- 4) дать содержательную интерпретацию сетевой модели.

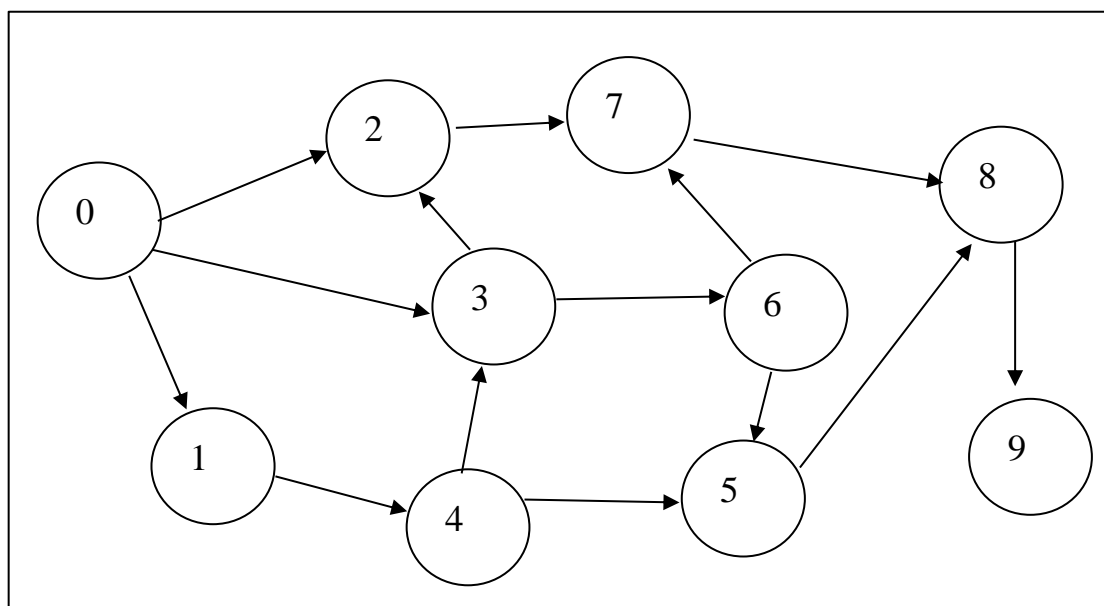


Рисунок 5.24 – Сетевой график к задаче 2

Таблица 5.10 – Исходные данные к задаче 2

| работа | вариант | | | | | | | | | | | | | | | |
|--------|---------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | |
| (0,1) | 2 | 3 | 4 | 2 | 5 | 2 | 4 | 3 | 2 | 8 | 2 | 3 | 4 | 2 | 5 | |
| (0,2) | 3 | 5 | 3 | 4 | 2 | 7 | 5 | 6 | 1 | 1 | 3 | 5 | 3 | 4 | 7 | |
| (0,3) | 4 | 3 | 4 | 2 | 6 | 5 | 3 | 7 | 9 | 3 | 4 | 3 | 4 | 6 | 6 | |
| (1,4) | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 5 | 2 | 2 |
| (2,7) | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 5 | 5 | 5 | |
| (3,2) | 6 | 6 | 6 | 7 | 6 | 7 | 6 | 6 | 6 | 6 | 3 | 6 | 2 | 3 | 5 | |
| (3,6) | 7 | 7 | 5 | 7 | 7 | 7 | 6 | 7 | 2 | 7 | 7 | 1 | 5 | 1 | 7 | |
| (4,5) | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | |
| (5,8) | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| (6,5) | 5 | 5 | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | |
| (6,7) | 4 | 2 | 4 | 4 | 5 | 4 | 4 | 8 | 4 | 1 | 4 | 2 | 4 | 3 | 5 | |
| (7,8) | 3 | 5 | 3 | 3 | 5 | 3 | 3 | 4 | 2 | 6 | 3 | 4 | 3 | 3 | 5 | |
| (8,9) | 6 | 6 | 7 | 6 | 6 | 7 | 6 | 2 | 6 | 3 | 6 | 6 | 1 | 6 | 1 | |

Продолжение таблицы 5.10

| работа | вариант | | | | | | | | | | | | | | |
|--------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| (0,1) | 3 | 3 | 4 | 2 | 15 | 2 | 4 | 3 | 12 | 8 | 2 | 3 | 4 | 2 | 5 |
| (0,2) | 3 | 4 | 3 | 14 | 2 | 17 | 5 | 16 | 1 | 11 | 3 | 5 | 3 | 4 | 7 |
| (0,3) | 4 | 3 | 15 | 2 | 6 | 5 | 13 | 7 | 9 | 3 | 14 | 3 | 4 | 6 | 6 |
| (1,4) | 2 | 12 | 2 | 12 | 2 | 12 | 2 | 12 | 2 | 2 | 2 | 12 | 5 | 2 | 2 |
| (2,7) | 15 | 5 | 5 | 5 | 15 | 5 | 5 | 5 | 15 | 5 | 5 | 4 | 15 | 5 | 5 |
| (3,2) | 6 | 16 | 6 | 17 | 6 | 17 | 6 | 6 | 6 | 16 | 3 | 6 | 2 | 13 | 5 |
| (3,6) | 7 | 7 | 15 | 7 | 7 | 7 | 16 | 7 | 2 | 7 | 17 | 1 | 5 | 1 | 17 |
| (4,5) | 3 | 13 | 3 | 13 | 3 | 3 | 3 | 13 | 3 | 3 | 3 | 13 | 3 | 13 | 3 |
| (5,8) | 12 | 2 | 2 | 2 | 12 | 2 | 1 | 5 | 12 | 2 | 2 | 2 | 12 | 2 | 2 |
| (6,5) | 5 | 15 | 5 | 5 | 5 | 15 | 3 | 5 | 5 | 15 | 5 | 15 | 5 | 15 | 5 |
| (6,7) | 4 | 2 | 4 | 4 | 5 | 4 | 14 | 8 | 4 | 1 | 14 | 2 | 4 | 3 | 15 |
| (7,8) | 3 | 5 | 3 | 3 | 5 | 3 | 3 | 14 | 2 | 16 | 3 | 14 | 3 | 13 | 5 |
| (8,9) | 6 | 6 | 7 | 6 | 6 | 7 | 6 | 2 | 16 | 3 | 6 | 6 | 11 | 6 | 1 |

5.9 Задания для самостоятельной проработки и подготовке к контрольным работам

Задача 1 Для представленного сетевого графика (рисунок 5.25) рассчитайте параметры работ и событий.

Необходимо определить критический путь, рассчитать коэффициенты напряженности работ.

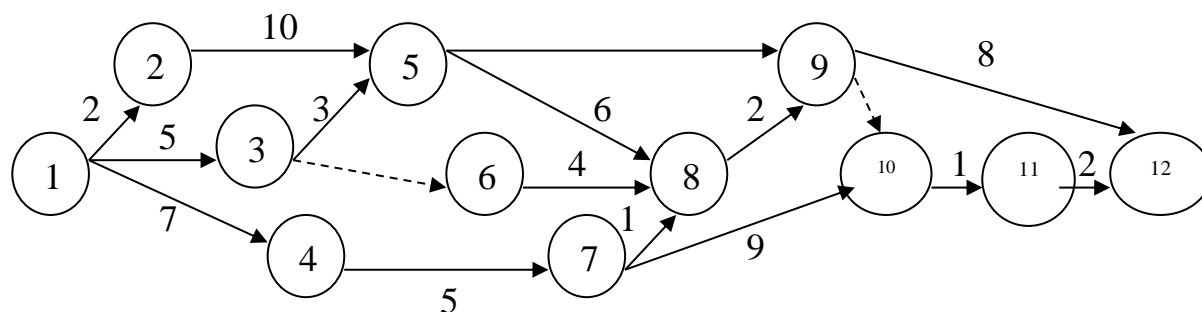


Рисунок 5.25 – Сетевой график к задаче 1

Задача 2 По представленному в предыдущем примере сетевому графику рассчитайте коэффициенты напряженности работ и выделите работы, по которым возможно проведение оптимизации.

Задача 3 Для заданного перечня работ (таблица 5.11) составить сетевую модель и провести её оптимизацию по количеству используемых трудовых ресурсов.

Таблица 5.11 – исходные данные

| Работа | Описание | Непосредственно предшествующие работы | Продолжительность суток | Потребность в рабочих |
|--------|---|---------------------------------------|-------------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| a | Начало | | 0 | - |
| b | Рытье котлована и заливка основания | a | 4 | 7 |
| c | Заливка бетонного фундамента | b | 2 | 6 |
| d | Сооружение деревянного каркаса, в том числе крыши | c | 4 | 5 |
| e | Выполнение кирпичной кладки | d | 6 | 10 |
| f | Укладка канализационных и водопроводных труб в подвальном помещении | c | 1 | 14 |
| g | Заливка пола подвального помещения | f | 2 | 3 |
| h | Установка водопроводных труб | f | 3 | 10 |
| i | Прокладка проводов | d | 2 | 4 |
| j | Установка отопления и вентиляции | d, g | 4 | 6 |
| k | Крепление штукатурных плит и штукатурные работы | i, j, h | 10 | 9 |
| l | Кладка покрытия пола | k | 3 | 9 |
| m | Установка кухонной арматуры | l | 1 | 3 |
| n | Завершение слесарно-водопроводных работ | l | 2 | 4 |
| o | Завершение плотницких работ | l | 3 | 5 |
| p | Кровельные работы и нанесение гидроизоляции | e | 2 | 12 |

Продолжение таблицы 5.11

| | | | | |
|---|---|------|---|----|
| q | Крепление водосточных желобов и водост. труб | p | 1 | 7 |
| r | Кладка коллектора ливневых вод | c | 1 | 13 |
| s | Циклевка и покрытие пола лаком | o, t | 2 | 2 |
| t | Покраска | m, n | 3 | 4 |
| u | Завершение установки электрооборудования | t | 1 | 2 |
| v | Земляные работы | q, r | 2 | 15 |
| w | Заливка пешеходных дорожек и благоустройство территории | v | 5 | 6 |
| x | Окончание | a | 0 | - |

Список использованных источников

1. Автоматизированный программный комплекс "Исследование и оптимизация сетевых графиков" ОГУ [Электронный ресурс] /А.В. Раменская // Университетский фонд электронных ресурсов (УФЭР) ФГБОУ ВО. Режим доступа: https://ufer.osu.ru/index.php?option=com_uferdbsearch&view=uferdbsearch&action=details&ufer_id=660
2. Айвазян, С.А. Прикладная статистика и основы эконометрики / С.А. Айвазян, В.С. Мхитарян. – М.: Издательское объединение «ЮНИТИ», 1998. – 1005 с.
3. Алексеева, Е.В. Численные методы оптимизации: Учеб. пособие / Е.В. Алексеева, О.А. Кутненко, А.В. Плясунов // Новосиб. ун-т. – Новосибирск, 2008. – 128 с.
4. Андреева, Е.А. Вариационное исчисление и методы оптимизации: учебное пособие / Е.А. Андреева, В.М. Цирулёва. – Оренбург-Тверь: ГОУ ОГУ, Твер. гос ун-т, 2004. - 575 с.
5. Аттетков, А.В. Методы оптимизации: учебное пособие / А.В. Аттетков, С.В. Галкин, В.С. Зарубин. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. – 440 с.
6. Базара, М. Нелинейное программирование. Теория и алгоритмы. / М. Базара, К. Шетти Пер. с англ. - М.: Мир, 1982. – 583 с.
7. Банди, Б. Методы оптимизации. Вводный курс / Б. Банди М.: Радио и связь, 1988. -128 с.
8. Беллман, Р. Динамическое программирование и уравнения в частных производных / Р. Беллман, Э. Энджел ; пер. с англ. С.П. Чеботарева, под ред. А.М. Летова. - М. : Мир, 1974. - 207 с.
9. Беллман, Р. Прикладные задачи динамического программирования / Р. Беллман, С. Дрейфус ; пер. с англ. Н.М. Митрофановой и [др.], под ред. А.А. Первозванского. - М. : Наука, 1965. - 458 с.

10. Вагнер, Г. Основы исследования операций. Том I. : пер. с англ. Б.Т. Вавилова. – М.: Издательство «Мир», 1972. – 337 с.
11. Васильев, Ф.П. Численные методы решения экстремальных задач / Ф.П. Васильев - М.: Наука, 1988. - 552 с.
12. Вержбицкий, В.М. Основы численных методов : учеб. для вузов / В. М. Вержбицкий . - М. : Высш. школа, 2002. - 840 с.
13. Галеев, Э.М. Оптимизация: Теория, примеры, задачи. / Э.М. Галеев, В.М. Тихомиров – М.: Эдиториал УРСС, 2000. – 320с.
14. Денисова, С.Т. Методы оптимальных решений: практикум / С.Т. Денисова, Р.М. Безбородникова, Т.А. Зеленина; Оренбургский государственный университет- Оренбург: ОГУ, 2015. – 190 с.
15. Домашова, Д.В. Методы решения задач многокритериальной оптимизации: метод. указания к лаб. практикуму и самостоят. работе студентов / Д. В. Домашова, Е. Н. Седова; М-во образования и науки Рос. Федерации, Федер. агентство по образованию, Гос. образоват. учреждение высш. проф. образования "Оренбург. гос. ун-т", Каф. мат. методов и моделей в экономике. - Электрон. текстовые дан. (1 файл: 0.49 Мб). - Оренбург : ГОУ ОГУ, 2008. - 49 с.
16. Домашова, Д.В. Управление предприятием: модели, методы и информационные технологии : [монография] / Д.В. Домашова, Е.М. Крипак, А.Г. Реннер; Федер. гос. бюджет. образоват. учреждение высш. проф. образования "Гос. ун-т М-ва финансов Рос. Федерации". - М. : Спецкнига, 2012. - 288 с
17. Домашова, Д.В. Численные методы решения задач нелинейного программирования: метод. указания к лаб. практикуму и самостоят. работе студентов / Д. В. Домашова, Е. Н. Седова; М-во образования и науки Рос. Федерации, Федер. агентство по образованию, Гос. образоват. учреждение высш. проф. образования "Оренбург. гос. ун-т", Каф. мат. методов и моделей в экономике. - Электрон. текстовые дан. (1 файл: 2.17 Мб). - Оренбург : ГОУ ОГУ, 2008. - 23 с.
18. Дэннис, Дж. (мл.). Численные методы безусловной оптимизации и решения нелинейных уравнений / Дж. Дэннис мл., Р. Шнабель; Перевод с англ. О. П. Бурдакова; Под ред. Ю. Г. Евтушенко. - М. : Мир, 1988. - 440 с.

19. Исследование операций в экономике : учеб. пособие для вузов / под ред. Н. Ш. Кремера. - М. : Юнити, 2004. - 407 с.
20. Исследование операций в экономике : учеб. пособие для вузов / Н.Ш. Кремер, Б.А. Путко, И.М. Трицин, М.Н. Фридман ; под ред. проф. Н.Ш. Кремера. – М.: ЮНИТИ, 2002. – 407 с.
21. Карманов, В.Г. Математическое программирование. - М.: Наука, 1975. – 272 с.
22. Костевич, Л.С. Исследование операций. Теория игр : учеб. пособие / Л.С. Костевич, А.А. Лапко. – 2-е изд., перераб. и доп. – Минск : Выш.шк., 2008. – 368 с.
23. Лотов, А.В. Многокритериальные задачи принятия решений: Учебное пособие / А.В. Лотов, И.И. Поспелова – М.: МАКС Пресс, 2008.– 197 с.
24. Майорова, Н.Л. Методы оптимизации : учебное пособие / Н.Л. Майорова, Д.В. Глазков; Яросл. гос. ун-т им П.Г.Демидова. – Ярославль : ЯрГУ , 2015. – 112с.
25. Методы и модели принятия решений в сфере управления персоналом: учебное пособие / Е.М. Крипак, Д.В. Домашова, Р.М. Безбородникова, Т.А. Зеленина; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2014. – 160 с.
26. Методы принятия оптимальных решений : учебное пособие для студентов, обучающихся по программам высшего образования по направлениям подготовки 01.03.04 Прикладная математика, 38.03.05 Бизнес-информатика, 38.04.01 Экономика / под ред. А. Г. Реннера; Р. М. Безбородникова [и др.] ; М-во образования и науки Рос. Федерации, Федер. гос. бюджет. образоват. учреждение высш. образования "Оренбург. гос. ун-т". - Оренбург : Университет, 2016 Ч. 1. - Оренбург : Университет. - 2016. – 245 с.
27. Ногин, В.Д. Принятие решений в многокритериальной среде. Количественный подход / В.Д. Ногин - М.: Физматлит, 2002. – 176 с.
28. Пантелеев, А.В. Методы оптимизации в примерах и задачах: учебное пособие / А. В. Пантелеев, Т.А. Летова. – М.: Высш. шк., 2002. – 544 с.
29. Парубек, Г.Э. Сетевое планирование и управление: методы построения,

расчетов и оптимизации сетевых графиков. – М.: Экономика, 1967. – 143 с.

30. Поддиновский, В.В. Парето-оптимальные решения многокритериальных задач. / В.В. Подиновский, В.Д. Ногин - М.: Наука; Главная редакция физико-математической литературы, 1982. – 256 с.

31. Попова, Т.М. Методы безусловной оптимизации : Тексты лекций / Т.М. Попова. – Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2013. – 76.

32. Ракитский, Ю.В. Численные методы решения жестких систем. / Ю.В. Ракитский, С.М. Устинов, И.Г. Черноруцкий - М.: Наука, 1979. – 208 с.

33. Раменская, А.В. Сетевые графики /О.А. Корнева, А.В. Раменская, В.В. Голунов // Сборник научных статей молодых ученых «Перспектива», № 12. - Оренбург: ГОУ ОГУ. – 2009. – С. 114-121.

34. Северин, В. П. Методы многомерной безусловной минимизации : учеб. пособие по курсу «Методы оптимизации» / В.П. Северин. – Харьков : НТУ «ХПИ», 2013. – 160 с.

35. Соловьев, В. И. Методы оптимальных решений: Учебное пособие. - М.: Финансовый университет, 2012. – 364 с

36. Таха, Хэмди, А. Введение в исследование операций : Пер. с англ. / Х.А. Таха. – 7-е изд. – М.: Издательский дом «Вильямс», 2005. – 912 с.

37. Химмельблау, Д. Прикладное нелинейное программирование. Пер. с англ. И.М. Быховской Б.Т. Вавилова, под ред. М.Л. Быховского, М.: Мир, 1975. – 534 с.

38. Черноруцкий, И.Г. Методы оптимизации в теории управления / И.Г. Черноруцкий. - СПб.: Питер, 2004. – 256 с.

39. Черноруцкий, И. Г. Методы принятия решений / И.Г. Черноруцкий. - СПб.: БХВ-Петербург, 2005. – 416 с.

40. Эддоус, М. Методы принятия решений / М. Эддоус, Р. Стэвсфилд ; пер. с англ. под ред. член-корр. РАН И.И. Елисеевой. – М.: Аудит, ЮНИТИ, 1997. –590 с.

41. Ярыгин, А.Н. Методы нахождения оптимального решения экономических задач многокритериальной оптимизации / А.Н. Ярыгин, Н.В. Колачева, С.Ш. Палфёрова// Вектор науки ТГУ. – 2013. - № 1(23). - С. 388-393.

Приложение А

(справочное)

Учет погрешности приближенных вычислений

При численном решении математических и прикладных задач неизбежно появление погрешности вычислений. Под погрешностью понимается величина, характеризующая точность результата.

Выделяют три основных вида погрешностей.

1) Погрешность задачи – «связана с приближенным характером исходной содержательной модели (в частности, с невозможностью учесть все факторы в процессе изучения моделируемого явления), а также ее математического описания, параметрами которого служат обычно приближенные числа (например, из-за принципиальной невозможности выполнения абсолютно точных измерений). Для вычислителя погрешность задачи следует считать неустранимой (безусловной), хотя постановщик задачи иногда может ее изменить»[12].

2) Погрешность метода – «связанна со способом решения поставленной математической задачи и появляющаяся в результате подмены исходной математической модели другой или конечной последовательностью других, например, линейных моделей. При создании численных методов закладывается возможность отслеживания таких погрешностей и доведения их до сколь угодно малого уровня. Отсюда естественно отношение к погрешности метода как к устранимой (или условной)» [12].

3) Погрешность действий (округлений) «обусловлена необходимостью выполнять арифметические операции над числами, усеченными до количества разрядов, зависящего от применяемой вычислительной техники (если, разумеется, не используются специальные программные средства, реализующие, например, арифметику рациональных чисел)» [12].

Эти погрешности в совокупности определяют полную погрешность результата решения задачи. Погрешность задачи характеризует точность, с которой следует

рассчитывать математическую модель. Погрешность, связанная с ошибками в исходной информации – это неустранимая погрешность. Причинами этих ошибок может быть, например, неточность измерений, невозможность представления данной величины конечной дробью. Нет смысла решать задачу существенно точнее, чем это диктуется неопределенностью исходных данных, следовательно погрешность метода подчиняют погрешности задачи. Так как при выводе оценок погрешностей численных методов обычно исходят из предположения, что все операции над числами выполняются точно, то погрешность округлений не должна существенно отражаться на результатах реализации методов, т.е. должна подчиняться погрешности метода. Поведение вычислительной погрешности зависит от правила округлений и алгоритма численного решения задачи. Влияние погрешностей округлений следует учитывать на всех стадиях решения вычислительных задач.

Рассмотрим подходы к оценке погрешностей. Обычно различают абсолютную и относительную погрешность.

Пусть x и a два "близких" числа. Условимся считать x – точным, a – приближенным.

ОПРЕДЕЛЕНИЕ Пусть x – некоторое число. Число a называется его приближенным значением, если a в определенном смысле мало отличается от x и заменяет x в вычислениях, $x \approx a$.

Величина $\Delta a = |x - a|$ называется абсолютной погрешностью приближенного числа a , а $\delta a = \frac{\Delta a}{|a|}$, $a \neq 0$ его относительной погрешностью. Числа $\Delta_a : \Delta_a \geq \Delta a$ и

$\delta_a : \delta_a = \frac{\Delta_a}{|a|} \geq \delta a$ называют оценками (или границами) абсолютной и относительной

погрешностей соответственно. Так как, как правило, истинные погрешности Δa и δa неизвестны, то их оценки Δ_a и δ_a (там где это не будет приводить к недоразумениям) будем называть просто абсолютной и относительной погрешностями соответственно.

Приложение Б

(справочное)

Элементы выпуклого анализа

Дополнительную информацию по приведенному разделу, включая доказательства приведенных теорем можно посмотреть, обратившись к источникам: 37.

Пусть R^n – конечномерное линейное пространство и x^1, x^2 – элементы этого пространства. Множество $x \subset R^n$ вида: $x = \{x \in R^n : x = \lambda x^1 + (1-\lambda)x^2\}, \lambda \in [0,1]$ будем называть отрезком с концами $x^1, x^2 : [x^1, x^2]$.

ОПРЕДЕЛЕНИЕ Подмножество X линейного пространства R^n называют выпуклым множеством, если оно вместе с любыми двумя точками $x^1, x^2 \in X$ содержит весь отрезок $[x^1, x^2]$, т.е для любых $x^1, x^2 \in X$ и $\lambda \in [0,1]$ выполняется соотношение $\lambda x^1 + (1-\lambda)x^2 \in X$.

Пустое множество считаем выпуклым по определению.

ОПРЕДЕЛЕНИЕ Множество X называется аффинным, если оно вместе с любыми своими элементами x^1, x^2 содержит и любую их линейную комбинацию вида: $\lambda x^1 + (1-\lambda)x^2 \in X, \lambda \in R^n$.

Аффинное множество вместе со своими любыми двумя точками содержит и прямую, проходящую через эти две точки. Аффинное множество – частный случай выпуклого множества.

ОПРЕДЕЛЕНИЕ Множество $x \subset R^n$ называется конусом, если $\forall x \in X, \lambda x \in X$ при $\lambda \in R : \lambda \geq 0$.

Пусть $x^1, x^2 \dots x^k$ – произвольные элементы пространства R^n , тогда

1) их линейную комбинацию $\sum_{i=1}^k \lambda_i x^i$ назовем выпуклой комбинацией, если

$$\sum_{i=1}^k \lambda_i = 1, \lambda_i \geq 0, i = 1, \dots, k;$$

2) Их линейную комбинацию $\sum_{i=1}^k \lambda_i x^i$ назовем аффинной комбинацией,

если $\sum_{i=1}^k \lambda_i = 1, \lambda_i \in \mathbb{R}, i = 1, \dots, k;$

3) Их линейную комбинацию $\sum_{i=1}^k \lambda_i x^i$ называют неотрицательной, если

$$\lambda_i \geq 0, i = 1, \dots, k.$$

ТЕОРЕМА Б.1

Для того, чтобы множество $X \subset \mathbb{R}^n$ было выпуклым, необходимо и достаточно, чтобы любая выпуклая комбинация принадлежала этому множеству X .

Выпуклое множество содержит всевозможные выпуклые комбинации своих точек.

Выпуклый конус содержит все неотрицательные комбинации своих точек.

Аффинное множество содержит все аффинные комбинации своих точек.

ОПРЕДЕЛЕНИЕ Выпуклая комбинация $\sum_{i=1}^k \lambda_i x^i$ элементов линейного

пространства называют строго выпуклой комбинацией, если все $\lambda_i > 0, i = 1, \dots, k$.

ОПРЕДЕЛЕНИЕ Элемент $x \in X$ называют крайней точкой множества $X \subset \mathbb{R}^n$ или его вершиной, если он не может быть представлен в виде строго выпуклой комбинации двух различных элементов из X .

Иначе говоря, если $x = \lambda x^1 + (1 - \lambda)x^2, x^1, x^2 \in X, \lambda \in (0, 1)$, то $x^1 = x^2 = x$.

Если выпуклое множество не имеет крайних точек, то его называют строго выпуклым.

ОПРЕДЕЛЕНИЕ Выпуклой оболочкой произвольного подмножества $X \subset \mathbb{R}^n$ называют пересечение всех выпуклых множеств в \mathbb{R}^n , которые содержат в себе множество X .

ОПРЕДЕЛЕНИЕ Пересечение всех выпуклых конусов, содержащих множество X , называют конической оболочкой.

ОПРЕДЕЛЕНИЕ Пересечение всех аффинных множеств, содержащих множество X , называют аффинной оболочкой.

ТЕОРЕМА Б.2 Выпуклая (коническая, аффинная) оболочка множества X совпадает с множеством всех выпуклых (конических, аффинных) комбинаций элементов множества X .

ТЕОРЕМА Б.3 (Теорема Каратеодори)

Пусть $X \subset \mathbb{R}^n$, тогда любую точку его выпуклой оболочки можно представить в виде выпуклой комбинации не более чем $n+1$ точки из множества X .

ОПРЕДЕЛЕНИЕ Функцию $f(x)$, где $f \in \mathbb{R}, x \in \mathbb{R}^n$ называют выпуклой функцией, если $\forall x^1, x^2 \in X, \forall \lambda \in [0,1]$ выполнено неравенство $f(\lambda x^1 + (1-\lambda)x^2) \leq \lambda f(x^1) + (1-\lambda)f(x^2)$.

ОПРЕДЕЛЕНИЕ Функцию $f(x)$, где $f \in \mathbb{R}, x \in \mathbb{R}^n$ называют строго выпуклой функцией, если $\forall x^1, x^2 \in X, \forall \lambda \in (0,1), x^1 \neq x^2$ выполнено неравенство: $f(\lambda x^1 + (1-\lambda)x^2) < \lambda f(x^1) + (1-\lambda)f(x^2)$.

ОПРЕДЕЛЕНИЕ Функция $f(x)$ определенная на некотором множестве X называют сильно выпуклой с коэффициентом $k>0$, если выполняются условия:

$$f(\lambda x^1 + (1-\lambda)x^2) \leq \lambda f(x^1) + (1-\lambda)f(x^2) - \frac{k}{2} \lambda(1-\lambda) \|x^1 - x^2\|^2, \forall x^1, x^2 \in X, \forall \lambda \in [0,1]$$

ЗАМЕЧАНИЕ Аналогично определяются понятия вогнутой функции. Функция $f(x)$ является вогнутой (строго вогнутой, строго вогнутой с коэффициентом $k>0$) если $-f(x)$ выпуклая (строго выпуклая, строго выпуклая с коэффициентом $k>0$).

Свойства выпуклых функций

ТЕОРЕМА Б.3 Если функции $f_1(x), \dots, f_n(x)$ – выпуклы на выпуклом множестве

X , то $\sum_{i=1}^n \lambda_i f_i(x), \lambda_i \geq 0$ - так же выпуклая функция.

ТЕОРЕМА Б.4 Если функция $f(x)$ – определенная на выпуклом множестве X выпукла, тогда $\forall x^i \in X, \lambda_i \geq 0, i = 1, \dots, k$: $\sum_{i=1}^k \lambda_i = 1$ выполнялось неравенство

Иенсена:

$$f\left(\sum_{i=1}^k \lambda_i x^i\right) \leq \sum_{i=1}^k \lambda_i f(x^i).$$

ЗАМЕЧАНИЕ. Из неравенства Иенсена можно вывести, что среднее арифметическое больше среднего геометрического.

ТЕОРЕМА Б.4 (Необходимое и достаточное условие выпуклости функции)

Для того, чтобы функция $f(x)$, определенная на выпуклом множестве $X \subset \mathbb{R}^n$, была выпуклой (строго выпуклой) необходимо и достаточно, чтобы для любых попарно-различных элементов $x^i \in X, i = 1, \dots, k, \forall \lambda_i \geq 0 (\lambda_i > 0), i = 1, \dots, k$ таких что $\sum_{i=1}^k \lambda_i = 1$ выполнялось неравенство Иенсена: $f\left(\sum_{i=1}^k \lambda_i x^i\right) \leq \sum_{i=1}^k \lambda_i f(x^i)$ (строгое

неравенство Иенсена: $f\left(\sum_{i=1}^k \lambda_i x^i\right) < \sum_{i=1}^k \lambda_i f(x^i)$).

ТЕОРЕМА Б.5 Если $g(x)$ – выпуклая функция на выпуклом множестве $X \subset \mathbb{R}^n$, а $h(t)$ – выпуклая неубывающая функция одного действительного переменного, определенная на множестве $g(X)$, то сложная функция $f(x) = h(g(x))$ является выпуклой на множестве X . Если к тому же $g(x)$ – строго выпукла, а $h(t)$ возрастает на множестве \mathbb{R} , то $f(x)$ – строго выпукла на множестве X .

ТЕОРЕМА Б.6 Если $g_i(x), i = 1, \dots, m$ – выпуклые функции на выпуклом множестве $X \subset \mathbb{R}^n$, а функция $h(y)$ – скалярная функция многих переменных, выпуклая на \mathbb{R}^m и не убывающая по каждому своему аргументу, то сложная функция $\varphi(x) = h(g_1(x), g_2(x), \dots, g_m(x))$ является выпуклой на множестве X . Если к тому же, хотя бы одна из функций $g_i(x)$ строго выпукла на множестве $X \subset \mathbb{R}^n$, а $h(y)$ – возрастает по каждому своему аргументу, то $\varphi(x)$ – строго выпукла на X .

ОПРЕДЕЛЕНИЕ Пусть $f(x)$ – произвольная функция, определенная на

выпуклом множестве X , функцию $\phi(t) = f(tx^1 + (1-t)x^2)$ одного переменного, заданного на промежутке $S(x^1, x^2)$, $x^1, x^2 \in X$, x – выпуклая, при $t \in S: (tx^1 + (1-t)x^2) \in X$ будем называть сечением функции $f(x)$

ТЕОРЕМА Б.7

Для того, чтобы $f(x)$, определенная на выпуклом множестве $X \subset \mathbb{R}^n$ была выпуклой(строго выпуклой), необходимо и достаточно, чтобы любое сечение этой функции было выпуклой вниз (строго выпуклой) функцией.

Критерии выпуклости для дифференцируемых функции

Дифференцируемость функции позволяет сформулировать простые признаки, с помощью которых можно выяснить, является ли она выпуклой.

Так например, для функции одной переменной $f(x)$ дифференцируема на промежутке $[a, b]$, $f(x)$ является выпуклой(строго выпуклой) на интервале (a, b) , тогда и только тогда, когда $f'(x)$ на (a, b) не убывает(возрастает).

Отсюда с помощью формулы конечных приращений легко заключить, что для выпуклой функции $f(x)$ на любых $x^1, x^2: a < x^1 < x^2 < b$ выполняется неравенство:

$$f'(x^1)(x^2 - x^1) \leq f(x^2) - f(x^1) \leq f'(x^2)(x^2 - x^1). \quad (\text{Б.1})$$

Неравенство (Б.1) является не только необходимым, но и достаточным. Критерий (Б.1) можно обобщить на случай многих переменных.

ТЕОРЕМА Б.8 Необходимые и достаточные условия выпуклости дифференцируемой функции

Пусть скалярная функция $f(x)$ дифференцируема на множестве $X \subset \mathbb{R}^n$, тогда для выпуклости $f(x)$ на множестве X , необходимо и достаточно, чтобы $\forall x^1, x^2 \in X$ выполнялось неравенство:

$$\langle \text{grad} f(x^2), h \rangle \leq f(x^1) - f(x^2) \leq \langle \text{grad} f(x^1), h \rangle, h = x^1 - x^2. \quad (\text{Б.2})$$

ЗАМЕЧАНИЕ Для строгой выпуклости $f(x)$ непрерывной дифференцируемой на множестве $X \subset R^n$ необходимо и достаточно, чтобы $\forall x^1, x^2 \in X$ выполнялось строгое неравенство:

$$\langle \text{grad}f(x^2), h \rangle < f(x^1) - f(x^2) < \langle \text{grad}f(x^1), h \rangle, h = x^1 - x^2. \quad (\text{Б.3})$$

Доказательство:

1) Докажем необходимость. Пусть $f(x)$ выпукла на множестве X , рассмотрим сечение $\phi(t) = f(tx^1 + (1-t)x^2)$, заданное произвольными точками $x^1, x^2 \in X$. Функция $\phi(t)$ определена и дифференцируема на отрезке $[0,1]$. Согласно теореме о сечении выпуклой функции, функция $\phi(t)$ – выпукла на отрезке $[0,1]$, а следовательно справедливо:

$$\phi'(0) \leq \phi(1) - \phi(0) \leq \phi'(1). \quad (\text{Б.4})$$

В силу правила дифференцирования сложной функции выполняется:

$$\phi'(t) = \langle \text{grad}f(x), h \rangle, x = tx^1 + (1-t)x^2, \quad (\text{Б.5})$$

поэтому

$$\phi'(0) = \langle \text{grad}f(x^2), h \rangle; \quad \phi'(1) = \langle \text{grad}f(x^1), h \rangle. \quad (\text{Б.6})$$

Подставляя (Б.6) в (Б.4) получаем (Б.2).

2) Докажем достаточность. Пусть $\forall x^1, x^2 \in X$ выполняется условие (Б.2). Возьмем произвольные $y^1, y^2 \in X$ и рассмотрим сечение $\phi(t) = f(ty^1 + (1-t)y^2)$. Выберем произвольно t_1 и t_2 из области определения $\phi(t)$, тогда у нас есть две точки: $x^i = t_i y^1 + (1-t_i)y^2, i = 1, 2 \in X$ и для них выполняется неравенство (Б.2). Из равенств

(Б.5) при $t=t_1=t_2$, получаем: $\varphi'(t_i) = \langle \text{grad}f(x^i), y^1 - y^2 \rangle, i = 1, 2$. Так как $h = x^1 - x^2 = (t_2 - t_1)(y^1 - y^2)$, то:

$$\langle \text{grad}f(x^i), h \rangle = \langle \text{grad}f(x^i), y^1 - y^2 \rangle (t_2 - t_1) = \varphi'(t_i)(t_2 - t_1), i = 1, 2. \quad (\text{Б.7})$$

Таким образом неравенство (Б.2) равносильно неравенству (Б.7), так как t_1 и t_2 выбирались произвольно, следовательно $\varphi(t)$ является выпуклой, а следовательно в силу теоремы Б.7 о сечении выпуклой функции, $f(x)$ выпукла на множестве X . Аналогично доказывается строгая выпуклость.

ч.т.д.

ТЕОРЕМА Б.9 Для того, чтобы дважды дифференцируемая на открытом выпуклом множестве $X \subset \mathbb{R}^n$ функция $f(x)$ была выпуклой, необходимо и достаточно, чтобы ее матрица Гессе $H(x)$ была неотрицательно определена в любой точке x из множества X . Для строгой выпуклости функции $f(x)$ необходимо и достаточно, чтобы матрица Гессе $H(x)$ была положительно определена.

Рассмотрим понятие производной по направлению.

Теорема Б.10 Если $f(x)$ – выпуклая функция, тогда для любого $\forall x \in X \subset \mathbb{R}^n$ и для любого единичного вектора $\bar{l} = (l_1, l_2, \dots, l_n)$ существует производная функции $f(x_1, x_2, \dots, x_n)$ в точке $x = (x_1, x_2, \dots, x_n)$ по направлению \bar{l} :

$$f'_{\bar{l}}(x) = \lim_{t \rightarrow +0} \frac{f(x + t\bar{l}) - f(x)}{t}. \quad (\text{Б.8})$$

Значение этого выражения показывает, как быстро меняется значение функции при сдвиге аргумента в направлении вектора \bar{l} .

Производная по направлению \bar{l} равна скалярному произведению градиента функции $f(x_1, x_2, \dots, x_n)$ на единичный вектор \bar{l} и определяется следующим образом:

$$f'_1(x) = \frac{\partial f(x)}{\partial \bar{l}} = \langle \text{grad} f(x), \bar{l} \rangle = \frac{\partial f(x)}{\partial x_1} l_1 + \frac{\partial f(x)}{\partial x_2} l_2 + \dots + \frac{\partial f(x)}{\partial x_n} l_n. \quad (\text{Б.9})$$

К примеру, для функции трех аргументов $f(x_1, x_2, x_3)$:

$$\frac{\partial f(x)}{\partial l} = \frac{\partial f(x)}{\partial x_1} \cos \alpha + \frac{\partial f(x)}{\partial x_2} \cos \beta + \frac{\partial f(x)}{\partial x_3} \cos \gamma.$$

ТЕОРЕМА Б.11 Непрерывно-дифференцируемая функция $f(x)$ выпуклая тогда и только тогда, когда выполняются условия:

- 1) $f(x) - f(x^*) \geq \langle \text{grad} f(x^*), x - x^* \rangle$
- 2) $\langle \text{grad} f(x + t\bar{l}), \bar{l} \rangle$ - неубывающая функция аргумента $t \forall x \in X$.

СЛЕДСТВИЕ Выпуклая функция непрерывна в каждой внутренней точке X .

УТВЕРЖДЕНИЕ Б.1 Если $f(x)$ – непрерывно-дифференцируемая функция, то ее производная по направлению является линейной функцией этого направления.

Приведем понятие обобщенного градиента функции (субградиента).

ОПРЕДЕЛЕНИЕ Вектор $\hat{f}_x(x) \in \mathbb{R}^n$ называют обобщенным градиентом (субградиентом) функции $f(x)$ в точке x если выполняется неравенство:

$$f(y) - f(x) \geq \langle \hat{f}_x(x), y - x \rangle, \quad \forall y \in X. \quad (\text{Б.10})$$

Множество обобщенных градиентов (обозначим $M(x)$) непрерывной выпуклой функции в точке x является выпуклым, замкнутым и ограниченным.

УТВЕРЖДЕНИЕ Б.2 Для выпуклой функции $f(x)$ производная по направлению \bar{l} равна:

$$f'_1(x) = \max_{y \in M(x)} \langle l, y \rangle. \quad (\text{Б.11})$$

СЛЕДСТВИЕ Направление $\bar{l}(x)$ наискорейшего убывания функции в точке x равно:

$$\bar{l}(x) \stackrel{\Delta}{=} \underset{\|\bar{l}\|=1}{\operatorname{argmin}} f'_1(x) = \underset{\|\bar{l}\|=1}{\operatorname{argmin}} (\max \langle \bar{l}, y \rangle), y \in M(x). \quad (\text{Б.12})$$

Приложение В

(справочное)

Квадратичная форма

Квадратичной формой относительно переменных $x_1 \dots x_m$ называется функция вида:

$$C(x, x) = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_i x_j, \quad (\Gamma.1)$$

или в матричном виде:

$$C(x, y) = x^T C x, \quad (\Gamma.2)$$

где $C = \{c_{ij}\}, i = 1, \dots, m, j = 1, \dots, m$ – матрица коэффициентов квадратичной формы, причем $c_{ij} = c_{ji}$.

Билинейной формой называется функция вида:

$$C(x, y) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_i y_j. \quad (\Gamma.3)$$

Если матрица квадратичной формы имеет полный ранг, то квадратичную форму называют невырожденной, иначе вырожденной.

Квадратичная форма $C(x, x)$ называется положительно определенной, если для любого $x \neq 0$ $C(x, x) > 0$, отрицательно определенной $\forall x \neq 0$ $C(x, x) < 0$. Положительные и отрицательные формы называются знакоопределенными.

ТЕОРЕМА В.1

Квадратичная форма $C(x,x)$ является положительно определенной, если все угловые миноры соответствующей матрицы строго положительные.

Для определения типа квадратичной формы можно воспользоваться критерием Сильвестра.

Квадратичная форма называется знакопеременной, если она принимает как положительные, так и отрицательные значения. Квадратичная форма называется неположительно (неотрицательно) определенной, если выполняется условие $C(x,x) \geq 0$ ($C(x,x) \leq 0$) и $\exists x \neq 0$, такой, что $C(x,x) = 0$. Квадратичная форма называется квазизнакоопределенной если $C(x,x) \geq 0$ ($C(x,x) \leq 0$), но форма не является знакоопределенной.

Приложение Г
(справочное)
Теоремы отделимости

Дополнительную информацию по приведенному разделу, включая доказательства приведенных теорем можно посмотреть, обратившись к источникам: б.

При выводе необходимого условия экстремума функции Лагранжа в задачах выпуклого нелинейного программирования используется свойство отделимости непересекающихся выпуклых функций.

Рассмотрим понятие гиперплоскости. Гиперплоскость – подпространство Евклидова пространства коразмерности 1, т.е. размерности на 1 меньше, чем искомое пространство. В частности, термин «гиперплоскость» в трех-мерном пространстве означает плоскость, в двумерном - прямую.

Уравнение гиперплоскости в пространстве E^n задается в виде:

$$\Pi(c,1) = \{x \in E^n : \langle c, x \rangle = 1\}, \quad (\text{Г.1})$$

где $c \in E^n$ – ненулевой вектор называемый «нормалью» в гиперплоскости;

$1 \in \mathbb{R}$ – элемент пространства \mathbb{R}^1 .

Например, пусть $c \in E^n$ нормальный вектор к гиперплоскости, тогда уравнение гиперплоскости, проходящей через точку x^0 имеет вид:

$$d = \langle c, x - x^0 \rangle = c_1(x_1 - x_1^0) + c_2(x_2 - x_2^0) + \dots + c_n(x_n - x_n^0).$$

В пространстве E^n гиперплоскость определяет два полупространства:

$$\Pi_{-}(c, l) = \{x \in E^n : \langle c, x \rangle < l\}, \quad (\Gamma.2)$$

$$\Pi_{+}(c, l) = \{x \in E^n : \langle c, x \rangle \geq l\}. \quad (\Gamma.3)$$

ТЕОРЕМА Г.1 Пусть X – выпуклое множество и $y \notin X$, тогда точка y может быть строго отделена от множества X некоторой гиперплоскостью, т.е. $\exists c \neq 0: \langle c, x \rangle \leq \langle c, y \rangle - \varepsilon, \varepsilon > 0, \forall x \in X$.

ТЕОРЕМА Г.2 [6] Пусть X – не пустое выпуклое множество и y принадлежит границе множества X , тогда существует гиперплоскость опорная к X в точке y , т.е. $\exists c \neq 0: \Pi(c, l) = \{x \in E^n : \langle c, x \rangle = l\}$ и $\langle c, y \rangle = l \forall x \in X$.

ТЕОРЕМА Г.3 [6] Пусть X и Y – выпуклые множества и они не пересекаются, тогда их можно отделить друг от друга, т.е. существует гиперплоскость $\Pi(c, l) = \{x \in E^n : \langle c, x \rangle = l\}$: такая что $\langle c, x \rangle \leq l, \forall x \in X, \langle c, y \rangle \geq l, y \in Y$. Такая гиперплоскость является разделяющей.

На рисунке Г.1 приведены опорная и разделяющая гиперплоскости.

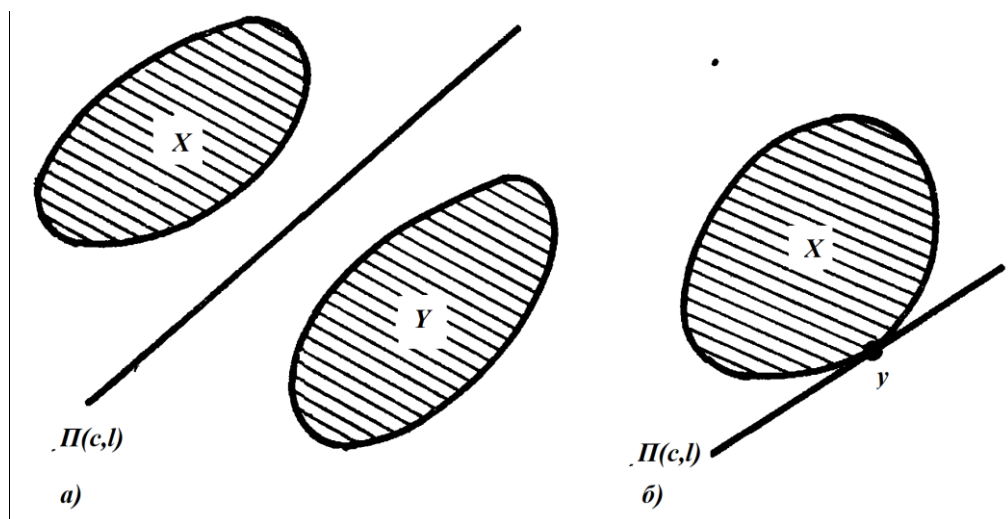


Рисунок Г.1 – Опорная и разделяющая гиперплоскости: а) разделяющая, б) опорная