

Университетский учебник

*Ю.М.Нефёдов*

# МЕТОДЫ ОПТИМИЗАЦИИ

Прикладная математика  
и информатика

Луганск  
2019

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
ЛУГАНСКОЙ НАРОДНОЙ РЕСПУБЛИКИ  
ГОУ ВПО « Луганский национальный университет  
имени Владимира Даля »**

*Ю. М. Нефёдов*

## **МЕТОДЫ ОПТИМИЗАЦИИ**

*Учебное пособие*

Второе издание, исправленное и дополненное

ЛУГАНСК 2019

УДК 519.6  
ББК 22.18

*Рекомендовано к изданию Ученым советом  
ГОУ ВПО «Луганский национальный университет  
имени Владимира Даля»*

**Нефёдов Ю. М.**

Методы оптимизации: Учебное пособие. – Луганск: ЛНУ имени В. Даля, 2019. – 245 с.

***Рецензенты:***

**Монсеенко И. А.**, доктор физико-математических наук, доцент, профессор кафедры теории упругости и вычислительной математики ГОУ ВПО «Донецкий национальный университет»

**Темникова С.В.**, кандидат технических наук, и.о. заведующего кафедрой фундаментальной математики ГОУ ЛПУ «Луганский национальный университет имени Тараса Шевченко».

В учебном пособии изложены основы теории оптимизации и алгоритмы решения задач. Приведены наглядные иллюстрации и конкретные примеры. Последовательно рассмотрены методы линейного программирования, дискретной оптимизации, безусловной и условной оптимизации в евклидовом пространстве. Большое внимание уделено построению численных методов решения задач оптимизации и их алгоритмической реализации.

Учебное пособие предназначено для студентов направления «Прикладная математика и информатика».

УДК 519.6  
ББК 22.18

© Нефёдов Ю.М. , 2019

© ГОУ ВПО «Луганский Национальный университет имени Владимира Даля»,

## СОДЕРЖАНИЕ

Предисловие.....	5
1. ВВЕДЕНИЕ В ОПТИМИЗАЦИЮ .....	6
1.1. Постановка задачи оптимизации.....	6
1.2. Классификация задач оптимизации.....	10
1.3. Общие свойства задач оптимизации.....	12
1.4. Пример задачи оптимизации.....	15
2. МЕТОДЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ .....	20
2.1. Постановка задачи линейного программирования.....	20
2.2. Теоретические основы линейного программирования.....	25
2.3. Симплекс-метод .....	38
2.4. Элементы теории двойственности.....	52
2.5. Двойственный симплекс-метод.....	59
2.7. Специальные задачи линейного программирования.....	62
3. МЕТОДЫ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ.....	82
3.1. Примеры дискретных оптимизационных задач.....	82
3.2. Метод отсекающих плоскостей (метод Гомори).....	85
3.3. Метод ветвей и границ.....	92
3.4. Булева задача линейного программирования.....	99
3.5. Метод динамического программирования .....	107
4. МЕТОДЫ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ.....	120
4.1. Предварительные сведения.....	120
4.2. Прямые методы минимизации.....	125
4.3. Методы, использующие производные функции.....	135
4.4. Методы минимизации многомодальных функций.....	139
5. МЕТОДЫ БЕЗУСЛОВНОЙ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ .....	146
5.1. Предварительные сведения.....	152
5.2. Выпуклые множества и выпуклые функции.....	159
5.3. Общие принципы многомерной оптимизации.....	159
5.4. Прямые методы безусловной оптимизации.....	163
5.5. Методы оптимизации, использующие производные функций.....	177
6. ЭЛЕМЕНТЫ ТЕОРИИ ВЫПУКЛОГО ПРОГРАММИРОВАНИЯ.....	189
6.1. Постановка задачи математического программирования.....	189
6.2. Необходимые условия минимума в терминах направлений..	192
6.3. Метод множителей Лагранжа.....	196
6.4. Условия оптимальности в выпуклом программировании....	203

6.5. Задача математического программирования со смешанными ограничениями .....	210
<b>1. МЕТОДЫ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ .....</b>	<b>213</b>
7.1. Задача квадратичного программирования .....	213
7.2. Методы возможных направлений.....	215
7.3. Градиентные методы.....	230
7.4. Методы последовательной безусловной минимизации.....	238
ЛИТЕРАТУРА.....	244

## ПРЕДИСЛОВИЕ

В настоящее время требования к математической подготовке инженера достаточно высоки. В частности, он должен уметь грамотно переводить на математический язык технические, экономические, естественнонаучные и другие прикладные задачи, анализировать зависимость их решений от условий, режимов, параметров реальных процессов и выбирать наилучшие варианты, т.е. обладать навыками математического моделирования и оптимизации реальных объектов. Поэтому курсу методов оптимизации отводится значительная роль в математической подготовке студентов вузов.

Предлагаемое учебное пособие написано на основе курса лекций «Методы оптимизации», которые в течение ряда лет читались автором в Восточноукраинском национальном университете для студентов специальностей «Прикладная математика» и «Информатика».

В первой главе описан общий подход к математическому моделированию задач оптимизации.

Вторая глава содержит теоретические основы и методы линейного программирования. Рассмотрены теоретические и вычислительные аспекты симплекс–метода, а также – его модификация на основе теории двойственности. Разобраны подходы к решению линейных задач специального типа.

Третья глава посвящена основным методам решения дискретных задач оптимизации. Рассмотрены алгоритмы решения целочисленных и булевых задач оптимизации. А также применение метода динамического программирования к решению дискретных задач.

В четвертой главе рассмотрены методы оптимизации функций одной переменной, которые широко используются в алгоритмах методов минимизации многомерных функций.

Пятая глава содержит материал об общих принципах многомерной оптимизации и безусловной минимизации функций нескольких переменных. В ней описаны некоторые наиболее эффективные и важные с методологической точки зрения алгоритмы минимизации.

В шестой главе рассмотрены задачи условной оптимизации, метод множителей Лагранжа, необходимые и достаточные условия Куна–Таккера для задач выпуклого программирования.

В седьмой главе описаны численные методы условной оптимизации.

Для усвоения материала книги достаточно знаний в пределах курсов математического анализа и линейной алгебры университета.

# Глава 1

## ВВЕДЕНИЕ В ОПТИМИЗАЦИЮ

Дано определение задачи оптимизации. Постановка задачи оптимизации рассмотрена с характеристикой основных этапов построения математической модели. Определена классификация задач оптимизации по виду целевой функции и множества допустимых решений.

Рассмотрены основные понятия, связанные с оптимизацией, и общие свойства решений задач оптимизации.

### 1.1. Постановка задачи оптимизации

*Оптимизация* – это процесс выбора наилучшего решения рассматриваемой задачи в смысле экстремального значения некоторого критерия (целевой функции).

Теория оптимизации включает в себя фундаментальные результаты и численные методы, позволяющие находить наилучший вариант из множества возможных альтернатив без их полного перебора и сравнения.

Для того чтобы использовать результаты и вычислительные процедуры теории оптимизации на практике, необходимо прежде всего построить *математическую модель* объекта оптимизации. Математическая модель – это более или менее полное математическое описание исследуемого процесса или явления.

В большинстве реальных ситуаций дать исчерпывающее математическое представление оптимизируемой системы с учетом всех взаимосвязей ее частей, взаимодействий с внешним миром, всех целей ее функционирования бывает затруднительно или невозможно. Поэтому при построении математической модели необходимо, как правило, выделять и учитывать в дальнейшем только наиболее важные, существенные стороны исследуемого объекта с тем, чтобы было возможным его математическое описание, а также последующее решение поставленной математической задачи. При этом неучтенные в математической модели факторы не должны существенно влиять на окончательный результат оптимизации. Таким образом, математическое моделирование является сложной и ответственной творческой задачей, требующей от исследователя глубоких знаний в соответствующей области, практического опыта, интуиции и критического анализа получаемых результатов.

Несмотря на то что общего рецепта построения математических моделей оптимизации не существует, можно условно разбить процесс математического моделирования на следующие основные этапы.

*Определение границ системы оптимизации.* Необходимость этого этапа диктуется невозможностью учета и исчерпывающего описания всех сторон большинства реальных систем. Выделив главные переменные, параметры и ограничения, следует приближенно представить систему как некоторую изолированную часть реального мира и упростить ее внутреннюю структуру.

Например, при оптимизации работы одного из цехов предприятия в некоторых случаях можно пренебречь влиянием особенностей функционирования других цехов, систем снабжения и сбыта всего предприятия, его взаимодействием с другими организациями, конъюнктурой рынка и многими другими факторами. Тогда цех будет рассматриваться как изолированная система, а его связи с внешним миром либо считаются зафиксированными, либо вовсе не учитываются.

Может оказаться, что первоначальные границы системы оптимизации выбраны неудачно. Это становится ясным при дальнейшем анализе системы и ее математической модели, при интерпретации результатов поиска оптимального решения, сопоставлении их с практикой и т.д. В одних случаях границы системы следует расширить, а в других – сузить. Например, если выясняется, что влиянием на работу исследуемого цеха других подразделений предприятия нельзя игнорировать, то необходимо включить в систему и эти подразделения. С другой стороны, может оказаться, что сам цех состоит из нескольких в большей степени независимо работающих участков, которые без значительного упрощения реальной ситуации можно рассматривать изолированно. Тогда для облегчения поиска оптимального решения разумно исследовать каждый участок как отдельную систему.

Вообще, в инженерной практике следует, насколько возможно, стремиться упрощать системы, подлежащие оптимизации, разбивать сложные системы на более простые подсистемы, если есть уверенность, что это повлияет на окончательный результат в допустимых пределах.

*Выбор управляемых переменных.* На этом этапе математического моделирования необходимо провести различие между теми величинами, значения которых можно изменять и выбирать с целью достижения наилучшего результата (*управляемыми переменными*), и величинами, которые фиксированы или определяются внешними факторами.

Определение тех значений управляемых переменных, которым соответствует наилучшая (*оптимальная*) ситуация, и представляет собой задачу оптимизации.

Одни и те же величины, в зависимости от выбранных границ оптимизируемой системы и уровня детализации ее описания, могут оказаться либо управляемыми переменными, либо нет. Например, в упомянутой ситуации с оптимизацией работы цеха объем поставок какого-либо сырья из другого цеха в одних случаях следует считать фиксированным и не зависящим от нашего выбора, а в других случаях – управляемым.

*Определение ограничений на управляемые переменные.* В реальных условиях на выбор значений управляемых переменных, как правило, наложены ограничения, связанные с ограниченностью имеющихся ресурсов, мощностей и других возможностей. При построении математической модели эти ограничения обычно записывают в виде равенств и неравенств или указывают множества, которым должны принадлежать значения управляемых переменных. Совокупность всех ограничений на управляемые переменные определяет так называемое *допустимое множество* задачи оптимизации.

Например, если годовой объем выпускаемой цехом продукции данного вида является управляемой переменной, то ее значения, во-первых, не могут быть отрицательными и, во-вторых, ограничены сверху максимальной производительностью оборудования цеха.

*Выбор числового критерия оптимизации.* Обязательной составной частью математической модели объекта оптимизации является числовой критерий, минимальному или максимальному значению которого (в зависимости от конкретной задачи) должен соответствовать наилучший вариант поведения исследуемого объекта. Величина этого критерия полностью определяется выбранными значениями управляемых переменных, т.е. он является функцией этих переменных и называется *целевой функцией*.

В инженерной практике используются различные критерии оптимизации. Например, это могут быть критерии экономического характера, такие как себестоимость, прибыль, капитальные затраты и т.д., технические или физические параметры системы – продолжительность технологического процесса, потребляемая энергия, максимальная механическая нагрузка и другие.

Следует отметить, что во многих случаях выбор критерия оптимизации не является очевидным и однозначным. Часто бывает трудно поставить в соответствие всей совокупности целей функционирования системы какой-либо один критерий. Это объясняется различными причинами, такими как сложность целевой функции, описывающей большую совокупность разнородных целей, неопределенность формулировок некоторых целей, препятствующая описанию их с помощью количественных характеристик, наличие противоречивых целей и т.д. Например, невозможно найти решение, обеспечивающее одновремен-

но минимальные затраты, максимальную надежность, минимальное энергопотребление и максимальное быстродействие.

Выход из этого положения определяется в каждом конкретном случае. Например, из многих критериев, характеризующих различные цели оптимизации, выбирают один, считая его основным, а остальные – второстепенными. Далее второстепенные критерии либо не учитываются, либо учитываются частично с помощью дополнительных ограничений на управляемые переменные. Эти ограничения обеспечивают изменение второстепенных критериев в заданных диапазонах приемлемых значений.

Другой путь состоит в формулировке комплексного критерия, т.е. целевой функции, включающей с разумно выбранными *весовыми коэффициентами* целевые функции, соответствующие различным целям.

Объединяя результаты рассмотренных этапов построения математической модели, ее записывают в виде математической задачи оптимизации, включающей построенную целевую функцию и найденные ограничения на управляемые переменные. В самом общем виде математическую задачу оптимизации можно сформулировать следующим образом: *минимизировать (максимизировать) целевую функцию с учетом ограничений на управляемые переменные.*

Под *минимизацией (максимизацией)* функции  $n$  переменных  $f(x) = f(x_1, \dots, x_n)$  на допустимом множестве  $U$   $n$ -мерного векторного пространства  $R^n$  понимается определение хотя бы одной из точек минимума (максимума) этой функции на множестве  $U$ .

Многие практические задачи оптимизации сводятся к математическим моделям вида:

$$f(x) \rightarrow \min(\max), \quad x \in U \subset R^n. \quad (1.1)$$

Отметим, что максимизация целевой функции, например  $g(x) \rightarrow \max$ , эквивалентна минимизации противоположной величины  $f(x) = -g(x) \rightarrow \min$ , и наоборот. Поэтому, не умаляя общности, будем рассматривать или только задачи минимизации, или только задачи максимизации в зависимости от удобств изложения определенных вопросов оптимизации.

В большинстве случаев допустимое множество  $U$  определяется ограничениями-равенствами и (или) неравенствами, т.е. рассматривается задача

$$f(x) \rightarrow \min, \quad x \in R^n; \quad (1.2)$$

$$\begin{cases} g_i(x) \leq 0, & i \in I_1; \end{cases} \quad (1.3)$$

$$\begin{cases} g_i(x) = 0, & i \in I_2; \end{cases} \quad (1.4)$$

где  $I_1, I_2 \subset \{1, 2, \dots, m\}$  – заданные множества индексов, удовлетворяющие условиям  $I_1 \cap I_2 = \emptyset$ ,  $I_1 \cup I_2 = \{1, 2, \dots, m\}$ . В математической модели (1.2) – (1.4) целевая функция определена выражением (1.2), система ограничений (1.3) – (1.4) определяет область допустимых решений задачи.

Разумеется, что степень адекватности математической модели реальному объекту оптимизации определяется достаточностью полноты информации о его свойствах и параметрах.

## 1.2. Классификация задач оптимизации

Задача (1.1) представляет собой общую постановку задачи оптимизации. Классификацию задач оптимизации можно проводить по нескольким признакам в зависимости от вида функции  $f(x)$  и множества  $U$ . Далее мы выделим наиболее важные для теории и приложений оптимизационные задачи.

1. *Задача безусловной оптимизации.* Задача (1.1) называется задачей *безусловной* оптимизации, если  $U = R^n$ , т. е. если она имеет вид

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

2. *Задача условной оптимизации.* Задача (1.1) называется задачей *условной* оптимизации, если  $U$  собственное подмножество пространства  $R^n$ , т.е.  $U \subset R^n$ .

3. *Задача математического программирования.* Важнейший класс задач условной оптимизации составляют *задачи математического программирования*. Так принято называть задачу (1.2) – (1.4), т.е. задачу, допустимое множество которой задается системой конечного числа неравенств и уравнений.

При этом условия вида (1.3) называются *ограничениями - неравенствами*, вида (1.4) – *ограничениями-равенствами*.

4. *Классическая задача на условный экстремум.* Это задача условной оптимизации (1.2) – (1.4), в которой отсутствуют ограничения-неравенства, т.е.  $I_1 = \emptyset$ , и математическая модель имеет вид:

$$\begin{aligned} f(x) &\rightarrow \min, \quad x \in R^n; \\ g_i(x) &= 0, \quad i = \overline{1, m}. \end{aligned}$$

Подчеркнем, что классическая задача на условный экстремум является частным случаем задачи математического программирования.

5. *Задача линейного программирования.* Это задача условной оптимизации (1.2) – (1.4), в которой целевая функция  $f(x)$  и ограничения  $g_i(x)$  являются линейными, т.е.:

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \max ;$$

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i, & i = \overline{1, k}; \\ \sum_{j=1}^n a_{ij} x_j = b_i, & i = \overline{k+1, m}; \\ x_j \geq 0, & j = \overline{1, s}. \end{cases}$$

6. *Задача нелинейного программирования.* Это задача (1.2) – (1.4), в которой хотя бы одна из функций  $f(x)$ ,  $g_i(x)$  не является линейной.

7. *Задача выпуклого программирования.* Это задача математического программирования (1.2) – (1.4), в которой все функции  $f(x)$ ,  $g_i(x)$  ( $i \in I_1$ ) выпуклы, а ограничения-равенства линейны. Ее допустимое множество  $U$  – выпукло.

8. *Задача квадратичного программирования.* В этой задаче требуется минимизировать квадратичную функцию на допустимом множестве  $U$ , заданном линейными ограничениями - неравенствами, т.е.:

$$f(x) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \sum_{i=1}^n b_i x_i + c \rightarrow \min ;$$

$$\begin{cases} \sum_{i=1}^n \alpha_{ij} x_j \leq \beta_i, & i = \overline{1, m}; \\ x_j \geq 0, & j = \overline{1, n}. \end{cases}$$

В дальнейшем увидим, что задача квадратичного программирования – это частный случай задачи выпуклого программирования. А задача линейного программирования является частным случаем задачи квадратичного программирования, если в целевой функции  $f(x)$  коэффициенты  $a_{ij} = 0$ , ( $i, j = \overline{1, n}$ ). Эти два класса задач (квадратичного и линейного программирования) в настоящее время являются наиболее изученными.

9. *Задачи дискретной оптимизации.* Широкий класс дискретных задач оптимизации составляют *целочисленные* задачи математического программирования, которые наряду с обычными ограничениями на допустимое множество (равенствами и неравенствами) содержат тре-

бование целочисленности, налагаемое на переменные. Часто допустимое множество задачи целочисленного программирования имеет вид

$$U = \{x \in D \mid g_i(x) \leq 0, i = \overline{1, k}; \quad g_i(x) = 0, i = \overline{k+1, m}\},$$

где  $D = D_1 \times \dots \times D_n$ , причем  $D_j \subset \{0, \pm 1, \pm 2, \dots\}$ . Если координаты  $x_j$  ( $j = \overline{1, n}$ ) вектора  $x$  могут принимать только *булевы* значения, т.е.  $D_j = \{0, 1\}$ , то такая задача дискретной оптимизации называется *булевой задачей математического программирования*.

Для каждого класса задач оптимизации разработаны свои методы решения. Поэтому классификация методов оптимизации определяется классификацией задач оптимизации.

### 1.3. Общие свойства задач оптимизации

Пусть целевая функция зависит от одной переменной, а допустимым множеством является отрезок вещественной оси:

$$f(x) \rightarrow \min, \quad x \in [a, b] = U \subset R^1. \quad (1.5)$$

**Определение 1.1.** Число  $x^* \in U$  называется *точкой глобального* (абсолютного) *минимума* или просто *точкой минимума* функции  $f(x)$  на множестве  $U$ , если  $f(x^*) \leq f(x)$  для всех  $x \in U$ .

**Определение 1.2.** Число  $x^0 \in U$  называется *точкой локального минимума* функции  $f(x)$ , если  $f(x^0) \leq f(x)$  для всех  $x \in U$ , достаточно близких к  $x^0$ , т.е. если существует  $\varepsilon > 0$  такое, что это неравенство выполняется для любого  $x \in \{x \mid x \in U, |x - x^0| < \varepsilon\}$ .

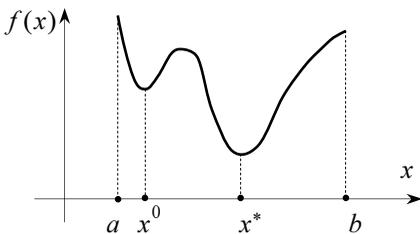


Рис. 1.1

Иначе говоря, точка  $x^0$  является *точкой минимума* функции в окрестности этой точки, а точка  $x^*$  доставляет минимум функции  $f(x)$  при всех значениях  $x \in U$ . Таким образом, глобальный минимум  $x^*$  является решением задачи минимизации функции.

Сформулированные в предыдущем параграфе задачи оптимизации имеют решение не при любых целевых функциях и допустимых множествах. Существуют задачи, в которых невозможно найти опти-

мальную точку и экстремальное значение функции. Например, не существует точек минимума функции одной переменной  $f(x)$  на множестве  $U$  в случаях, изображенных на рис. 1.2.

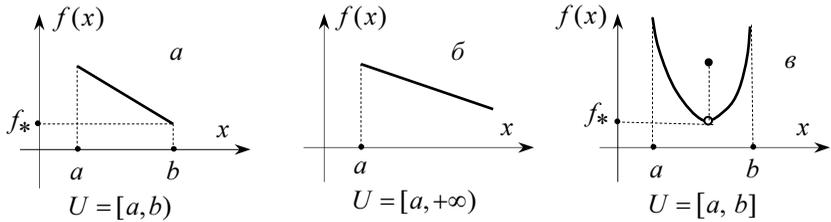


Рис. 1.2

В первом случае (рис.1.2, а) точка минимума не существует, поскольку множество  $U$  *незамкнутое*. Во втором случае (рис.1.2, б) – вследствие *неограниченности*  $U$ . Наконец, в третьем случае (рис.1.2, в) минимум не достигается из-за того, что функция  $f(x)$  *не является непрерывной*. Эти примеры приводят к мысли, что в случае непрерывной целевой функции и замкнутого ограниченного (т.е. компактного) допустимого множества задача оптимизации должна иметь решение.

Множество всех точек минимума функции  $f(x)$  на  $U$  будем в дальнейшем обозначать через  $U^*$ . Множество  $U^*$  может быть пустым, состоять из конечного или бесконечного числа точек.

### Пример 1.1.

Пусть функция  $f(x) = \sin^2(\pi/x)$  при  $x > 0$ .

1) На множестве  $U = \{x \mid x \in [1; 2]\}$  минимальное значение  $f(x)$  равно нулю, множество  $U^* = \{1\}$ , т.е. состоит из одной точки  $x^* = 1$ .

2) Если  $U = \{x \mid x \in [1/3; 1]\}$ , то  $U^* = \{1/3, 1/2, 1\}$ , т.е.  $U^*$  содержит три точки и  $f(x^*) = f_{\min} = 0$ .

3) Если  $U = \{x \mid x \in (0; 1]\}$ , то  $U^* = \{x \mid x = 1/n, n = 1, 2, \dots\}$  – счетное множество и  $f(x^*) = f_{\min} = 0$ .

4) В случае  $U = \{x \mid x \in [2; +\infty)\}$  функция  $f(x)$  не имеет наименьшего значения на  $U$ . В самом деле, для любой точки  $x \in U$  найдется точка  $y \in U$  такая, что  $f(x) > f(y)$ . Это значит, что  $U^* = \emptyset$ .

Пусть функция  $f(x)$  ограничена снизу на множестве  $U$ , т.е.  $f(x) \geq A > -\infty$  для всех  $x \in U$ . Число  $f_*$  называют *точной нижней гранью* или *инфимумом* функции  $f(x)$  на множестве  $U$

$$f_* = \inf_{x \in U} f(x), \quad (1.6)$$

если неравенство  $f_* \leq f(x)$  имеет место для всех  $x \in U$  и для любого числа  $\varepsilon > 0$  найдется точка  $x_\varepsilon \in U$  такая, что  $f(x_\varepsilon) < f_* + \varepsilon$  (т.е. среди значений  $f(x)$  на множестве  $U$  найдутся как угодно близкие к  $f_*$ ).

Если  $U^* \neq \emptyset$ , то  $\inf_U f(x) = \min_U f(x)$ . Таким образом, точная нижняя грань обобщает понятие минимума функции на случай  $U^* = \emptyset$ .

Аналогично вводится понятие точной верхней грани. Число  $f^*$  называют *точной верхней гранью* или *супремумом* функции  $f(x)$ , ограниченной сверху на множестве  $U$ , если неравенство  $f(x_\varepsilon) > f^* - \varepsilon$  справедливо для любого числа  $\varepsilon > 0$ , для которого найдется точка  $x_\varepsilon \in U$ . Для точной верхней грани используют обозначение

$$f^* = \sup_{x \in U} f(x).$$

В курсе математического анализа доказывается, что произвольная числовая функция на любом непустом допустимом множестве имеет точную нижнюю и точную верхнюю грани. Так, например, точная нижняя грань в случаях *a* и *b*, изображенных на рис. 1.2, обозначена через  $f_*$ . В случае *b* на этом рисунке  $f_* = -\infty$ .

Из примеров видно, что не всегда можно указать точку, в которой точная грань достигается, т.е. точку  $x^0$ , для которой  $f(x^0) = \inf_{x \in U} f(x)$ . Поэтому в обобщенной задаче минимизации  $f(x) \rightarrow \inf_{x \in U}$  под решением понимают не отдельную точку, как это имеет место в обычной задаче оптимизации, а последовательность точек  $\{x^{(k)}\}_{k=1}^\infty$ ,  $x^{(k)} \in U$ ,  $k=1, 2, \dots$ , такую, что

$$\lim_{k \rightarrow \infty} f(x^{(k)}) = f^0. \quad (1.7)$$

Эта последовательность всегда существует и называется *минимизирующей последовательностью*. Любая подпоследовательность мини-

минирующей последовательности сама является минимизирующей последовательностью. Поэтому, как правило, минимизирующих последовательностей «довольно много».

Таким образом, *обобщенная задача минимизации* целевой функции  $f(x)$  на множестве  $U$  заключается в отыскании числа  $f^0$  и последовательности точек  $\{x^{(k)}\}_{k=1}^{\infty}$ ,  $x^{(k)} \in U$ ,  $k = 1, 2, \dots$ , таких, что выполняются равенства (1.6) и (1.7). Как отмечено выше, эта задача всегда имеет решение.

Широкий класс функций, для которых  $U^* \neq \emptyset$ , определяется известной из математического анализа теоремой Вейерштрасса, согласно которой непрерывная на замкнутом ограниченном множестве функция достигает на этом множестве своих минимального и максимального значений.

Классический метод минимизации функции  $f(x)$ , дифференцируемой достаточное число раз, рассматривается в курсе математического анализа. При решении практических задач этот метод имеет ограниченное применение и в учебном пособии не рассматривается.

Каждый класс задач оптимизации имеет конкретный вид условий разрешимости или неразрешимости.

#### 1.4. Пример задачи оптимизации

Рассмотрим пример построения математической модели оптимизации, используя этапы, перечисленные в параграфе 1.1.

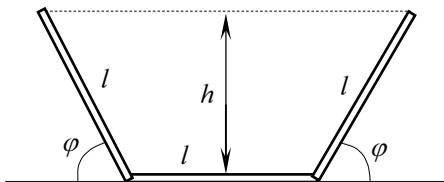


Рис. 1.3

поперечное сечение представляет собой равнобокую трапецию (рис. 1.3).

Составим варианты математической модели оптимизации проекта, выделяя этапы, перечисленные в параграфе 1.1.

##### *Определение границ объекта оптимизации*

Считаем, что выбор пути, по которому будет проложен канал, диктуется внешними, не зависящими от проектировщика условиями, оптимизации не подлежит (например, начальный и конечный пункты транспортировки воды, рельеф местности, расположение существующих транспортных магистралей и населенных пунктов и т.д.). Это

предположение существенно сужает границы рассматриваемого объекта и выбрано здесь потому, что оно позволяет построить сравнительно простые и наглядные математические модели оптимизации.

Таким образом, оптимизация должна относиться только к форме и размерам поперечного сечения канала, которые полностью определяются шириной  $l$  бетонных плит и углом наклона  $\varphi$  его боковых стенок к плоскости дна (рис. 1.3).

Дальнейшее сужение границ объекта оптимизации может быть связано с предположением о том, что ширина бетонных плит фиксирована, т.е. используются стандартные плиты заданной ширины  $l_0$ , и выбор величины  $l$  не находится во власти проектировщика. Мы рассмотрим несколько математических моделей, одна из которых включает это предположение, а другие – нет.

Далее, следует иметь в виду множество различных факторов, влияющих на качество проекта: затраты на строительство канала, его пропускную способность, потери воды при транспортировке и другие. Выбирая на основе предыдущего опыта, предварительных оценок, экспериментального исследования наиболее существенные факторы, следует ограничить их число, т.е. окончательно определить границы объекта оптимизации.

Рассмотрим следующие варианты этих ограничений:

1) используются стандартные бетонные плиты, т.е. их ширина  $l$  является заданной и равна  $l_0$ . Существенным обстоятельством считается только пропускная способность канала;

2) плиты изготавливаются по целевому заказу, т.е. их ширина может быть в некоторых пределах выбрана при проектировании. Пропускная способность канала должна быть не меньше заданной величины, а стоимость строительства является наиболее важным обстоятельством;

3) ширина плит может варьироваться, как и в варианте 2. Во внимание следует принимать пропускную способность канала, стоимость его строительства и потери воды из-за ее просачивания через стенки и дно канала, а также испарения в атмосферу.

*Выбор управляемых переменных*

Все характеристики поперечного сечения канала полностью определяются только шириной  $l$  бетонных плит и углом  $\varphi$  наклона боковых стенок к горизонтали (рис. 1.3). Поэтому управляемыми переменными являются: в варианте 1 – угол  $\varphi$  (ширина  $l = l_0$  – фиксирована); в вариантах 2 и 3 – угол  $\varphi$  и ширина  $l$ .

*Определение ограничений на управляемые переменные*

Из рис. 1.3 видны возможные пределы изменения угла  $\varphi$  :

$$0 < \varphi \leq \pi/2$$

(при  $\varphi = \pi/2$  – поперечное сечение канала превращается в квадрат, и при дальнейшем увеличении  $\varphi$  конструкция становится неустойчивой). Определяя ограничения на ширину  $l$ , необходимо учитывать, во-первых, что  $l$  может быть только положительной и, во-вторых, что ширина плит не может быть произвольно большой и ограничивается на известном уровне  $l_{\max}$  возможностями оборудования завода, производящего плиты. Таким образом,

$$0 < l \leq l_{\max}.$$

Далее, в варианте 2 возникает еще одно ограничение, связанное с тем, что задана минимально допустимая пропускная способность  $W_{\min}$  проектируемого канала. Под пропускной способностью канала  $W$  понимается объем воды, проходящей через его поперечное сечение в единицу времени. Очевидно,  $W = S \cdot v$ , где  $S$  – площадь поперечного сечения,  $v$  – скорость воды, которую для простоты будем считать постоянной вдоль всего канала.

Из элементарной геометрии следует, что  $S = l^2(1 + \cos \varphi) \sin \varphi$ , поэтому упомянутое ограничение на управляемые переменные  $l$  и  $\varphi$  записывается следующим образом:

$$W = v l^2 (1 + \cos \varphi) \sin \varphi \geq W_{\min}.$$

*Выбор числового критерия оптимизации*

Случай 1. Единственной целью оптимизации системы является обеспечение наибольшей пропускной способности канала:

$$W = v l^2 (1 + \cos \varphi) \sin \varphi.$$

Оптимальной ситуации соответствует максимум величины  $W$ , и числовым критерием оптимизации в этом случае является целевая функция одной переменной:

$$f_1(\varphi) = v l^2 (1 + \cos \varphi) \sin \varphi \rightarrow \max.$$

Случай 2. Целью оптимизации системы является минимизация стоимости строительства канала. Будем считать, что общие затраты  $C$  на строительство определяются следующими факторами:

– стоимостью  $C_1$  земляных работ, пропорциональной объему извлеченного грунта:  $C_1 = c_1 S L = c_1 l^2 (1 + \cos \varphi) L \sin \varphi$ , где  $c_1$  – затраты на извлечение единицы объема грунта,  $L$  – длина канала;

– общей стоимостью  $C_2$  производства, доставки к месту работ и укладки бетонных плит. Объединяя эти затраты, полагаем их пропорциональными суммарной площади бетонного покрытия, т.е.  $C_2 = c_2 p L$ , где  $c_2$  – указанные затраты, отнесенные к единице площади плит,  $p = 3l$  – общая ширина бетонных плит, укладываемых на любом участке канала. Поэтому  $C_2 = 3c_2 l L$ .

Таким образом, общая стоимость строительства составит:

$$C = C_1 + C_2 = \left[ c_1 l^2 (1 + \cos \varphi) \sin \varphi + 3c_2 l \right] L.$$

В рассматриваемом случае целью оптимизации проекта является обеспечение наименьшей стоимости строительства канала, т.е. оптимальной ситуации соответствует минимум величины  $C$ , и числовым критерием будет целевая функция двух переменных:

$$f_2(l, \varphi) = C = \left[ c_1 l^2 (1 + \cos \varphi) \sin \varphi + 3c_2 l \right] L \rightarrow \min.$$

Случай 3. В данном случае необходимо учесть сразу несколько, к тому же противоречивых, целей оптимизации (максимум пропускной способности, и минимум стоимости строительства и потерь воды).

Выразим величину потерь воды  $Q$  через управляемые переменные  $l$  и  $\varphi$ . Для этого необходимо знать высоту  $h$  воды в канале, которая в реальном случае может зависеть от времени:  $h = h(t)$ . Данное обстоятельство затрудняет построение математической модели. Поэтому можно вместо  $h$  использовать среднюю величину за большой

период времени  $T$ :  $\bar{h} = \frac{1}{T} \int_0^T h(t) dt$ , выбирая  $T$  кратным году. Если

же канал предполагается использовать только на максимальную мощность, то  $h = l \sin \varphi$  (рис. 1.3). Мы примем именно это предположение.

Тогда потери воды на испарение, пропорциональные площади ее открытой поверхности, равны  $Q_1 = q_1 l (1 + 2 \cos \varphi) L$ , где  $q_1$  – количество воды, испаряющейся с единицы площади поверхности в единицу времени.

Потери  $Q_2$  на просачивание воды пропорциональны площади бетонного покрытия канала:  $Q_2 = 3q_2 l L$ , где  $q_2$  – потери, относящиеся к единице площади поверхности бетона в единицу времени. Таким образом, функцию, соответствующую потерям воды, следует минимизировать, т.е.:

$$f_3(l, \varphi) = Q_1 + Q_2 = q_1 l (1 + 2 \cos \varphi) L + 3q_2 l L \rightarrow \min .$$

В том случае, когда математическая модель оптимизации должна учитывать три различные цели, а именно достижение максимальной пропускной способности канала, минимальной стоимости его строительства и минимальных потерь воды, то можно построить общую целевую функцию с использованием весовых коэффициентов. Например, можно использовать целевую функцию:

$$f(l, \varphi) = -\alpha_1 f_1(l, \varphi) + \alpha_2 f_2(l, \varphi) + \alpha_3 f_3(l, \varphi) \rightarrow \min ,$$

где  $\alpha_1$ ,  $\alpha_2$  и  $\alpha_3$  – весовые коэффициенты, выбранные с учетом важности каждой из целей (чем важнее цель, тем большее значение приписывается соответствующему коэффициенту).

Другой путь построения математической модели при наличии нескольких целей оптимизации состоит в том, что выбирается основная цель, а остальные учитываются с помощью дополнительных ограничений на управляемые переменные. Этот путь здесь не рассматривается.

## Глава 2

### МЕТОДЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В настоящей главе рассмотрены теоретические и вычислительные аспекты симплекс-метода, который является основным методом решения задач линейного программирования.

На основании теории двойственности рассмотрен двойственный симплекс-метод, алгоритмические особенности которого позволяют решать специального вида задачи линейного программирования.

Из специальных задач линейного программирования описаны транспортные, параметрические, дробно-линейные задачи и методы их решения.

#### 2.1. Постановка задачи линейного программирования

Задача линейного программирования является простейшим и наиболее полно исследованным частным случаем задачи математического программирования, состоящей в минимизации или максимизации ли-

нейной функции  $f(x) = f(x_1, \dots, x_n) = \sum_{j=1}^n c_j x_j$  на некотором множе-

стве  $U \subset R^n$ , заданном системой линейных ограничений и условием неотрицательности вектора переменных  $x \in U$ .

**Определение 2.1.** *Задачей линейного программирования* называется задача поиска экстремального значения линейной функции  $f(x)$ , определенной на некотором выпуклом *многограннике*  $U \subset R^n$ .

Таким образом, задачу линейного программирования можно записать в следующем виде:

$$f(x) = \sum_{j=1}^n c_j x_j = \langle c, x \rangle \rightarrow \max(\min); \quad (2.1)$$

$$\begin{cases} \sum_{j=1}^n a_j^{(i)} x_j = \langle \mathbf{a}^{(i)}, \mathbf{x} \rangle \leq b_i, & i \in I_1; \end{cases} \quad (2.2)$$

$$\begin{cases} \sum_{j=1}^n a_j^{(i)} x_j = \langle \mathbf{a}^{(i)}, \mathbf{x} \rangle = b_i, & i \in I_2; \end{cases} \quad (2.3)$$

$$x_j \geq 0; \quad j \in J. \quad (2.4)$$

где  $\mathbf{c} = (c_1, \dots, c_n)$  и  $\mathbf{a}^{(i)} = (a_1^{(i)}, \dots, a_n^{(i)})$  – фиксированные векторы в  $R^n$ ;  $I_1$  и  $I_2$  – некоторые множества индексов, удовлетворяющие условиям  $I_1 \cup I_2 = \{1, \dots, m\}$ ,  $I_1 \cap I_2 = \emptyset$ ;  $J = \{1, \dots, k\}$ ,  $k \leq n$ .

Функция (2.1) называется *целевой функцией* задачи (2.1) – (2.4). Для определенности далее будем считать, что  $f(\mathbf{x}) \rightarrow \max$ . В том случае, когда требуется найти *минимум* целевой функции  $f(\mathbf{x})$ , можно перейти к поиску *максимума* новой целевой функции  $f_1(\mathbf{x}) = -f(\mathbf{x})$ , т.к.  $\max(f_1(\mathbf{x})) = \max(-f(\mathbf{x})) = -\min(f(\mathbf{x}))$ . Условия (2.2) и (2.3) называются *системой ограничений*, а условия (2.4) – условиями *неотрицательности* данной задачи. Системы уравнений (2.2), неравенств (2.3) и (2.4) образуют *область допустимых решений*  $U \subset R^n$ . Математическая модель (2.1) – (2.4) представляет собой *общую форму* математической модели задачи линейного программирования. Она получается при непосредственной формализации задачи. При этом видно, что условиям неотрицательности (2.4) могут удовлетворять не все переменные.

Кроме общей формы математической модели (2.1) – (2.4) в линейном программировании определены еще две формы математической модели: *стандартная* и *каноническая*.

*Стандартной* (или *симметричной*) *формой* задачи ЛП называется задача линейного программирования, в которой  $I_2 = \emptyset$ :

$$f(\mathbf{x}) = \sum_{j=1}^n c_j x_j = \langle \mathbf{c}, \mathbf{x} \rangle \rightarrow \max;$$

$$\sum_{j=1}^n a_j^{(i)} x_j = \langle \mathbf{a}^{(i)}, \mathbf{x} \rangle \leq b_i, \quad i = \overline{1, m};$$

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

Канонической (или основной) формой задачи линейного программирования называется задача, в которой отсутствуют ограничения–неравенства, т.е.  $I_1 = \emptyset$  :

$$f(x) = \sum_{j=1}^n c_j x_j = \langle c, x \rangle \rightarrow \max ; \quad (2.5)$$

$$\sum_{j=1}^n a_j^{(i)} x_j = \langle a^{(i)}, x \rangle = b_i, \quad i = \overline{1, m} ; \quad (2.6)$$

$$x_j \geq 0, \quad j = \overline{1, n} . \quad (2.7)$$

Вектор  $x = (x_1, x_2, \dots, x_n) \subset U$  называется допустимым решением. Вектор  $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ , при котором целевая функция  $f(x)$  принимает свое максимальное значение, называется оптимальным решением задачи линейного программирования.

Указанные три формы задачи линейного программирования (общая, стандартная и каноническая) эквивалентны между собой. Иначе говоря, любую задачу из одной формы в другую можно привести с помощью несложных преобразований. Это означает, что симплекс–методом, который применим только к канонической форме, можно решить любую задачу линейного программирования.

Преобразование, например, общей или канонической форм задачи в стандартную форму позволяет уменьшить размерность задачи.

Для перехода от одной формы задачи к другой достаточно следующих эквивалентных преобразований:

1) От ограничений–неравенств в задачах линейного программирования всегда можно перейти к ограничениям–равенствам введением дополнительных неотрицательных переменных. Например, прибавив к левой части неравенства  $\langle a^{(i)}, x \rangle \leq b_i$  дополнительную переменную  $x_{n+i} \geq 0$ , получим ограничение–равенство  $\langle a^{(i)}, x \rangle + x_{n+i} = b_i$ . При этом в выражение целевой функции новая переменная  $x_{n+i}$  входит с коэффициентом  $c_{n+i} = 0$ . Размерность пространства управляемых переменных задачи увеличится на число введенных дополнительных переменных.

Имеет место и обратный переход: отбрасывая неотрицательную базисную переменную, можно перейти от уравнения к неравенству. Таким образом, справедливы следующие преобразования:

$$\langle a_i, x \rangle \leq b_i, \quad i = \overline{1, m}; \quad \longleftrightarrow \quad \begin{cases} \langle a_i, x \rangle + x_{n+i} = b_i, \\ x_{n+i} \geq 0, \quad i = \overline{1, m}. \end{cases}$$

2) Если некоторая переменная  $x_j$  не подчинена условию неотрицательности, то ее следует заменить двумя дополнительными неотрицательными переменными  $x'_j$  и  $x''_j$ , приняв  $x_j = x'_j - x''_j$ . Правомерность такой замены очевидна, так как любое число можно представить в виде разности двух неотрицательных чисел.

### Примеры задач линейного программирования

Существует большое количество задач линейного программирования в различных сферах практической деятельности. Рассмотрим несколько примеров, ставших классическими.

**Пример 2.1.** (*Задача планирования производства*). Пусть некоторому предприятию требуется определить оптимальный план производства  $n$  типов товаров, обеспечивающий максимальную прибыль. При этом известны технология производства и объемы используемых ресурсов. Предполагаются известными следующие параметры:

$m$  – число видов ресурсов;

$a_{ij}$  – количество  $i$ -го ресурса, необходимого для производства единицы  $j$ -го вида товара ( $i = \overline{1, m}$ ,  $j = \overline{1, n}$ );

$b_i$  – запас  $i$ -го ресурса на предприятии,  $b_i \geq 0$ ;

$c_j$  – цена единицы  $j$ -го товара.

Технология производства предполагается линейной, т.е. все затраты ресурсов растут прямо пропорционально объему выпуска. Пусть число  $x_j$  показывает планируемый объем производства  $j$ -го товара.

*Целевая функция* задачи – максимальная прибыль при реализации выпускаемой продукции:

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \max. \quad (2.8)$$

Из условия ограниченности ресурсов, с одной стороны, и суммарных затрат ресурсов для выполнения плана производства – с другой, строится система ограничений математической модели:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}; \quad (2.9)$$

$$x_j \geq 0, \quad j = \overline{1, n}. \quad (2.10)$$

Таким образом, задача планирования производства ставится следующим образом: среди всех векторов  $x = (x_1, x_2, \dots, x_n)$ , удовлетворяющих ограничениям (2.9) и (2.10), найти такой, при котором целевая функция (2.8) принимает наибольшее значение. Получилась задача линейного программирования в стандартной форме.

**Пример 2.2.** (*Задача оптимальной смеси*). Задача определения оптимального состава смеси возникает тогда, когда из имеющихся видов сырья необходимо получить путем смешивания новый продукт с заданными характеристиками. При этом требуется, чтобы стоимость конечного продукта была минимальной.

Предположим, что смесь требуется составить из  $n$  различных видов сырья, каждое из которых содержит  $m$  интересующих нас элементов (веществ). Пусть  $a_{ij}$  ( $i = \overline{1, m}, j = \overline{1, n}$ ) – количество  $i$ -го вещества в единице  $j$ -го вида сырья, стоимость единицы которого равна  $c_j$ ,  $j = \overline{1, n}$ . Обозначим через  $b_i$  наименьшее допустимое количество  $i$ -го вещества и через  $d_j$  – запас (объем) сырья  $j$ -го вида.

Пусть  $x_j$  – количество сырья  $j$ -го вида, которое необходимо использовать для составления смеси. Тогда введенные обозначения позволяют построить следующую математическую модель задачи:

целевая функция 
$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \min ;$$

ограничения 
$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = \overline{1, m};$$

$$0 \leq x_j \leq d_j, \quad j = \overline{1, n}.$$

Условие неотрицательности переменных  $x_j \geq 0$  является естественным и вытекает из определения этих переменных.

**Пример 2.3.** (*Транспортная задача*). Пусть некоторый однородный продукт хранится на  $m$  складах и потребляется в  $n$  пунктах. Известны следующие параметры:

$a_i$  – запас продукта на  $i$ -м складе,  $a_i > 0$  ( $i = \overline{1, m}$ );

$b_j$  – потребность в  $j$ -м пункте,  $b_j > 0$  ( $j = \overline{1, n}$ );

$c_{ij}$  – стоимость перевозки единичного количества продукта с  $i$ -го склада в  $j$ -ый пункт,  $c_{ij} \geq 0$ .

При этом предполагается, что суммарные запасы равны суммарным потребностям:

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

Транспортная задача формулируется как каноническая форма задачи линейного программирования в следующем виде:

$$f(x) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min ;$$

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

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

где  $x_{ij}$  – количество продукта, перевозимого с  $i$ -го склада в  $j$ -ый пункт.

Иными словами, требуется так организовать перевозки продукта со складов в пункты потребления, чтобы при полном удовлетворении потребностей минимизировать суммарные транспортные расходы.

## 2.2. Теоретические основы линейного программирования

Напомним основные понятия линейной алгебры, которые будем использовать в дальнейшем при рассмотрении теоретических вопросов линейного программирования и алгоритмов решения задач.

**Определение 2.2.** Множество  $R^n$  называется *линейным векторным пространством*, если для любых двух элементов  $x, y \in R^n$  определена *сумма*  $z = x + y$ ,  $z \in R^n$  и для любого элемента  $x \in R^n$  и любого действительного числа  $\lambda \in R$  определено *произведение*  $\lambda x \in R^n$ . Причем введенные операции сложения и умножения удовлетворяют следующим свойствам:

- $x + y = y + x$  (коммутативность);
- $(x + y) + z = x + (y + z)$  (ассоциативность);
- $\lambda(x + y) = \lambda x + \lambda y$  (дистрибутивность);
- для всякого  $x \in R^n$  существует такой элемент  $-x \in R^n$ , что  $x + (-x) = 0$ .

Элементы  $x = (x_1, x_1, \dots, x_n)$  этого пространства называются *векторами* (или *точками*), где  $x_i$  – *компоненты* вектора  $x$ .

**Определение 2.3.** Если вектор  $y \in R^n$  выражается через векторы  $x^{(1)}, x^{(2)}, \dots, x^{(k)} \in R^n$  в виде  $y = \sum_{i=1}^k \lambda_i x^{(i)}$ ,  $\lambda_i \in R$ , то говорят, что  $y \in R^n$  есть *линейная комбинация* векторов  $x^{(1)}, x^{(2)}, \dots, x^{(k)} \in R^n$ .

Далее будем использовать следующее обозначение  $i$ -го вектора:  $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)})$ ,  $x^{(i)} \in R^n$ .

**Определение 2.4.** Векторы  $x^{(1)}, \dots, x^{(k)} \in R^n$  называются *линейно независимыми*, если равенство  $\lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \dots + \lambda_k x^{(k)} = 0$ , выполняется лишь при  $\lambda_1 = \lambda_2 = \dots = \lambda_k = 0$ . Если же это равенство может выполняться и в том случае, когда не все числа  $\lambda_1, \lambda_2, \dots, \lambda_k$  равны нулю, то векторы  $x^{(1)}, \dots, x^{(k)} \in R^n$  называются *линейно зависимыми*.

**Определение 2.5.** Линейное векторное пространство  $R^n$  называется  *$n$ -мерным*, если в нем найдется система  $n$  линейно независимых векторов, но любые  $(n + 1)$  векторов линейно зависимые.

**Определение 2.6.** Совокупность  $n$  линейно независимых векторов  $n$ -мерного пространства  $R^n$  называется *базисом* этого пространства.

**Определение 2.7.** Пусть в вещественном линейном пространстве  $R^n$  каждой паре векторов  $x, y \in R^n$  поставлено в соответствие вещественное число, обозначаемое  $\langle x, y \rangle$ , т.е. задано отображение  $\langle x, y \rangle \in R^n \times R^n \rightarrow R^1$ , и при этом выполнены следующие требования:

- $\langle x, y \rangle = \langle y, x \rangle$  (коммутативность);
- $\langle \lambda x, y \rangle = \lambda \langle x, y \rangle$ ,  $\lambda \in R^1$  (однородность);
- $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$  (дистрибутивность);
- $\langle x, x \rangle \geq 0$ , причем равенство имеет место тогда и только тогда, когда  $x = 0$  (дефинитность).

Тогда говорят, что в  $R^n$  задано *скалярное произведение*.

**Определение 2.8.** Линейное пространство  $R^n$  называется *евклидовым пространством*, если для любых двух векторов  $x, y \in R^n$  определено их *скалярное произведение*:  $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$ .

Понятие скалярного произведения позволяет ввести понятие *длины вектора*

$$|x| = \sqrt{\langle x, x \rangle} = \sqrt{\sum_{i=1}^n x_i^2}.$$

Все последующие построения выполняются в евклидовом пространстве, поэтому сохраним для него введенное обозначение  $R^n$  линейного пространства.

**Теорема 2.1.** (О представлении вектора в произвольном базисе).  
Любой вектор  $x \in R^n$  можно представить, и при этом единственным образом, в виде линейной комбинации векторов базиса.

**Доказательство.** Пусть векторы  $e^{(1)}, \dots, e^{(n)} \in R^n$  образуют некоторый базис, а  $x \in R^n$  – произвольный вектор. Тогда система векторов  $x, e^{(1)}, \dots, e^{(n)} \in R^n$  линейно зависима (по определению 2.5). Следовательно, в линейной комбинации  $\lambda_0 x + \lambda_1 e^{(1)} + \lambda_2 e^{(2)} + \dots + \lambda_n e^{(n)} = 0$  не все  $\lambda_i = 0$   $i = 0, \dots, n$ . В частности,  $\lambda_0 \neq 0$ , иначе система векторов  $e^{(1)}, \dots, e^{(n)} \in R^n$  линейно зависима, что противоречит исходному предположению. Таким образом,

$$x = -\frac{\lambda_1}{\lambda_0} e^{(1)} - \frac{\lambda_2}{\lambda_0} e^{(2)} - \dots - \frac{\lambda_n}{\lambda_0} e^{(n)}. \quad (2.11)$$

Для доказательства *единственности* представления (2.11) допустим, что для  $x$  существуют два различных разложения:

$$x = \alpha_1 e^{(1)} + \alpha_2 e^{(2)} + \dots + \alpha_n e^{(n)}, \quad (2.12)$$

$$x = \beta_1 e^{(1)} + \beta_2 e^{(2)} + \dots + \beta_n e^{(n)}. \quad (2.13)$$

Вычитая почленно из (2.12) выражение (2.13), получим:

$$0 = (\alpha_1 - \beta_1) e^{(1)} + (\alpha_2 - \beta_2) e^{(2)} + \dots + (\alpha_n - \beta_n) e^{(n)}. \quad (2.14)$$

В силу независимости векторов  $e^{(1)}, e^{(2)}, \dots, e^{(n)}$  разложение (2.14) возможно лишь при  $(\alpha_i - \beta_i) = 0$ ,  $i = \overline{1, n}$  или при  $\alpha_i = \beta_i$ ,  $i = \overline{1, n}$ . Представления (2.12) и (2.13) совпадают.

**Определение 2.9.** Выпуклой комбинацией векторов  $x^{(i)} \in R^n$ ,  $i = \overline{1, k}$  называется множество вида:

$$U = \left\{ x \in R^n \mid x = \sum_{i=1}^k \lambda_i x^{(i)}, \sum_{i=1}^k \lambda_i = 1, \lambda_i \geq 0, i = \overline{1, k} \right\}.$$

**Определение 2.10.** Множество  $U \subset R^n$  называется выпуклым, если вместе с любыми двумя точками этого множества оно содержит и отрезок, соединяющий эти точки, т.е. если  $x^{(1)}, x^{(2)} \in U$ ,  $x = \lambda_1 x^{(1)} + \lambda_2 x^{(2)}$ ,  $\lambda_1 + \lambda_2 = 1$ ,  $\lambda_1 \geq 0$  и  $\lambda_2 \geq 0$ , то  $x \in U$ .

**Теорема 2.2.** (О пересечении выпуклых множеств). Пересечение конечного числа выпуклых множеств является множеством выпуклым.

*Доказательство.* Пусть  $U_i \subset R^n$ ,  $i = \overline{1, n}$  – выпуклые множества,  $U = \bigcap_{i=1}^n U_i$  и точки  $x^{(1)}, x^{(2)} \in U$ . Следовательно,  $x^{(1)}, x^{(2)} \in U_i$  для каждого  $i = \overline{1, n}$ . В силу выпуклости множеств  $U_i \subset R^n$ ,  $i = \overline{1, n}$  получаем, что отрезок  $[x^{(1)}, x^{(2)}] \subset U_i$ , для каждого  $i = \overline{1, n}$ . Отсюда следует, что отрезок  $[x^{(1)}, x^{(2)}]$  принадлежит пересечению множеств  $U_i$ ,  $i = \overline{1, n}$ , т.е.  $[x^{(1)}, x^{(2)}] \subset U$ .

**Определение 2.11.** Выпуклой оболочкой произвольного множества  $U$  называется множество, полученное как пересечение всех выпуклых множеств, содержащих  $U$ .

### Пример 2.4

Показать, что множество точек  $x = (x_1, x_2) \in R^2$  полуплоскости  $U = \{ (x_1, x_2) \mid ax_1 + bx_2 \leq c \}$  выпукло.

*Решение.* Пусть  $x^{(1)} = (x_1^{(1)}, x_2^{(1)})$  и  $x^{(2)} = (x_1^{(2)}, x_2^{(2)}) \in U$ , а  $x = (x_1, x_2) = \lambda_1 x^{(1)} + \lambda_2 x^{(2)}$  при  $\lambda_1 + \lambda_2 = 1$ ,  $\lambda_1 \geq 0$  и  $\lambda_2 \geq 0$  точки отрезка, соединяющего точки  $x^{(1)}$  и  $x^{(2)}$  или  $x_1 = \lambda_1 x_1^{(1)} + \lambda_2 x_1^{(2)}$  и  $x_2 = \lambda_1 x_2^{(1)} + \lambda_2 x_2^{(2)}$ . Покажем, что  $x \in U$ . Имеем:

$$\begin{aligned} ax_1 + bx_2 &= a(\lambda_1 x_1^{(1)} + \lambda_2 x_1^{(2)}) + b(\lambda_1 x_2^{(1)} + \lambda_2 x_2^{(2)}) = \\ &= \lambda_1 (ax_1^{(1)} + bx_2^{(1)}) + \lambda_2 (ax_1^{(2)} + bx_2^{(2)}). \end{aligned}$$

Используя неравенства  $ax_1^{(1)} + bx_2^{(1)} \leq c$  и  $ax_1^{(2)} + bx_2^{(2)} \leq c$  (т.к. точки  $x^{(1)}$  и  $x^{(2)}$  принадлежат допустимой полуплоскости  $U$ ), получим:

$$ax_1 + bx_2 \leq \lambda_1 c + \lambda_2 c = c, \text{ т.е. } x \in U.$$

Аналогично можно показать, что полупространство  $U \subset R^n$  выпукло.

Из теоремы 2.2 и примера 2.4 следует, что множество решений совместной системы линейных неравенств является выпуклым многогранником (на плоскости – выпуклым многоугольником), т.к. каждое неравенство определяет одно полупространство, являющееся выпуклым множеством точек, а системе неравенств соответствует пересечение полупространств. Таким образом, множеством решений совместной системы линейных неравенств являются точки, которые принадлежат пересечению полупространств решений всех неравенств.

Наиболее простым многогранником в смысле наименьшего числа граней является симплекс.

**Определение 2.12.** *Симплексом ( $n$ -мерным симплексом)* называется выпуклая оболочка такого  $(n + 1)$ -элементного множества вершин многогранника не менее чем  $n$ -мерного векторного пространства  $R^n$ , в котором любые  $n$  вершин линейно независимы.

В частности, отрезок является одномерным симплексом. Двумерный симплекс по определению треугольник, а трехмерный симплекс – тетраэдр.

**Определение 2.13.** *Крайней точкой*  $x \in U \subset R^n$  выпуклого множества  $U$  называется такая точка, которая не может быть представлена выпуклой комбинацией каких-либо двух точек  $x^{(1)}, x^{(2)} \in U$ .

**Определение 2.14.** Множество

$$S = \{x \mid |x - x^{(0)}| < \varepsilon; x, x^{(0)} \in R^n\}$$

называется  $\varepsilon$ -окрестностью точки  $x^{(0)}$ .

**Определение 2.15.** Точка  $x^{(0)} \in U$  называется *внутренней точкой* множества  $U$ , если существует такая ее  $\varepsilon$ -окрестность, все точки которой принадлежат  $U$ .

**Определение 2.16.** Точка  $x^{(0)} \in U$  называется *граничной точкой* множества  $U$ , если любая ее окрестность содержит как точки, принадлежащие множеству  $U$ , так и точки, не принадлежащие множеству  $U$ .

**Определение 2.17.** Множество  $U \subset \mathbb{R}^n$  называется *ограниченным*, если существует такая константа  $C$ , что  $|x| \leq C$ , для любого  $x \in U$ .

**Определение 2.18.** Множество  $U \subset \mathbb{R}^n$  называется *замкнутым*, если оно содержит все свои граничные точки.

**Теорема 2.3.** (О представлении). Любая точка замкнутого, ограниченного, выпуклого многогранника  $U \subset \mathbb{R}^n$  может быть представлена выпуклой комбинацией крайних точек этого многогранника.

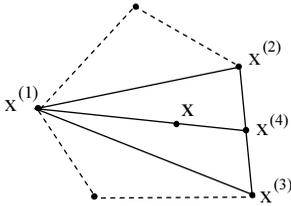


Рис. 2.1

**Доказательство.** Проведем доказательство индукцией по наименьшей размерности пространства  $\mathbb{R}^n$ , содержащего  $U$ .

1) Если  $n = 1$ , то  $U$  является отрезком на прямой и утверждение теоремы очевидно.

2) Предположим, что  $n = 2$ . Тогда  $U \subset \mathbb{R}^2$  – многоугольник, имеющий  $m \geq 3$

вершин. Выпуклый многоугольник с  $m > 3$  вершинами всегда можно разбить с помощью непересекающихся диагоналей на  $(m - 2)$  треугольников. Тогда произвольная точка  $x \in U$  будет принадлежать одному из треугольников (рис. 2.1). Если точка  $x$  – граничная точка симплекса (треугольника), то имеем случай (1), т.е.  $n = 1$ . Докажем, что точка треугольника удовлетворяет теореме. Без ограничения общности можно положить, что точка  $x$  принадлежит треугольнику с вершинами  $x^{(1)}, x^{(2)}, x^{(3)}$ . Пусть  $x^{(1)}, x^{(2)}, x^{(3)} \in U$  – угловые точки симплекса. Через точку  $x$  проведем отрезок  $[x^{(1)}, x^{(4)}]$  (рис. 2.1). Так как точка  $x$  принадлежит этому отрезку, то

$$x = t_1 x^{(1)} + t_4 x^{(4)}, \quad (2.15)$$

где  $t_1 + t_4 = 1$ ,  $t_1 \geq 0$ ,  $t_4 \geq 0$ .

Точка  $x^{(4)}$  принадлежит отрезку  $[x^{(2)}, x^{(3)}]$ , следовательно, является выпуклой комбинацией его концов, т.е.:

$$x^{(4)} = t_2 x^{(2)} + t_3 x^{(3)}; \quad t_2 + t_3 = 1, \quad t_2 \geq 0, \quad t_3 \geq 0. \quad (2.16)$$

Подставляя (2.16) в (2.15), получаем:

$$x = t_1 x^{(1)} + t_4 (t_2 x^{(2)} + t_3 x^{(3)}) = t_1 x^{(1)} + t_2 t_4 x^{(2)} + t_3 t_4 x^{(3)}.$$

Вводим новые переменные  $\lambda_1 = t_1$ ,  $\lambda_2 = t_2 t_4$ ,  $\lambda_3 = t_3 t_4$ , при этом  $\lambda_1 + \lambda_2 + \lambda_3 = t_1 + t_2 t_4 + t_3 t_4 = t_1 + t_4(t_2 + t_3) = t_1 + t_4 = 1$ . Следовательно,

$$x = \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \lambda_3 x^{(3)}, \quad \sum_{i=1}^3 \lambda_i = 1, \quad \lambda_i \geq 0, \quad i = 1, 2, 3, \quad (2.17)$$

т.е. точка  $x$  – выпуклая комбинация угловых (крайних) точек  $x^{(1)}$ ,  $x^{(2)}$ ,  $x^{(3)}$ . Добавляя к правой части выражения (2.17) остальные  $(m-3)$  вершины, получим:

$$x = \lambda_1 x^{(1)} + \lambda_2 x^{(2)} + \lambda_3 x^{(3)} + 0x^{(4)} + \dots + 0x^{(m)}; \quad \sum_{i=1}^m \lambda_i = 1, \quad \lambda_i \geq 0, \quad i = \overline{1, m}.$$

Таким образом, точка  $x$  представлена выпуклой комбинацией угловых точек многоугольника.

3) Предположим, что теорема доказана для  $k = n-1$ . Если  $x$  – граничная точка многогранника  $U \subset R^n$ , то  $x$  принадлежит гиперплоскости  $H = \{x \in R^n : \langle c, x \rangle = b\}$ , которая имеет размерность  $n-1$ . По индукции заключаем, что существуют такие крайние точки  $x^{(i)}$ ,  $i = \overline{1, m}$  выпуклого замкнутого множества  $H \cap U$ , что

$$x = \sum_{i=1}^m \lambda_i x^{(i)}, \quad \sum_{i=1}^m \lambda_i = 1, \quad \lambda_i \geq 0, \quad i = \overline{1, m}.$$

Из определения многогранника следует, что все  $x^{(i)}$  являются также угловыми (крайними) точками множества  $U$ .

Пусть  $x$  – внутренняя точка множества  $U$ . Проведем через  $x$  прямую  $l$ . Пересечение  $l \cap U$  является отрезком  $[x^{(1)}, x^{(2)}]$  с концами  $x^{(1)}, x^{(2)} \in U$ . Поскольку для граничных точек  $x^{(1)}, x^{(2)}$  теорема верна, то она верна и для  $x$ . Действительно, для граничных точек имеют место соотношения:

$$x^{(1)} = \sum_{i=1}^s \alpha_i y^{(i)}, \quad \sum_{i=1}^s \alpha_i = 1, \quad \alpha_i \geq 0, \quad i = \overline{1, s};$$

$$x^{(2)} = \sum_{i=1}^r \beta_i z^{(i)}, \quad \sum_{i=1}^r \beta_i = 1, \quad \beta_i \geq 0, \quad i = \overline{1, r},$$

где  $y^{(i)}, z^{(i)}$  – угловые точки множества  $U$ . Поскольку  $x$  – внутренняя точка множества  $U$ , то существует такое  $\gamma \in [0, 1]$ , что  $x = \gamma x^{(1)} + (1-\gamma)x^{(2)}$ . Тогда:

$$x = \sum_{i=1}^s \gamma \alpha_i y_i + \sum_{i=1}^r (1-\gamma) \beta_i z_i,$$

и т.к.  $\sum_{i=1}^s \gamma \alpha_i + \sum_{i=1}^r (1-\gamma) \beta_i = \gamma \sum_{i=1}^s \alpha_i + (1-\gamma) \sum_{i=1}^r \beta_i = \gamma + (1-\gamma) = 1,$

то это доказывает утверждение теоремы.

Система ограничений (2.6) канонической формы математической модели (2.5) – (2.7) представляет собой систему линейных алгебраических уравнений:

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m} \quad (2.18)$$

или в матричной форме

$$Ax = b, \quad x \in R^n, \quad b \in R^m.$$

*Решением системы уравнений (2.18)* называется вектор  $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$ , для которого все уравнения (2.18) являются верными числовыми равенствами.

Система уравнений *совместна*, если у нее есть хотя бы одно решение. Совместная система *определенна*, если она имеет единственное решение, и *неопределенна*, если имеет бесконечное множество решений. Две системы уравнений *эквивалентны (равносильны)*, если любое решение первой системы является решением второй и любое решение второй является решением первой.

Процесс решения системы уравнений основан на *эквивалентных преобразованиях* исходной системы в систему, решения которой очевидны. Он основан на следующих фактах:

1) Если к системе (2.18) добавить уравнение, которое является линейной комбинацией уравнений этой системы, то получится система, равносильная (2.18).

2) Если к одному уравнению системы (2.18) прибавить линейную комбинацию других уравнений системы, а остальные уравнения оставить без изменения, то получится система, эквивалентная исходной.

3) Если уравнения системы умножить на скаляры, отличные от нуля, то получится система, эквивалентная исходной.

4) Если в системе переставить уравнения, то получится система, равносильная исходной.

На этих преобразованиях основан, например, метод Жордана–Гаусса (метод полного исключения), который позволяет решать системы линейных алгебраических уравнений вида (2.18) и который используется для преобразований системы уравнений в симплекс-методе

при поиске оптимального решения задачи линейного программирования.

**Метод Жордана–Гаусса.**

Пусть уравнение с номером  $r$  системы (2.18) разрешается относительно  $k$ -ой переменной:

$$x_k = a_{rk}^{-1} \left( b_r - \sum_{j=1, j \neq k}^n a_{rj} x_j \right),$$

и это выражение для  $x_k$  подставляется в остальные уравнения. В результате получаем преобразованную систему уравнений, эквивалентную (2.18):

$$\begin{cases} x_k + \sum_{j=1, j \neq k}^n a_{rj} a_{rk}^{-1} x_j = a_{rk}^{-1} b_r; \\ \sum_{j=1, j \neq k}^n (a_{ij} - a_{ik} a_{rj} a_{rk}^{-1}) x_j = b_i - a_{ik} a_{rk}^{-1} b_r, \quad i \neq r. \end{cases}$$

Из последних выражений вытекают следующие формулы преобразований метода Жордана–Гаусса:

$$a'_{rj} = \frac{a_{rj}}{a_{rk}}, \quad b'_r = \frac{b_r}{a_{rk}}, \quad (i = r); \quad (2.19)$$

$$a'_{ij} = a_{ij} - \frac{a_{rj} a_{ik}}{a_{rk}}, \quad b'_i = b_i - \frac{b_r a_{ik}}{a_{rk}}, \quad i \neq r, \quad (2.20)$$

где  $a_{rk} (\neq 0)$  – разрешающий элемент, относительно переменной  $x_k$  которого выполняются преобразования по формулам (2.19) и (2.20);  $a'_{ij}, b'_i$  – новые значения соответствующих коэффициентов и свободных членов системы уравнений.

В результате выполнения  $m$  итераций получим следующую систему уравнений, эквивалентную исходной системе (2.18):

$$x_i + \sum_{j=m+1}^n \alpha_{ij} x_j = \beta_i, \quad i = \overline{1, m},$$

где  $\alpha_{ij} = a_{ij}^{(m)}$ ,  $\beta_i = b_i^{(m)}$  – коэффициенты системы уравнений после  $m$  итераций.

Отсюда следует, что общее решение системы уравнений (2.18) имеет вид:

$$x_i = \beta_i - \sum_{j=m+1}^n \alpha_{ij} x_j, \quad i = \overline{1, m}. \quad (2.21)$$

Переменные  $x_1, \dots, x_m$ , относительно которых решена система, называются *базисными*, а остальные переменные  $x_{m+1}, \dots, x_n$  – *свободными*.

Введем некоторые понятия.

– частное решение системы (2.18), соответствующее нулевым значениям свободных переменных, называется *базисным решением* этой системы;

– если базисное решение системы (2.18) содержит более чем  $n - m$  нулевых компонент (т.е. хотя бы одна базисная переменная обращается в нуль), то это решение называется *вырожденным*, в противном случае – *невырожденным*.

Например, общему решению (2.21) соответствует базисное решение  $x = (\beta_1, \dots, \beta_m, 0, \dots, 0)$ . Если  $\beta_i \neq 0$ ,  $i = \overline{1, m}$ , то решение невырожденное.

Отметим, что каждому выбору  $m$  базисных переменных системы (2.18) соответствует свое базисное решение. Поэтому число базисных решений равно числу всевозможных базисных миноров матрицы  $A$ , т.е. не превосходит числа сочетаний  $C_n^m$ .

### Пример 2.5

Методом Жордана-Гаусса решить следующую систему уравнений:

$$\begin{cases} x_1 - 3x_2 + x_3 - 4x_4 = 2, \\ x_1 + x_2 + x_3 + 2x_4 = 6, \\ 2x_1 - x_2 + x_3 + x_4 = -3, \\ 4x_1 - 3x_2 + 3x_3 - x_4 = 5. \end{cases}$$

**Решение.** Каждая итерация *алгоритма* решения системы уравнений включает следующие шаги:

1. Выбор ненулевого элемента  $a_{rk}$  матрицы  $A$ , который называется *разрешающим элементом*. Строка и столбец, на пересечении которых находится этот элемент называют соответственно разрешающими.

2. По формулам (2.19) пересчитывается разрешающая строка.

3. Все остальные элементы таблицы пересчитываются согласно формулам (2.20). Этим формулам соответствует так называемое *правило прямоугольника*: на разрешающем  $a_{rk}$  и преобразуемом  $a_{ij}$  элементах, как вершинах главной диагонали, мысленно строится прямоуголь-

ник, т.е. строится прямоугольник с вершинами  $a_{ij}$ ,  $a_{ik}$ ,  $a_{rk}$  и  $a_{rj}$ . Подстановка этих элементов в (2.20) позволяет получить значение  $a'_{ij}$ .

Решение системы выполняется в таблицах с использованием формул (2.19) и (2.20). В каждой таблице клетка с разрешающим элементом выделена двойной линией. На каждой итерации появляется новая базисная переменная. Процесс решения приведен в таблицах 2.1.

Таблицы 2.1

Баз.пер.	$x_1$	$x_2$	$x_3$	$x_4$	$b_i$
-	1	-3	1	-4	2
-	1	1	1	2	6
-	2	-1	1	1	-3
-	4	-3	3	-1	5

Баз.пер.	$x_1$	$x_2$	$x_3$	$x_4$	$b_i$
-	0	-4	0	-6	-4
$x_1$	1	1	1	2	6
-	0	-3	-1	-3	-15
-	0	-7	-1	-9	-19

Баз.пер.	$x_1$	$x_2$	$x_3$	$x_4$	$b_i$
-	0	-4	0	-6	-4
$x_1$	1	-2	0	-1	-9
$x_3$	0	3	1	3	15
-	0	-4	0	-6	-4

Баз.пер.	$x_1$	$x_2$	$x_3$	$x_4$	$b_i$
$x_2$	0	1	0	1.5	1
$x_1$	1	0	0	2	-7
$x_3$	0	0	1	-1.5	12
-	0	0	0	0	0

Итак, система уравнений линейно зависима (последняя строка последней таблицы заполнилась нулями), неопределенная.

Общее решение системы:

$$x_1 = -7 - 2x_4; \quad x_2 = 1 - 1.5x_4; \quad x_3 = 12 + 1.5x_4.$$

Базисные переменные  $x_1, x_2, x_3$  образуют единичный базис,  $x_4$  – свободная переменная.

Для проверки правильности решения системы уравнений *общее решение* подставляется в левую часть каждого уравнения системы. Получаются тождества.

Следующая теорема является основной для симплекс-метода. Согласно этой теореме, если решение задачи линейного программирования существует, то оно достигается хотя бы в одной из угловых точек (вершин) допустимого множества ( $n$ -мерного многогранника)  $U \subset R^n$ .

**Теорема 2.4.** (Об экстремальной точке).

1) *Линейная функция достигает своего максимального значения в угловой точке многогранника решений  $U \subset R^n$ .*

2) *Если максимум достигается более чем в одной крайней точке, то он достигается и в любой точке, являющейся выпуклой комбинацией этих крайних точек.*

**Доказательство**

1. Пусть  $x^{(1)}, x^{(2)}, \dots, x^{(k)} \in U \subset R^n$  крайние точки и  $x^* \in U$  – оптимальное решение. Таким образом, для любой точки  $x \in U$   $f(x^*) \geq f(x)$ . Предположим, что  $x^*$  – не крайняя точка. Тогда  $x^* = \sum_{i=1}^k \lambda_i x^{(i)}$ ,  $\sum_{i=1}^k \lambda_i = 1$ ,  $\lambda_i \geq 0$ ,  $i = \overline{1, k}$ . Следовательно, значение целевой функции:

$$\begin{aligned} f(x^*) &= f\left(\sum_{i=1}^k \lambda_i x^{(i)}\right) = f(\lambda_1 x^{(1)}) + f(\lambda_2 x^{(2)}) + \dots + f(\lambda_k x^{(k)}) = \\ &= \lambda_1 f(x^{(1)}) + \lambda_2 f(x^{(2)}) + \dots + \lambda_k f(x^{(k)}). \end{aligned}$$

Выбираем наибольшее значение из  $f(x^{(i)})$ . Пусть наибольшее  $f(x^{(p)}) = M$ ,  $1 \leq p \leq k$ . Заменим все  $f(x^{(i)})$  на  $M$ . Тогда  $f(x^*) \leq \lambda_1 M + \lambda_2 M + \dots + \lambda_k M = M \sum_{i=1}^k \lambda_i = M$ . С другой стороны,  $f(x^*) \geq M$ , т.к.  $x^*$  – оптимальное решение. Следовательно,  $f(x^*) = M = f(x^{(p)})$ , т.е.  $x^* = x^{(p)}$ , но  $x^{(p)}$  – угловая точка.

2. Пусть максимум достигается в крайних точках  $x^{(1)}, x^{(2)}, \dots, x^{(q)}$ . Тогда  $f(x^{(1)}) = f(x^{(2)}) = \dots = f(x^{(q)}) = M$ . Любой точке  $x^* = \sum_{i=1}^q \lambda_i x^{(i)}$ ,  $\sum_{i=1}^q \lambda_i = 1$ ,  $\lambda_i \geq 0$ ,  $i = \overline{1, q}$ , являющейся выпуклой

комбинацией крайних точек, соответствует значению целевой функции:

$$f(x^*) = f\left(\sum_{i=1}^q \lambda_i x^{(i)}\right) = \lambda_1 f(x^{(1)}) + \lambda_2 f(x^{(2)}) + \dots + \lambda_k f(x^{(q)}) = \\ = M \sum_{i=1}^q \lambda_i = M. \text{ Следовательно, точка } x^* \text{ является точкой макси-}$$

мального значения функции.

Предположим, что система уравнений (2.6) канонической формы математической модели задачи линейного программирования разрешена относительно  $x_i \geq 0, i = \overline{1, m}$ :

$$x_i + \sum_{j=m+1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m}; \quad (2.22)$$

$$x_i \geq 0, \quad i = \overline{1, m}; \quad x_j \geq 0, \quad j = \overline{m+1, n}, \quad (2.23)$$

где  $x_i$  – базисные переменные и  $x_j$  – свободные переменные системы (2.22).

**Определение 2.19.** *Опорным планом* задачи линейного программирования в канонической форме называется неотрицательное базисное решение системы уравнений (2.22).

**Определение 2.20.** *Оптимальным опорным планом* называется опорный план, при котором целевая функция  $f(x)$  достигает экстремального значения.

**Определение 2.21.** *Опорный план называется невырожденным*, если число его положительных компонент равно  $m$ . Задача при этом называется *невырожденной*. В противном случае опорный план и задача называются *вырожденными*.

Представим систему уравнений (2.22) в векторной форме:

$$e^{(1)}x_1 + \dots + e^{(m)}x_m + a^{(m+1)}x_{m+1} + \dots + a^{(n)}x_n = b, \quad (2.24)$$

где  $e^{(1)}, \dots, e^{(m)}$  – единичные векторы-столбцы базисных переменных,  $a^{(m+1)}, \dots, a^{(n)}$  – векторы-столбцы коэффициентов свободных переменных,  $b$  – вектор-столбец с элементами  $b_i \geq 0, i = \overline{1, m}$ .

**Теорема 2.5.** *(Об опорном плане).* *Опорному плану*  $x = (b_1, b_2, \dots, \dots, b_m, 0, \dots, 0)$  *разложения (2.24) соответствует угловая точка многогранника решений*  $U \subset R^n$ .

**Доказательство.** Предположим, что точка  $x$  не является угловой. Тогда она может быть представлена как выпуклая комбинация двух других точек  $x^{(1)}$  и  $x^{(2)}$  многогранника решений  $X$ , т.е.

$$x = \lambda_1 x^{(1)} + \lambda_2 x^{(2)}, \quad \lambda_1 + \lambda_2 = 1, \quad \lambda_1 \geq 0, \quad \lambda_2 \geq 0.$$

Так как компоненты векторов  $x^{(1)}$  и  $x^{(2)}$  неотрицательны,  $\lambda_1 \geq 0$  и  $\lambda_2 \geq 0$ , а последние  $(n - m)$  компонент вектора  $x$  равны нулю, то соответствующие  $(n - m)$  компонент векторов  $x^{(1)}$  и  $x^{(2)}$  также должны быть равны нулю, т.е.

$$x^{(1)} = (\alpha_1, \alpha_2, \dots, \alpha_m, 0, \dots, 0),$$

$$x^{(2)} = (\beta_1, \beta_2, \dots, \beta_m, 0, \dots, 0).$$

Поскольку  $x^{(1)}$  и  $x^{(2)}$  допустимые решения, то

$$e^{(1)}\alpha_1 + e^{(2)}\alpha_2 + \dots + e^{(m)}\alpha_m = b,$$

$$e^{(1)}\beta_1 + e^{(2)}\beta_2 + \dots + e^{(m)}\beta_m = b.$$

Вычитая из первого соотношения второе, получаем

$$e^{(1)}(\alpha_1 - \beta_1) + e^{(2)}(\alpha_2 - \beta_2) + \dots + e^{(m)}(\alpha_m - \beta_m) = 0.$$

По условию система векторов  $e^{(1)}, e^{(2)}, \dots, e^{(m)}$  линейно независимая, поэтому последнее соотношение выполняется, если  $\alpha_1 - \beta_1 = 0$ , ...,  $\alpha_m - \beta_m = 0$ , или  $\alpha_1 = \beta_1, \dots, \alpha_m = \beta_m$ . Таким образом, опорный план  $x$  невозможно представить как выпуклую комбинацию других двух точек многогранника решений. Следовательно,  $x$  – угловая точка.

Учитывая то, что экстремальное решение достигается в угловой точке многогранника решений (теорема 2.4) и опорному плану соответствует угловая точка (теорема 2.5), можно сделать следующий вывод: *оптимальный опорный план можно найти, перебрав все угловые точки многогранника решений.* Симплекс–метод дает ответ на вопрос, как найти оптимальный опорный план, не прибегая к процедуре полного перебора опорных планов.

### 2.3. Симплекс-метод

Рассмотрим наиболее распространенную схему алгоритма решения задачи линейного программирования – *симплекс-метод*.

По существу, симплекс–метод представляет собой последовательный перебор угловых точек многогранного допустимого множества, при котором значение целевой функции последовательно улучша-

ется (при поиске максимума – увеличивается). Направление поиска оптимального решения и окончание поиска, если оптимальное решение найдено, в симплекс-методе определяют *оценки оптимальности*.

### Оценки оптимальности опорного плана

Симплекс-метод применим к канонической форме математической модели. Предположим, что задача линейного программирования (2.5) – (2.7) невырожденная и разрешена относительно  $m$  первых переменных. Такое предположение несколько не умаляет общности получаемых результатов. Тогда математическая модель (2.5) – (2.7) принимает следующий вид:

$$f(x) = \sum_{i=1}^m c_i x_i + \sum_{j=m+1}^n c_j x_j \rightarrow \max ; \quad (2.25)$$

$$x_i + \sum_{j=m+1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m}; \quad (2.26)$$

$$x_i \geq 0, \quad i = \overline{1, m}; \quad x_j \geq 0, \quad j = \overline{m+1, n}. \quad (2.27)$$

Общее решение системы уравнений (2.26)

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

подставим в (2.25). Тогда:

$$\begin{aligned} f(x) &= \sum_{i=1}^m c_i \left( b_i - \sum_{j=m+1}^n a_{ij} x_j \right) + \sum_{j=m+1}^n c_j x_j = \\ &= \sum_{i=1}^m c_i b_i - \sum_{j=m+1}^n x_j \left( \sum_{i=1}^m c_i a_{ij} - c_j \right). \end{aligned}$$

Введем обозначение:

$$\Delta_j = \sum_{i=1}^m c_i a_{ij} - c_j, \quad j = \overline{m+1, n}. \quad (2.28)$$

Величины  $\Delta_j$ ,  $j = \overline{m+1, n}$ , соответствующие свободным переменным  $x_j$ , называются *оценками оптимальности* опорного плана  $x$ .

Отметим, что для базисных переменных  $x_l$   $l = \overline{1, m}$  выполняется равенство  $\Delta_l = \sum_{i=1}^m c_i e_i^{(l)} - c_l = c_l - c_l = 0$ ,  $l = \overline{1, m}$ , где (согласно (2.26)):

$$e_i^{(l)} = \begin{cases} 1, & \text{при } i = l, \quad c_i = c_i; \\ 0, & \text{при } i \neq l, \quad c_i \neq c_i. \end{cases}$$

С учетом введенного обозначения:

$$f(x) = \sum_{i=1}^m c_i b_i - \sum_{j=m+1}^n \Delta_j x_j. \quad (2.29)$$

Формула (2.29) определяет новое значение целевой функции в результате преобразования однократного замещения (т.е. замены одной из базисных переменных свободной переменной).

Например, если в базис опорного плана  $x = (b_1, \dots, b_m, 0, \dots, 0)$  с целевой функцией  $f(x)$  войдет переменная  $x_k$ ,  $k \geq m+1$  вместо  $x_r$ , то новому опорному плану  $x' = (b_1, \dots, 0_r, \dots, b_m, 0, \dots, b_k, \dots, 0)$  будет соответствовать следующее значение целевой функции:

$$f(x') = \sum_{i=1}^m c_i b_i - \Delta_k b_k = f(x) - \Delta_k b_k = f(x) - \Delta_k b_k.$$

В основе симплекс-метода лежат эквивалентные преобразования системы уравнений, обеспечивающие переход от одного *опорного плана* к другому, лучшему с точки зрения значения целевой функции.

Для применения симплекс-метода задача должна быть приведена в каноническую форму с *исходным опорным планом*.

Пусть каноническая форма математической модели задачи линейного программирования имеет следующий вид (2.25) – (2.27), где  $x_i$  – базисные переменные, а  $x_j$  – свободные переменные. Полагаем, что  $b_i > 0$ ,  $i = \overline{1, m}$ . Поэтому базисное решение  $x = (b_1, \dots, b_m, 0, \dots, 0)$  является *невырожденным опорным планом*.

**Алгоритм симплекс-метода** при поиске *максимума* целевой функции включает следующие шаги:

1. Построение канонической формы математической модели с помощью дополнительных неотрицательных переменных.
2. Построение исходного опорного плана (если его нет) с помощью *искусственных базисных переменных*.
3. Построение исходной симплекс-таблицы с оценками оптимальности, вычисленными по формуле (2.28) и значением целевой

функции  $f(x) = \sum_{i=1}^m c_i b_i$ .

4. Выбор разрешающего столбца  $k$  по наименьшей отрицательной оценке свободной переменной:

$$\Delta_k = \min\{\Delta_j < 0, j = \overline{m+1, n}\}.$$

5. Выбор разрешающей строки  $r$  согласно условию:

$$a_{rk} = \min\left\{\frac{b_i}{a_{ik}}, a_{ik} > 0, i = \overline{1, m}\right\}.$$

6. Преобразование симплекс-таблицы по формулам Жордана-Гаусса (2.19), (2.20) относительно разрешающего элемента  $a_{rk}$ . Получается новый опорный план  $x'$ , для которого  $f(x') \geq f(x)$ . Переход к шагу 4.

В следующей теореме содержатся признаки возможных решений, которые дополняют алгоритм симплекс-метода и используются при решении задачи.

**Теорема 2.6.** (Основная теорема симплекс-метода).

1) Опорный план  $x^* = (b_1, \dots, b_m, 0, \dots, 0)$  задачи (2.25) - (2.27) является оптимальным, если все оценки  $\Delta_j \geq 0, j = \overline{1, n}$ . (Признак оптимальности опорного плана).

2) Если  $x^*$  – оптимальный опорный план и хотя бы одна оценка свободной переменной  $\Delta_j = 0$ , то этот план не единственный. (Признак неоднозначности оптимального решения).

3) Если найдется такой номер  $k \geq m+1$ , для которого оценка  $\Delta_k < 0$  и все  $a_{ik} \leq 0, i = \overline{1, m}$ , то целевая функция  $f_{\max} \rightarrow \infty$  на множестве допустимых решений  $U \subset R^n$ . (Признак неограниченности области допустимых решений).

4) Если для опорного плана  $x$  найдется такой номер  $k \geq m+1$ , для которого оценка  $\Delta_k < 0$  и хотя бы один элемент  $a_{ik} > 0, i = \overline{1, m}$ , то существует такой новый опорный план  $x'$ , которому соответствует  $f(x') > f(x)$ . (Признак улучшения опорного плана).

**Доказательство**

1) Пусть вектор  $x^* = (b_1, \dots, b_m, 0, \dots, 0)$  является оптимальным опорным планом, тогда  $f(x^*) = \langle c, x^* \rangle$ . Для любого другого смежного опорного плана  $x$  значение целевой функции определяется по формуле (2.29), т.е.  $f(x) = f(x^*) - \sum_{j=m+1}^n \Delta_j x_j$ . Поскольку  $\Delta_j \geq 0$ ,

$j = \overline{1, n}$ , то отсюда следует, что  $f(x^*) \geq f(x)$ . Следовательно,  $x^*$  – точка максимума целевой функции.

2) Пусть  $x^*$  – оптимальное решение задачи (2.25) – (2.27) и для некоторого  $k \geq m+1$  оценка  $\Delta_k = 0$ . Тогда, как и в предыдущем случае  $f(x) = f(x^*) - \Delta_k x_k = f(x^*)$ . Следовательно,  $x$  также оптимальный опорный план.

3) Для доказательства этой части теоремы введем в рассмотрение вектор  $x(\lambda)$ ,  $\lambda > 0$  в следующем виде:

$$x_i(\lambda) = \begin{cases} b_i - \lambda a_{ik}, & i = \overline{1, m}; \\ \lambda, & i = k; \\ 0, & i = \overline{m+1, n}, i \neq k, \end{cases}$$

где по условию теоремы оценка  $\Delta_k < 0$  и  $a_{ik} \leq 0$  для всех  $i = \overline{1, m}$ .

Очевидно, что все компоненты  $x_i(\lambda) \geq 0$ , т.к.  $\lambda > 0$  и  $a_{ik} \leq 0$ . Подставим  $x(\lambda)$  в систему уравнений (2.26). Получим следующие тождества:

$$\begin{cases} b_1 - \lambda a_{1k} & + a_{1k} \lambda = b_1, \\ b_2 - \lambda a_{2k} & + a_{2k} \lambda = b_2, \\ \dots & \dots \\ b_m - \lambda a_{mk} & + a_{mk} \lambda = b_m. \end{cases} \quad \text{или} \quad \begin{cases} b_1 \equiv b_1, \\ b_2 \equiv b_2, \\ \dots \\ b_m \equiv b_m. \end{cases}$$

Следовательно,  $x(\lambda)$  – допустимое решение.

С другой стороны, подставив  $x(\lambda)$  в целевую функцию (2.25), получим:

$$f(x) = \sum_{i=1}^m c_i b_i - \lambda \sum_{i=1}^m c_i a_{ik} + \lambda c_k = \sum_{i=1}^m c_i b_i - \lambda \left( \sum_{i=1}^m c_i a_{ik} - c_k \right),$$

где  $\sum_{i=1}^m c_i a_{ik} - c_k = \Delta_k < 0$  (по условию теоремы).

Значение  $\lambda > 0$  может быть выбрано сколь угодно большим. Но при  $\lambda \rightarrow \infty$  имеем  $f(x(\lambda)) = f_{\max} \rightarrow \infty$ . Следовательно, область допустимых решений  $U$  не ограничена сверху.

4) Пусть найдутся такие  $k \geq m+1$  и  $i \leq m$ , что  $\Delta_k < 0$  и  $a_{ik} > 0$ . Выберем  $\lambda_r = \min\{b_i/a_{ik} \mid a_{ik} > 0, i = \overline{1, m}\}$ . При таком выборе  $\lambda_r$  вектор  $x(\lambda_r)$  является допустимым. Действительно, новое решение

$$x'_i = b_i - \lambda_r a_{ik} = b_i - \frac{b_r}{a_{rk}} a_{ik} = a_{ik} \left( \frac{b_i}{a_{ik}} - \frac{b_r}{a_{rk}} \right)$$

является опорным планом, т.к.

- при  $a_{ik} < 0$  очевидно, что  $x'_i > 0$ ;
- при  $a_{ik} > 0$  имеет место условие  $b_i/a_{ik} > b_r/a_{rk}$ , то  $x'_i > 0$ .

Следовательно,  $x' \in U$  и  $f(x') = f(x) - \lambda_r \Delta_k > f(x)$ , т.к.  $\Delta_k < 0$ , а  $\lambda_r > 0$  (по построению). Таким образом, новый опорный план  $x'$  лучше опорного плана  $x$ :  $f(x') > f(x)$ .

### Пример 2.6

Решить следующую задачу симплекс-методом:

$$f(x) = 4x_1 + 8x_2 - 2x_3 - 4x_4 \rightarrow \max;$$

$$\begin{cases} -x_1 + 2x_2 + x_3 = 2, \\ x_1 - x_2 + x_4 = 2, \\ x_1 \leq 4; \end{cases} \quad x_j \geq 0, \quad j = \overline{1, 4}.$$

Дать геометрическую интерпретацию решения, т.е. решить задачу графическим методом и сравнить с результатами решения симплекс-методом.

### Решение

Строится каноническая форма. Для этого вводится дополнительная переменная  $x_5 \geq 0$ :

$$f(x) = 4x_1 + 8x_2 - 2x_3 - 4x_4 \rightarrow \max,$$

$$\begin{cases} -x_1 + 2x_2 + x_3 = 2, \\ x_1 - x_2 + x_4 = 2, \\ x_1 + x_5 = 4; \end{cases} \quad x_j \geq 0, \quad j = \overline{1, 5}.$$

Решение выполняется в соответствии с алгоритмом симплекс-метода в табличной форме (см. табл. 2.2–2.4). Клетка с разрешающим элементом в каждой таблице выделена двойной линией.

Таблица 2.2

Баз. Пер.	$c_i$	$b_i$	4	8	-2	-4	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_3$	-2	2	-1	2	1	0	0
$x_4$	-4	2	1	-1	0	1	0
$x_5$	0	4	1	0	0	0	1
$f$	-	-12	-6	-8	0	0	0

Исходный опорный план  $x = (0, 0, 2, 2, 4)$ ;  $f = -12$ . Решение не оптимально, т.к. имеются отрицательные оценки. По наименьшей отрицательной оценке ( $\Delta_2 = -8$ ) выбирается разрешающий столбец коэффициентов переменной  $x_2$ . Выбирается первая строка в качестве разрешающей (единственный положительный элемент  $a_{12} > 0$ ). Таким образом, разрешающий элемент  $a_{rk} = a_{12} = 2$ . В результате преобразований по формулам Жордана-Гаусса получаем таблицу 2.3. Опорный план  $x = (0, 1, 0, 3, 4)$ ,  $f = -4$ . Имеется отрицательная оценка  $\Delta_1 = -10$ . В соответствии с шагами 4 и 5 алгоритма выбирается разрешающий элемент  $a_{rk} = a_{31} = 1$ , относительно которого выполняются преобразования табл. 2.3.

Таблица 2.3

Баз. Пер.	$c_i$	$b_i$	4	8	-2	-4	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_2$	8	1	-0.5	1	0.5	0	0
$x_4$	-4	3	0.5	0	0.5	1	0
$x_5$	0	4	1	0	0	0	1
$f$	-	-4	-10	0	4	0	0

Результат преобразования таблицы 2.3 представлен в таблице 2.4.

Таблица 2.4

Баз. пер.	$c_i$	$b_i$	4	8	-2	-4	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_2$	8	3	0	1	0.5	0	0
$x_4$	-4	1	0	0	0.5	1	-0.5
$x_1$	4	4	1	0	0	0	1
$f$	-	36	0	0	4	0	10

Опорный план  $x^* = (4, 3, 0, 1, 0)$  является оптимальным, т.к. нет отрицательных оценок. Значение целевой функции  $f_{\max} = f(x^*) = 36$ . Решим задачу *графическим методом*, используя геометрическую интерпретацию элементов модели. Наглядность интерпретации достигается только для случая двух переменных  $x_1$  и  $x_2$ . Поэтому для уменьшения размерности исходной задачи приводим ее в стандартную форму, отбрасывая неотрицательные базисные переменные  $x_3$  и  $x_4$  и исключая их из целевой функции. Переменную  $x_3 = x_1 - 2x_2 + 2$  из первого и переменную  $x_4 = -x_1 + x_2 + 2$  из второго уравнений системы ограничений подставляем в целевую функцию. Исходная задача в стандартной форме будет иметь следующий вид:

$$f(x) = 6x_1 + 8x_2 - 12 \rightarrow \max ;$$

$$\begin{cases} -x_1 + 2x_2 \leq 2, & (1) \\ x_1 - x_2 \leq 2, & (2) \\ x_1 \leq 4; & (3) \\ x_1 \geq 0, \quad x_2 \geq 0. \end{cases}$$

Каждое неравенство системы ограничений на плоскости в декартовой прямоугольной системе координат переменных  $x_1$  и  $x_2$  определяет некоторую полуплоскость. Пересечение полуплоскостей системы ограничений и условий неотрицательности образует *область допустимых решений (ОДР)* задачи. На рис. 2.2 область допустимых решений (ограниченный многоугольник) выделена внутренней штриховкой.

Целевая функция  $f(x_1, x_2) = c_1x_1 + c_2x_2$  графически изображается множеством параллельных прямых, называемых *линиями уровня*, каждой линии соответствует определенное значение целевой функции.

Вектор градиента целевой функции

$$f'_x = \text{grad } f = \left( \frac{\partial f(x_1, x_2)}{\partial x_1}, \frac{\partial f(x_1, x_2)}{\partial x_2} \right) = (c_1, c_2)$$

обладает следующими свойствами:

1) Направление вектора градиента определяет направление роста целевой функции.

2) Вектор градиента перпендикулярен линиям уровня целевой функции.

3) Модуль вектора градиента определяет скорость роста целевой функции в точке его построения.

*Алгоритм графического метода* решения задачи линейного программирования основан на этих свойствах линейной функции и содержит следующие шаги:

1. Построение области допустимых решений задачи линейного программирования.

2. Построение вектора градиента целевой функции и построение хотя бы одной линии уровня перпендикулярно вектору градиента.

3. Перемещение линии уровня в пределах области допустимых решений параллельно самой себе. При поиске максимума  $f(x)$  линия уровня перемещается в направлении вектора градиента до *крайней точки* области. При поиске минимума  $f(x)$  линия уровня перемещается в направлении противоположном направлению вектора градиента до *крайней точки* области. Координаты крайней точки определяют оптимальное решение задачи.

На основании неравенств (1), (2), (3) системы ограничений и условий неотрицательности задачи строится ОДР (рис. 2.2).

Вектор градиента целевой функции исходной задачи в стандартной форме имеет вид  $f'_x = \text{grad } f(x_1, x_2) = (6, 8)$ . Из начала координат в произвольном масштабе он построен на рис. 2.2. Проводится линия уровня целевой функции, например, через точку  $x = (0, 0)$  перпендикулярно вектору  $f'_x$ . Этой линии уровня соответствует  $f(x) = -12$ .

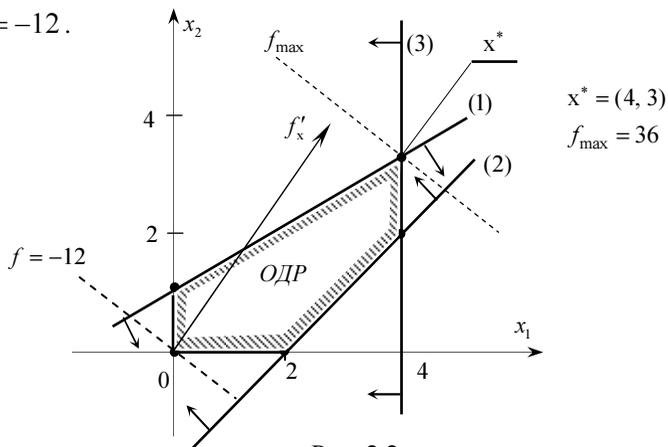


Рис. 2.2

Согласно построениям, крайней точкой ОДР, которой соответствует максимум целевой функции  $f(x)$ , является точка  $A$ . Точка  $A$  образована пересечением прямых (1) и (3). Таким образом, оптимальное решение задачи:  $x^* = (4, 3)$ ,  $f_{\max} = f(x^*) = 36$  (рис. 2.2).

Из сравнения решений симплекс–методом и графическим методом можно видеть, что каждой итерации симплекс–метода соответствует переход решения из одной вершины многоугольника в другую смежную ей вершину с увеличивающимся значением целевой функции.

Не приводя строгих доказательств и не рассматривая дополнительных примеров, укажем возможные случаи, которые могут встретиться при построении ОДР:

- получающаяся область имеет вид ограниченного выпуклого многоугольника (пример 2.6). В многомерном пространстве – ограниченный выпуклый многогранник;
- получается неограниченный выпуклый многоугольник (многогранник). Подобная ситуация, например, получится, если в примере 2.2 убрать ограничение (1). Оставшаяся часть области будет неограниченным выпуклым многоугольником;
- наконец, возможен случай, когда неравенства задачи противоречат друг другу. Тогда ОДР – *пустая*. Если в примере 2.2 изменить знаки неравенств (1) и (2) на  $(\geq)$ , то система ограничений становится противоречивой, не имеющей допустимых решений.

### Симплекс–метод с искусственным базисом

Задача линейного программирования в канонической форме может не иметь исходного опорного плана. В этом случае для построения опорного плана используются *искусственные базисные переменные*, в результате чего решается так называемая *вспомогательная* задача, которая не является эквивалентной *исходной* задаче.

Пусть следующая исходная задача в канонической форме не имеет опорного плана (нет ни одной неотрицательной базисной переменной):

$$f = \sum_{j=1}^n c_j x_j \rightarrow \max ; \quad (2.30)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m}; \quad (2.31)$$

$$x_j \geq 0, \quad j = \overline{1, n}; \quad b_i > 0, \quad i = \overline{1, m}. \quad (2.32)$$

Тогда строится вспомогательная задача следующего вида:

$$\bar{f}(x) = \sum_{j=1}^n c_j x_j - M \sum_{i=1}^m x_{n+i} \rightarrow \max ; \quad (2.33)$$

$$\sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i, \quad i = \overline{1, m}; \quad (2.34)$$

$$x_j \geq 0, \quad j = \overline{1, n+m}, \quad (2.35)$$

где  $M$  – некоторое сколь угодно большое число (*штрафной коэффициент*);  $x_{n+i} > 0$ ,  $i = \overline{1, m}$  – искусственные базисные переменные.

Вспомогательную задачу (2.33) – (2.35) часто называют *M-задачей*, которая решается уже рассмотренным симплекс-методом.

Следующая теорема дополняет алгоритм симплекс-метода еще одним условием (признаком), связанным с использованием искусственных базисных переменных.

**Теорема 2.7.** (*Основная теорема M-задачи*). Если в оптимальном опорном плане  $\bar{x}^* = (x_1^*, x_2^*, \dots, x_n^*, 0, \dots, 0)$  вспомогательной задачи все искусственные переменные  $x_{n+i}^* = 0$ ,  $i = \overline{1, m}$ , то опорный план  $x^* = (x_1^*, x_2^*, \dots, x_n^*)$  является оптимальным решением исходной задачи.

*Доказательство.* Пусть  $\bar{x}^* = (x_1^*, x_2^*, \dots, x_n^*, 0, \dots, 0)$  – оптимальный опорный план вспомогательной задачи (2.33) – (2.35). Тогда вектор  $x^* = (x_1^*, x_2^*, \dots, x_n^*)$  является допустимым решением исходной задачи (2.30) – (2.32). При этом  $\bar{f}(\bar{x}^*) = f(x^*)$ , т.к.  $\bar{x}^*$  отличается от  $x^*$  только компонентами, равными нулю.

Докажем, что  $x^*$  – оптимальный план исходной задачи. Предположим, что  $x^* = (x_1^*, x_2^*, \dots, x_n^*)$  не является оптимальным планом. Тогда можно найти такой оптимальный опорный план исходной задачи  $x = (x_1, x_2, \dots, x_n)$ , для которого  $f(x) > f(x^*)$ . Отсюда соответственно для вектора  $\bar{x} = (x_1, x_2, \dots, x_n, 0, \dots, 0)$ , который является опорным планом вспомогательной задачи, получаем

$$\bar{f}(\bar{x}) = f(x) > f(x^*) = \bar{f}(\bar{x}^*), \text{ т.е. } \bar{f}(\bar{x}) > \bar{f}(\bar{x}^*).$$

Следовательно, опорный план  $\bar{x}^*$  вспомогательной задачи не является оптимальным, что противоречит условию теоремы. Таким образом, опорный план  $x^* = (x_1^*, x_2^*, \dots, x_n^*)$  является оптимальным планом исходной задачи.

**Следствие.** (Признак пустой области допустимых решений). Если оптимальный опорный план вспомогательной задачи содержит хотя бы одну искусственную переменную  $x_{n+i} > 0$ ,  $i = \overline{1, m}$ , то исходная задача не имеет допустимых решений.

### Пример 2.7

Решить задачу симплекс-методом:

$$f(x) = -16x_1 + 3x_2 + 12x_3 \rightarrow \max ;$$

$$\begin{cases} x_1 + x_2 \geq 8, \\ 4x_1 + 5x_2 + x_3 = 80, \\ x_1 - x_2 \geq 4; \\ x_j \geq 0, \quad j = 1, 2, 3. \end{cases}$$

Дать геометрическую интерпретацию решения, т.е. решить графическим методом.

### Решение

Строится каноническая форма исходной математической модели:

$$f(x) = -16x_1 + 3x_2 + 12x_3 \rightarrow \max ;$$

$$\begin{cases} x_1 + x_2 - x_4 = 8, \\ 4x_1 + 5x_2 + x_3 = 80, \\ x_1 - x_2 - x_5 = 4; \\ x_j \geq 0, \quad j = \overline{1, 5}. \end{cases}$$

Базисное решение  $x = (0, 0, 80, -8, -4)$  не является опорным планом, т.к. нарушено условие неотрицательности:  $x_4 < 0$  и  $x_5 < 0$ . Используя искусственные базисные переменные, строим *вспомогательную* задачу:

$$\bar{f}(x) = -16x_1 + 3x_2 + 12x_3 - Mx_6 - Mx_7 \rightarrow \max ;$$

$$\begin{cases} x_1 + x_2 - x_4 + x_6 = 8, \\ 4x_1 + 5x_2 + x_3 = 80, \\ x_1 - x_2 - x_5 + x_7 = 4; \\ x_j \geq 0, \quad j = \overline{1,7}. \end{cases}$$

опорным планом  $x = (0, 0, 80, 0, 0, 8, 4)$  и значением целевой функции  $\bar{f} = 960 - 12M$ .

Для решения используется рассмотренный алгоритм симплекс-метода. Решение по итерациям приведено в таблицах 2.5. – 2.7.

Таблица 2.5

Баз. пер.	$c_i$	$b_i$	-16	3	12	0	0	-M	-M
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_6$	-M	8	1	1	0	-1	0	1	0
$x_3$	12	80	4	5	1	0	0	0	0
$x_7$	-M	4	1	-1	0	0	-1	0	1
$\bar{f}$	-	960	64	57	0	0	0	0	0
(M)	-	-12	-2	0	0	1	1	0	0

Таблица 2.6

Баз. пер.	$c_i$	$b_i$	-16	3	12	0	0	-M	-M
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_6$	-M	4	0	2	0	-1	1	1	-1
$x_3$	12	64	0	9	1	0	4	0	-4
$x_1$	-16	4	1	-1	0	0	-1	0	1
$\bar{f}$	-	704	0	121	0	0	64	0	-64
(M)	-	-4	0	-2	0	1	-1	0	2

Текущее решение:  $x = (4, 0, 64, 0, 0, 4, 0)$  и  $\bar{f} = 704 - 4M$ .

Таблица 2.7

Баз. пер.	$c_i$	$b_i$	-16	3	12	0	0	-M	-M
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$
$x_2$	3	2	0	1	0	-0.5	0.5	0.5	-0.5
$x_3$	12	46	0	0	1	4.5	-0.5	-4.5	0.5
$x_1$	-16	6	1	0	0	-0.5	-0.5	0.5	0.5
$\bar{f} = f$	-	462	0	0	0	60.5	3.5	-60.5	-3.5
(M)	-	0	0	0	0	0	0	1	1

Решение  $x^* = (6, 2, 46, 0, 0, 0, 0)$  и  $\bar{f}_{\max} = f_{\max} = 462$  является оптимальным решением исходной задачи, т.к. все искусственные переменные равны нулю (т.е.  $x_6 = 0$  и  $x_7 = 0$ ). Следовательно, оптимальное решение исходной задачи:  $x_1^* = 6$ ,  $x_2^* = 2$ ,  $x_3^* = 46$  и значение целевой функции  $f_{\max} = 462$ .

Решим задачу *графическим методом*, используя геометрическую интерпретацию элементов модели. Исключая  $x_3$ , строим стандартную форму исходной задачи:

$$f(x) = -64x_1 - 57x_2 + 960 \rightarrow \max ;$$

$$\begin{cases} x_1 + x_2 \geq 8, & (1) \\ 4x_1 + 5x_2 \leq 80, & (2) \\ x_1 - x_2 \geq 4; & (3) \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Используя систему ограничений (1) – (3) и условия неотрицательности, строим область допустимых решений (ОДР).

Вектор градиента целевой функции стандартной формы задачи имеет вид:  $f'_x = \text{grad } f(x_1, x_2) = (-64, -57)$ . В произвольном масштабе он построен на рис. 2.3.

Перпендикулярно вектору градиента  $f'_x$  через начало координат проводим линию уровня (пунктирная линия), которой соответствует значение  $f = 960$ .

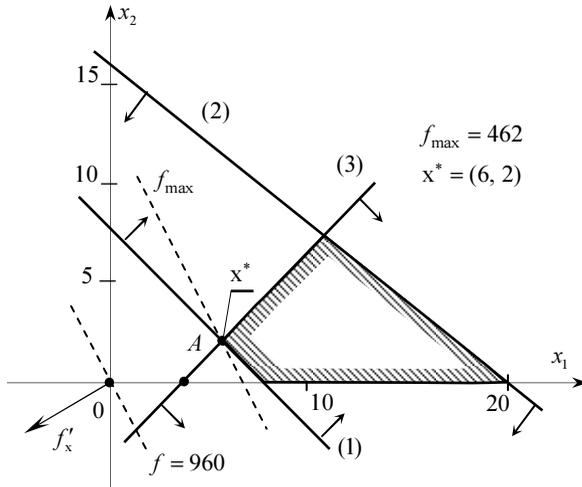


Рис. 2.3

Перемещая линию уровня в пределах области допустимых решений в направлении вектора градиента до крайней точки ОДР, получаем оптимальное решение в точке  $A$  (рис. 2.3). Эта точка образуется пересечением прямых (1) и (3). Решая соответствующую им систему уравнений:

$$\begin{cases} x_1 + x_2 = 8, & (1) \\ x_1 - x_2 = 4, & (3) \end{cases}$$

находим оптимальное решение:  $x^* = (6, 2)$ ,  $f_{\max} = f(x^*) = 462$ .

#### 2.4. Элементы теории двойственности

С каждой задачей линейного программирования (назовем ее *прямой*) связана другая задача линейного программирования, которая называется *двойственной*. Прямую и двойственную задачи принято называть *парой двойственных задач*. Двойственные задачи делятся на *симметричные* и *несимметричные*.

К симметричным двойственным задачам относятся задачи, в которых система ограничений содержит только неравенства и условие неотрицательности охватывает все переменные.

Пусть *прямая* задача имеет вид:

$$f(x) = \langle c, x \rangle \rightarrow \max, \quad x \in R^n; \quad (2.36)$$

$$Ax \leq b, \quad (2.37)$$

$$x \geq 0. \quad (2.38)$$

**Определение. 2.22.** Следующая задача поиска минимального значения целевой функции

$$g(y) = \langle b, y \rangle \rightarrow \min, \quad y \in R^m; \quad (2.39)$$

$$A^T y \geq c, \quad (2.40)$$

$$y \geq 0 \quad (2.41)$$

называется *двойственной* по отношению к прямой (2.36) – (2.38).

Из сравнения пары двойственных задач (2.36) – (2.38) и (2.39) – (2.41) следуют **правила построения двойственной задачи**:

1. Установление соответствия между целевой функцией (2.36) и системой ограничений (2.37): если  $f \rightarrow \max$ , то все неравенства приводятся к неравенствам со знаком « $\leq$ »; если  $f \rightarrow \min$  – то приводятся к неравенствам со знаком « $\geq$ ».

2. Введение двойственных переменных  $y_i \geq 0, i = \overline{1, m}$  по количеству неравенств прямой задачи. В несимметричных двойственных задачах уравнению системы ограничений одной задачи соответствует произвольная переменная двойственной задачи.

3. Построение целевой функции двойственной задачи, коэффициентами которой являются свободные члены системы ограничений прямой задачи. При этом если целевая функция прямой задачи задается на максимум, то целевая функция двойственной задачи (2.39) – на минимум, и наоборот.

4. Построение системы ограничений двойственной задачи. Главная матрица системы ограничений является транспонированной матрицей прямой задачи. Свободными членами – коэффициенты целевой функции прямой задачи.

Содержанием теории двойственности является изучение связей между взаимодвойственными задачами.

Рассмотрим основные теоремы, которые устанавливают связь между решениями прямой задачи и двойственной.

**Теорема 2.8.** (О допустимых решениях взаимодвойственных задач). Если  $x$  и  $y$  – допустимые решения прямой (2.36) - (2.38) и двойственной (2.39) - (2.41) задач, т.е.  $Ax \leq b$  и  $A^T y \geq c$ , то  $c^T x \leq b^T y$ , т.е. значения целевой функции прямой задачи не превышают значений целевой функции двойственной задачи.

**Доказательство.** Систему неравенств (2.37) умножим слева на  $y^T \geq 0$ :  $y^T Ax \leq y^T b$ . Аналогично систему (2.40) умножим слева на

$x^T \geq 0$ :  $x^T A y \geq x^T c$ . С учетом того, что матрицы  $y^T A x$  и  $x^T A y$  имеют порядок  $(1 \times 1)$ , справедливыми являются следующие соотношения:  $y^T A x = (y^T A x)^T = x^T A^T y$ . Следовательно,  $y^T b \geq y^T A x = x^T A^T y \geq x^T c$ , или  $y^T b \geq x^T c$ .

**Теорема 2.9.** (Об оптимальных решениях двойственных задач).

1. Если  $x^*$  и  $y^*$  – допустимые решения взаимодвойственных задач и  $c^T x^* = b^T y^*$ , то  $x^*$  и  $y^*$  – оптимальные решения этих задач.

2. Если целевая функция одной из пары двойственных задач не ограничена, то область допустимых решений другой задач пустая.

**Доказательство**

1. Согласно теореме 2.8 для всех допустимых решений  $x$  и  $y$  справедливо  $c^T x \leq b^T y$ , значит, и  $c^T x \leq b^T y^*$ . Тогда из условия теоремы  $c^T x^* = b^T y^*$  следует, что  $c^T x^* \geq c^T x$  и по определению оптимального решения  $x^*$  – оптимальное решение прямой задачи.

Аналогично  $c^T x^* \leq b^T y$ . Тогда, из условия  $c^T x^* = b^T y^*$  теоремы следует, что  $b^T y \geq b^T y^*$  и по определению  $y^*$  – оптимальное решение двойственной задачи.

2. Из теоремы 2.8 следует, что  $f(x) < g(y)$ , т.к. равенство  $f(x) = g(y)$  соответствует оптимальным решениям (первая часть данной теоремы).

Если  $f(x) \rightarrow \infty$ , тогда  $g(y) > \infty$ , что лишено смысла. Следовательно, двойственная задача не имеет решения.

Если  $g(y) \rightarrow -\infty$ , тогда  $f(x) < -\infty$ , что также лишено смысла. Следовательно, исходная задача не имеет решения.

**Теорема 2.10.** (О дополнительной нежесткости). Для оптимальности опорных планов  $x^*$  и  $y^*$  пары двойственных задач необходимо и достаточно, чтобы они удовлетворяли следующим системам уравнений (условиям дополнительной нежесткости):

$$y_i^* \left( \sum_{j=1}^n a_{ij} x_j^* - b_i \right) = 0, \quad i = \overline{1, m}; \quad (2.42)$$

$$x_j^* \left( \sum_{i=1}^m a_{ij} y_i^* - c_j \right) = 0, \quad j = \overline{1, n}. \quad (2.43)$$

### Доказательство

*Необходимость.* Пусть  $x^*$  и  $y^*$  – оптимальные планы взаимодвойственных задач. Покажем, что условия (2.42) и (2.43) удовлетворяются. Так как  $x^*$  – опорный план прямой задачи, то  $\sum_{j=1}^n a_{ij}x_j^* \leq b_i$ ,  $i = \overline{1, m}$ . Аналогично  $y^*$  – опорный план двойственной задачи и  $\sum_{i=1}^m a_{ij}y_i^* \geq c_j$ ,  $j = \overline{1, n}$ . Умножая первую систему неравенств на  $y_i^*$ ,  $i = \overline{1, m}$ , а вторую – на  $x_j^*$  и почленно складывая соответствующие неравенства, получим:

$$\sum_{i=1}^m y_i^* \sum_{j=1}^n a_{ij}x_j^* \leq \sum_{i=1}^m y_i^* b = g^* ; \quad (2.44)$$

$$\sum_{j=1}^n x_j^* \sum_{i=1}^m a_{ij}y_i^* \geq \sum_{j=1}^n x_j^* c_j = f^* ; \quad (2.45)$$

В соответствии с теоремой 2.9  $f^* = g^*$ . Следовательно,

$$\sum_{i=1}^m y_i^* \sum_{j=1}^n a_{ij}x_j^* \leq \sum_{i=1}^m y_i^* b = \sum_{j=1}^n x_j^* c_j \leq \sum_{j=1}^n x_j^* \sum_{i=1}^m a_{ij}y_i^* .$$

Легко заметить, что крайние члены последнего выражения – одна и та же величина  $\sum_{i=1}^m \sum_{j=1}^n a_{ij}x_j^*y_i^*$ . Тогда (2.44) и (2.45) следует рассматривать как строгие равенства:

$$\begin{aligned} \sum_{i=1}^m y_i^* \sum_{j=1}^n a_{ij}x_j^* &= \sum_{i=1}^m y_i^* b_i ; \\ \sum_{j=1}^n x_j^* \sum_{i=1}^m a_{ij}y_i^* &= \sum_{j=1}^n x_j^* c_j , \end{aligned}$$

или, соответственно:

$$\sum_{i=1}^m y_i^* \left( \sum_{j=1}^n a_{ij}x_j^* - b_i \right) = 0 ; \quad (2.46)$$

$$\sum_{j=1}^n x_j^* \left( \sum_{i=1}^m a_{ij}y_i^* - c_j \right) = 0 ; \quad (2.47)$$

В (2.46) каждое слагаемое по  $i$  отрицательное, т.к.  $y_i^* \geq 0$ , а  $\sum_{j=1}^n a_{ij}x_j^* - b_i \leq 0$ , следовательно, сумма по  $i$  равна нулю лишь при условии, что каждое слагаемое равно нулю, т.е.:

$$y_i^* \left( \sum_{j=1}^n a_{ij}x_j^* - b_i \right) = 0, \quad i = \overline{1, m}.$$

Аналогично в (2.47)  $x_j^* \geq 0$  и  $\sum_{i=1}^m a_{ij}y_i^* - c_j \geq 0$ , следовательно:

$$x_j^* \left( \sum_{i=1}^m a_{ij}y_i^* - c_j \right) = 0, \quad j = \overline{1, n}.$$

Этим доказана необходимость.

*Достаточность.* Пусть для некоторых опорных планов  $x$  и  $y$  удовлетворяются условия:

$$\sum_{j=1}^n a_{ij}x_j \leq b_i \quad \text{и} \quad \sum_{i=1}^m a_{ij}y_i \geq c_j.$$

Покажем, что эти планы оптимальны. Перепишем условия (2.42) и (2.43) в следующем виде:

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

$$x_j \sum_{i=1}^m a_{ij}y_i = c_j x_j, \quad j = \overline{1, n}.$$

Первую систему сложим почленно по  $i$ , получим:

$$\sum_{i=1}^m y_i \sum_{j=1}^n a_{ij}x_j = \sum_{i=1}^m b_i y_i = g(y).$$

Вторую систему сложим почленно по  $j$ , получим:

$$\sum_{j=1}^n x_j \sum_{i=1}^m a_{ij}y_i = \sum_{j=1}^n c_j x_j = f(x).$$

Левые части последних двух выражений равны, следовательно,  $f(x) = g(y)$ , что в соответствии с теоремой 2.9 означает оптимальность опорных планов  $x$  и  $y$ , т.е.  $x = x^*$  и  $y = y^*$ , что и требовалось доказать.

На основании теоремы 2.10 можно установить соответствие между переменными взаимодвойственных задач (2.36) – (2.38) и (2.39) –

(2.41). Для решения этих задач симплекс–методом необходимо каждую из них привести в каноническую форму с помощью введения соответствующих дополнительных неотрицательных переменных. Тогда системы ограничений каждой из взаимодвойственных задач примут вид:

$$\sum_{j=1}^n a_{ij}x_j + x_{n+i} = b_i, \quad x_{n+i} \geq 0, \quad i = \overline{1, m}; \quad (2.48)$$

$$\sum_{i=1}^m a_{ij}y_i - y_{m+j} = c_j, \quad y_{m+j} \geq 0, \quad j = \overline{1, n}. \quad (2.49)$$

**Теорема 2.11.** Положительным компонентам оптимального решения одной из взаимодвойственных задач соответствуют нулевые компоненты оптимального решения другой задачи, т.е. для любых  $i = \overline{1, m}$  и  $j = \overline{1, n}$ : если  $x_j^* > 0$ , то  $y_{m+j}^* = 0$ ; если  $x_{n+i}^* > 0$ , то  $y_i^* = 0$ ; и аналогично, если  $y_i^* > 0$ , то  $x_{n+i}^* = 0$ ; если  $y_{m+j}^* > 0$ , то  $x_j^* = 0$ .

*Доказательство.* Выразим дополнительные переменные из систем ограничений (2.48) исходной задачи и (2.49) двойственной задачи, представленных в канонической форме:

$$x_{n+i} = b_i - \sum_{j=1}^n a_{ij}x_j, \quad i = \overline{1, m}; \quad (2.50)$$

$$y_{m+j} = \sum_{i=1}^m a_{ij}y_i - c_j, \quad j = \overline{1, n}. \quad (2.51)$$

Умножая каждое равенство системы (2.50) на соответствующие переменные  $y_i \geq 0$  и складывая почленно полученные равенства, найдем:

$$\sum_{i=1}^m x_{n+i}y_i = \sum_{i=1}^m b_i y_i - \sum_{i=1}^m \sum_{j=1}^n a_{ij}x_j y_i. \quad (2.52)$$

Аналогично, умножая каждое равенство системы (2.51) на соответствующие переменные  $x_j \geq 0$  и складывая почленно полученные равенства, найдем:

$$\sum_{j=1}^n x_j y_{m+j} = \sum_{i=1}^m \sum_{j=1}^n a_{ij}x_j y_i - \sum_{j=1}^n c_j x_j. \quad (2.53)$$

Равенства (2.50) и (2.51) будут справедливы для любых допустимых значений переменных, в том числе и для оптимальных значений

$x_j^*, x_{n+i}^*, y_i^*, y_{m+j}^*$ . В силу теоремы 2.9 имеем  $\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*$ , по-

этому из записи правых частей равенств (2.52) и (2.53) следует, что они должны отличаться только знаком. С другой стороны, из неотрицательности выражений  $\sum_{i=1}^m x_{n+i} y_i$  и  $\sum_{j=1}^n x_j y_{m+j}$ , входящих в выражения (2.52) и (2.53) следует, что те же правые части этих равенств должны быть неотрицательны.

Эти условия могут выполняться одновременно только при равенстве нулю правых частей для оптимальных значений переменных :

$$\begin{cases} \sum_{i=1}^m x_{n+i}^* y_i^* = 0; \\ \sum_{j=1}^n x_j^* y_{m+j}^* = 0. \end{cases} \quad (2.54)$$

В силу условия неотрицательности переменных каждое из слагаемых в равенствах (2.54) должно равняться нулю:

$$\begin{cases} x_j^* y_{m+j}^* = 0, & j = \overline{1, n}; \\ x_{n+i}^* y_i^* = 0, & i = \overline{1, m}; \end{cases}$$

откуда следует утверждение теоремы.

Из теоремы 2.11 следует соответствие между переменными взаимодвойственных задач, т.е. можем установить пары переменных, как показано в таблице 2.8.

Таблица 2.8

Переменные исходной задачи									
Первоначальные					Дополнительные				
$x_1$	...	$x_j$	...	$x_n$	$x_{n+1}$	...	$x_{n+i}$	...	$x_{n+m}$
↓		↓		↓	↓		↓		↓
$y_{m+1}$	...	$y_{m+j}$	...	$y_{m+n}$	$y_1$	...	$y_j$	...	$y_m$
Дополнительные					Первоначальные				
Переменные двойственной задачи									

## 2.5. Двойственный симплекс-метод

Основные теоретические положения двойственности позволили построить эффективный метод решения различных задач линейного программирования, который называется *двойственным симплекс-методом*.

В двойственном симплекс-методе опорный план строится алгоритмически, поэтому искусственные базисные переменные не используются.

### *Алгоритм двойственного симплекс-метода*

1. Построение канонической формы математической модели так, чтобы коэффициенты базисных переменных в совокупности определяли единичную матрицу.

2. Построение симплекс-таблицы с соответствующими оценками оптимальности  $\Delta_j$ .

*Первый этап* (построение опорного плана):

3. Выбор разрешающей строки  $r$ . Строка выбирается по наименьшему отрицательному значению базисной переменной:

$$b_r = \min \{b_i < 0, i = \overline{1, m}\}.$$

Если все  $b_i \geq 0, i = \overline{1, m}$ , то переходим к шагу 6.

4. Выбор разрешающего столбца  $k$ . Столбец выбирается по наибольшему отношению элементов оценочной строки к отрицательным элементам разрешающей строки:

$$a_{rk} = \max \left\{ \frac{\Delta_j}{a_{rj}}, a_{rj} < 0, j = \overline{1, n} \right\}.$$

Если в разрешающей строке нет ни одного отрицательного элемента, то задача не имеет допустимых решений (*признак пустой области допустимых решений*).

5. Преобразование симплекс-таблицы по формулам Жордана-Гаусса относительно выбранного разрешающего элемента  $a_{rk}$ .

Пункты 3-5 повторяются до получения опорного плана или решение прекращается согласно признаку пустой ОДР.

*Второй этап* (поиск оптимального опорного плана):

6. Выбор разрешающего столбца по наименьшей отрицательной оценке свободной переменной:

$$\Delta_k = \min \{ \Delta_j < 0, j = \overline{m+1, n} \}.$$

7. Выбор разрешающей строки согласно условию:

$$a_{rk} = \min \left\{ \frac{b_i}{a_{ik}}, a_{ik} > 0, i = \overline{1, m} \right\}.$$

8. Преобразование симплекс-таблицы по формулам Жордана–Гаусса относительно разрешающего элемента  $a_{rk}$ . Переход к шагу 6.

Второй этап алгоритма полностью совпадает с алгоритмом симплекс-метода, рассмотренного в параграфе 2.3.

### Пример 2.8

Решить следующую задачу двойственным симплекс-методом.

$$\begin{aligned} f(x) &= 8x_1 + 2x_2 \rightarrow \max ; \\ \begin{cases} x_1 + 2x_2 \geq 4, \\ x_1 + x_2 \leq 6, \\ x_1 \geq 0, x_2 \geq 0. \end{cases} \end{aligned}$$

### Решение

Для решения задачи используем рассмотренный алгоритм двойственного симплекс-метода.

Строим каноническую форму:

$$\begin{aligned} f(x) &= 8x_1 + 2x_2 \rightarrow \max ; \\ \begin{cases} -x_1 - 2x_2 + x_3 = -4, \\ x_1 + x_2 + x_4 = 6; \\ x_j \geq 0, j = \overline{1, 4}. \end{cases} \end{aligned}$$

Заполняем симплекс-таблицу. Решение представлено в таблицах 2.8–2.10.

Таблица 2.8

Баз. пер.	$c_i$	$b_i$	8	2	0	0
			$x_1$	$x_2$	$x_3$	$x_4$
$x_3$	0	-4	-1	-2	1	0
$x_4$	0	6	1	1	0	1
$f(x)$	-	0	-8	-2	0	0

Исходное решение  $x = (0, 0, -4, 6)$ ;  $f = 0$ .

Таблица 2.9

Баз. пер.	$c_i$	$b_i$	8	2	0	0
			$x_1$	$x_2$	$x_3$	$x_4$
$x_1$	8	4	1	2	-1	0
$x_4$	0	2	0	-1	1	1
$f(x)$	-	32	0	14	-8	0

Текущее решение  $x = (4, 0, 0, 2)$ ;  $f = 32$ .

Исходное решение:

$$x = (0, 0, -2, 16, -4); y = (0, 0, 0, -2, 6); f(x) = g(y) = 0.$$

В соответствии с алгоритмом двойственного симплекс-метода разрешающая строка выбирается по наименьшему значению отрицательной базисной переменной ( $x_5 = -4$ ), а разрешающий столбец – по наибольшему отношению оценок оптимальности к отрицательным элементам разрешающей строки (т.е. первый столбец). Таким образом, разрешающий элемент  $a_{rk} = a_{31} = -1$ .

Относительно выбранного разрешающего элемента выполняем модифицированные преобразования Жордана-Гаусса. Результат преобразований представлен в таблице 2.13.

Таблица 2.13

С.п.	Б.п.	$y_3$	$y_5$	$g$
	Б.п.	$x_5$	$x_2$	
$-y_1$	$x_3$	1	-2	-6
$-y_2$	$x_4$	1	1	12
$-y_4$	$x_1$	-1	1	4
$f$		-2	8	8

Текущее решение:

$$x = (4, 0, -6, 12, 0); y = (0, 0, -2, 0, 8); f(x) = g(y) = 8.$$

Продолжаем решение. Выбираем разрешающий элемент в таблице 2.13:  $a_{rk} = a_{12} = -2$ . Выполняем преобразования. Результат преобразований – в таблице 2.14.

Таблица 2.14

С.п.	Б.п.	$y_3$	$y_1$	$g$
	Б.п. \ С.п.	$x_5$	$x_3$	
$-y_5$	$x_2$	-0.5	-0.5	3
$-y_2$	$x_4$	1.5	0.5	9
$-y_4$	$x_1$	-0.5	0.5	1
$f$		2	4	-16

В таблице 2.14 получено оптимальное решение, т.к. опорному плану исходной задачи соответствуют неотрицательные оценки:

$$x^* = (1, 3, 0, 9, 0);$$

$$y^* = (4, 0, 2, 0, 0);$$

$$f_{\max} = f(x^*) = 16.$$

$$g_{\min} = g(y^*) = -16.$$

## 2.6. Специальные задачи линейного программирования

В этом параграфе рассмотрим два типа задач линейного программирования, которые имеют специфические методы решения: транспортные задачи и параметрические задачи.

### Транспортная задача

Транспортная задача представляет собой задачу линейного программирования, ее можно решать симплекс-методом. Однако специфическая структура математической модели транспортной задачи позволила разработать более эффективные вычислительные методы.

Наиболее распространенным в практике является *метод потенциалов*. Математическая модель транспортной задачи рассмотрена в примере 2.3. Аналитическая форма модели имеет вид:

$$f(x) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min ; \quad (2.55)$$

$$\begin{cases} \sum_{j=1}^n x_{ij} = a_i, & i = \overline{1, m}; \\ \sum_{i=1}^m x_{ij} = b_j, & j = \overline{1, n}; \end{cases} \quad (2.56)$$

$$x_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}. \quad (2.57)$$

Если выполняется условие

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

то транспортная задача называется *закрытой* (замкнутой), а если это условие нарушается, то задача называется *открытой*.

**Теорема 2.12.** (О разрешимости транспортной задачи). Транспортная задача имеет решение тогда и только тогда, когда она замкнутого типа.

**Доказательство**

*Необходимость.* Суммируя первые  $m$  уравнений системы (2.56), а затем – остальные  $n$  уравнений, получим:

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} = \sum_{i=1}^m a_i ,$$

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

Сравнивая левые части последних двух равенств, получаем условие замкнутости.

*Достаточность.* Допустим, что задача замкнута, и положим

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

Нетрудно убедиться в том, что  $x_{ij} = (a_i b_j) / d$  удовлетворяют системе ограничений (2.56) и условиям (2.57). Следовательно, область допустимых решений транспортной задачи не пуста и, очевидно, ограничена в  $R^{m \times n}$ . В силу теоремы Вейерштрасса функция (2.55) достигает своего экстремума в этой области.

На практике часто встречается случай *открытой транспортной задачи*, когда, например, производство превышает потребление:

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

В этом случае ограничения по поставке и потреблению становятся неравенствами:

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad i = \overline{1, m}$$

и, соответственно:

$$\sum_{i=1}^m x_{ij} \geq b_j, \quad j = \overline{1, n} ,$$

отражая тот факт, что запасы поставщиков используются не полностью для удовлетворения всех потребностей потребителей.

Открытую транспортную задачу легко привести к закрытой введением *фиктивного* пункта потребления с объемом:

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j, \text{ при } c_{i,n+1} = 0, i = \overline{1, m}.$$

Для характеристики запасов, остающихся в пунктах поставки, необходимо ввести дополнительные переменные:  $x_{i,n+1}, i = \overline{1, m}$ .

В результате условия поставки принимают вид:

$$\sum_{j=1}^{n+1} x_{ij} = a_i, i = \overline{1, m},$$

а условия потребления:

$$\sum_{i=1}^m x_{ij} = b_j, j = \overline{1, n+1}.$$

Целевая функция при этом не меняется, поскольку стоимость поставки в фиктивный пункт равна нулю.

В равной степени возможным является обратный случай, когда потребление превышает предложение:

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

В этом случае ограничения по поставке и потреблению также становятся неравенствами:

$$\sum_{j=1}^n x_{ij} \geq a_i, i = \overline{1, m}$$

и, соответственно:

$$\sum_{i=1}^m x_{ij} \leq b_j, j = \overline{1, n},$$

отражая тот факт, что запасы поставщиков недостаточны для удовлетворения всех потребностей потребителей.

Рассуждая аналогично изложенному выше, введем фиктивный пункт поставки с объемом производства:

$$a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i, \text{ при } c_{m+1,j} = 0, j = \overline{1, n}.$$

Для характеристики недостачи в пунктах потребления необходимо ввести дополнительные переменные:  $x_{m+1,j}, j = \overline{1, n}$ .

В результате условия потребления принимают вид:

$$\sum_{i=1}^{m+1} x_{ij} = b_j, \quad j = \overline{1, n},$$

а условия поставки:

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = \overline{1, m+1}.$$

Как и в предыдущем случае, целевая функция при этом не меняется, потому что затраты на поставку из фиктивного пункта равны нулю.

Легко проверить, что система ограничений (2.56) замкнутой задачи, содержащая  $m+n$  уравнений, имеет  $(m+n-1)$  линейно независимых уравнений. Следовательно, опорный план транспортной задачи может иметь не более  $(m+n-1)$  базисных переменных. Если в опорном плане число отличных от нуля компонент равно в точности  $(m+n-1)$ , то план называется *невырожденным*, а если меньше, то *вырожденным*.

*Опорный план*  $x_{ij}$  ( $i = \overline{1, m}$ ,  $j = \overline{1, n}$ ) транспортной задачи должен обладать следующими свойствами:

- 1) Должны удовлетворяться ограничения (2.56) и (2.57).
- 2) Число базисных переменных должно равняться  $(m+n-1)$ .
- 3) В транспортной таблице (при решении методом потенциалов) опорный план должен быть *ацикличным* (это свойство будет рассмотрено в примере 2.10).

Построение алгоритма метода опирается на следующую теорему, доказательство которой мы не приводим.

**Теорема 2.12.** (*Основная теорема метода потенциалов*). *Опорный план транспортной задачи оптимален, если найдутся такие  $m$  чисел  $u_i$ ,  $i = \overline{1, m}$ , называемые потенциалами строк, и такие  $n$  чисел  $v_j$ ,  $j = \overline{1, n}$ , называемые потенциалами столбцов, что будут выполнены следующие соотношения:*

$$(1) \quad u_i + v_j = c_{ij} \quad \text{для базисных переменных } x_{ij};$$

$$(2) \quad c_{ij} - (u_i + v_j) \geq 0 \quad \text{для свободных переменных } x_{ij}.$$

Условие (2) теоремы 2.12 является признаком оптимальности опорного плана. Это условие используется для построения *оценок оптимальности*:  $s_{ij} = c_{ij} - (u_i + v_j)$ . Наличие отрицательных оценок, т.е. нарушение хотя бы одного из условий (2), говорит о том, что опорный план транспортной задачи неоптимален и его можно улуч-

шить, вводя в базис свободную переменную с отрицательной оценкой (выполняя при этом преобразование опорного плана).

**Замечание.** Как следует из теоремы 2.12, система линейных уравнений (1) содержит  $m+n$  неизвестных и  $(m+n-1)$  уравнений. Следовательно, одна из неизвестных оказывается свободной. Учитывая инвариантность оценочной матрицы транспортной задачи по отношению к частному решению системы (1), любую из неизвестных  $u_i$  (или  $v_j$ ) полагают, например, равной нулю (обычно  $u_1 = 0$ ), а затем последовательно находят все остальные.

Для решения транспортной задачи методом потенциалов используется табличная форма математической модели:

$a_i \backslash b_j$	$b_1$	...	$b_n$
$a_1$	$x_{11} \quad c_{11}$	...	$x_{1n} \quad c_{1n}$
...	...	...	...
$a_m$	$x_{m1} \quad c_{m1}$	...	$x_{mn} \quad c_{mn}$

**Алгоритм метода потенциалов :**

1. Построение исходного опорного плана. Для этого используется одно из правил (например, правило *северо-западного угла*, правило *минимальной стоимости* или др.).
2. Определение потенциалов строк  $u_i$  и столбцов  $v_j$  транспортной таблицы в соответствии с условием (1)  $u_i + v_j = c_{ij}$  для базисных переменных (загруженных клеток).
3. Вычисление оценок оптимальности  $s_{ij}$  свободных переменных по формуле  $s_{ij} = c_{ij} - (u_i + v_j)$ . Если отрицательных оценок нет, то получен оптимальный опорный план транспортной задачи. Если есть хотя бы одна отрицательная оценка, то решение можно улучшить.
4. Выбор *перспективной* клетки по наименьшей отрицательной оценке и преобразование опорного плана относительно перспективной клетки по *правилу пересчета* (правило пересчета рассмотрим в примере 2.10).
5. Проверка нового опорного плана на оптимальность, т.е. переход к шагу 2 алгоритма.

Рассмотрим правила построения исходного опорного плана.

**Правило северо-западного угла.** Заполнение транспортной таблицы начинается с левой верхней клетки переменной  $x_{11}$  и заканчивается клеткой для переменной  $x_{mn}$ . На каждом шаге остаток по строке (или по столбцу) помещается в соседнюю клетку, но так, чтобы не нарушались условия (2.56). Если на некотором шаге (но не на последнем) остаток оказался равным нулю, то в соседнюю клетку (вниз или вправо) записывается *базисный нуль* и клетка считается заполненной, а опорный план оказывается вырожденным.

**Правило минимальной стоимости.** Построение начинается с клетки с минимальной единичной стоимостью  $c_{ij}$ . В эту клетку помещается меньшее из двух чисел, связанных с ней, т.е.  $x_{ij} = \min\{a_i, b_j\}$ . Остаток по строке (или по столбцу) помещается в клетку с минимальной стоимостью без нарушения условия (2.56), и так до полного распределения запасов.

Рассмотрим применения алгоритма метода потенциалов на примере транспортной задачи закрытого типа.

**Пример 2.10.** На трех базах хранится груз в количествах:

$$a_1 = 210; a_2 = 100; a_3 = 600 \text{ (ед.)}$$

Этот груз требуется перевезти в пять пунктов назначения в количествах, равных:

$$b_1 = 150; b_2 = 160; b_3 = 200; b_4 = 100; b_5 = 300 \text{ (ед.)}$$

Тарифы перевозок единицы груза  $c_{ij}$  заданы следующей таблицей (матрицей):

$$C = \begin{pmatrix} 8 & 6 & 4 & 1 & 8 \\ 2 & 7 & 12 & 3 & 7 \\ 3 & 5 & 7 & 15 & 1 \end{pmatrix}.$$

Найти оптимальный опорный план методом потенциалов, построив исходный опорный план по правилу минимальной стоимости.

**Решение**

Исходный опорный план представлен в таблице 2.15.

Построение исходного опорного плана начинается с клетки (1,4), имеющей минимальный тариф  $c_{14} = 1$ . В эту клетку помещается  $x_{14} = \min\{a_1, b_4\} = \min\{210, 100\} = 100$ . Остаток по строке 110 помещается в клетку этой же строки с минимальными затратами  $c_{13} = 4$ , т.е.  $x_{13} = 110$  и т.д.. Распределение запасов потребителям осуществлено по

цепочке  $(1,4) \rightarrow (1,3) \rightarrow (3,3) \rightarrow (3,5) \rightarrow (3,1) \rightarrow (3,2) \rightarrow (2,2)$ . Полученному опорному плану соответствует значение целевой функции :  
 $f = 4 \cdot 110 + 1 \cdot 100 + 7 \cdot 100 + 3 \cdot 150 + 5 \cdot 60 + 7 \cdot 90 + 1 \cdot 300 = 2920$ .

Таблица 2.15

$a_i \backslash b_j$	150	160	200	100	300	$u_i$		
210	8	6	110	4	100	1	8	0
100	2 (+)	7 (-)	12	3	7	5		
600	3 (-)	5 (+)	90	7	15	300	1	3
$\}$	0	2	4	1	-2			

Условие ацикличности опорного плана заключается в невозможности соединения прямоугольным замкнутым контуром базисных клеток транспортной таблицы.

Согласно п.2 алгоритма определяются потенциалы строк и столбцов, для чего решается система уравнений:

Полагая  $u_1 = 0$ , последовательно находим:

$$\left\{ \begin{array}{l} u_1 + v_3 = 4, \\ u_1 + v_4 = 1, \\ u_3 + v_3 = 7, \\ u_3 + v_2 = 5, \\ u_2 + v_2 = 7, \\ u_3 + v_1 = 3, \\ u_3 + v_5 = 1. \end{array} \right. \quad \begin{array}{l} v_3 = 4, \quad v_4 = 1, \quad u_3 = 3, \quad v_2 = 2, \quad u_2 = 5, \quad v_1 = 0, \\ v_5 = -2. \end{array}$$

В соответствии с шагом 3 алгоритма по формуле  $s_{ij} = c_{ij} - (u_i + v_j)$  вычисляются оценки свободных клеток:  $s_{11} = 0$ ,  $s_{12} = 4$ ,  $s_{15} = 10$ ,  $s_{21} = -3$ ,  $s_{23} = 3$ ,  $s_{24} = -3$ ,  $s_{25} = 4$ ,  $s_{34} = 11$ . Так как есть отрицательные оценки, то опорный план не является оптимальным и его можно улучшить. В качестве перспективной клетки выбирается клетка с наименьшей отрицательной оценкой. Выбираем, например, клетку (2,1), т.е. свободную переменную  $x_{21}$  необходимо ввести в базис. Для этого выполним преобразование однократного замещения по следующему **правилу пересчета**:

1. Строится *цикл пересчета*: перспективная клетка соединяется прямоугольным замкнутым контуром с базисными клетками. Каждой свободной клетке соответствует единственный цикл пересчета.

2. начиная с (+) в перспективной клетке. В клетках с (-) выбирается меньшее значение базисной переменной (эта переменная в результате пересчета станет свободной).

3. На величину выбранной базисной переменной изменяются значения базисных переменных по углам контура пересчета. Эта величина добавляется в клетки с (+) и отнимается из клеток с (-). Получается новый опорный план. Если в процессе пересчета «освобождается» несколько базисных клеток, то свободной оставляют клетку с наибольшим тарифом, а остальные заполняют «базисным нулем». Опорный план при этом становится вырожденным.

В нашей задаче для выбранной клетки (2,1) строится цикл пересчета (2,1)–(2,2)–(3,2)–(3,1)–(2,1). В таблице 2.15 он показан пунктиром. Меньшее из чисел в клетках с (-) равно 100. Это число в клетки с (+) добавляется, а из клеток с (-) отнимается. Клетка (2,2) становится свободной.

В результате получился новый опорный план, который представлен в таблице 2.16. Ему соответствует следующее значение целевой функции  $f = 2620$ .

Таблица 2.16

$a_i \backslash b_j$	150	160	200	100	300	$u_i$
210	8	6	110	4	100	0
100	100	7	12	3	7	5
600	50	160	90	15	300	3
$v_j$	0	2	4	1	-2	

Следующая итерация начинается с шага 2 алгоритма метода потенциалов.

Новый опорный план проверяется на оптимальность. Аналогично определяются потенциалы строк и столбцов (таблица 2.16). По формуле  $s_{ij} = c_{ij} - (u_i + v_j)$  для каждой свободной клетки определяются оценки:  $s_{11} = 8$ ,  $s_{12} = 4$ ,  $s_{15} = 10$ ,  $s_{22} = 3$ ,  $s_{23} = 6$ ,  $s_{24} = 0$ ,  $s_{25} = 7$ ,  $s_{34} = 11$ . Отрицательных оценок нет, следовательно, полученный опорный план является оптимальным. Общая стоимость перевозок  $f_{\min} = 2620$ .

Одна из оценок свободных переменных  $s_{24} = 0$ , следовательно, оптимальное решение является *альтернативным*, т.е. неоднозначным.

Используя контур пересчета для клетки (2,4): (2,4)–(1,4)–(1,3)–(3,3)–(3,1)–(2,1)–(2,4) и вводя параметр  $t \in [0,1]$ , оптимальное решение можно записать в следующем виде:

$$x^* = \begin{pmatrix} 0 & 0 & 110 + 90t & 100 - 90t & 0 \\ 100 - 90t & 0 & 0 & 90t & 0 \\ 50 + 90t & 160 & 90 - 90t & 0 & 300 \end{pmatrix}.$$

Задавая конкретным значением параметра  $t$  от 0 до 1, можно получить частный вид оптимального решения при том же значении целевой функции  $f_{\min} = 2620$ . При  $t = 0$  и  $t = 1$  имеем два оптимальных опорных плана с тем же значением целевой функции.

### Параметрические задачи

Задачи линейного программирования, в которых коэффициенты целевой функции и (или) свободные члены системы ограничений зависят от некоторого параметра, называются *задачами параметрического программирования*.

Математическую модель такой задачи можно представить в следующем виде:

$$f(x, t) = \sum_{j=1}^n (c_j + c'_j t) x_j \rightarrow \max;$$

$$\sum_{j=1}^n a_{ij} x_j = (b_i + b'_i t), \quad i = \overline{1, m};$$

$$x_j \geq 0, \quad j = \overline{1, n}; \quad t \in [t_1; t_2].$$

Решение параметрической задачи заключается в том, чтобы при каждом значении параметра  $t \in [t_1; t_2]$  найти определенное решение задачи. Это может быть оптимальное решение, пустая или неограниченная область допустимых решений. Практически решение задачи сводится к выделению подынтервалов изменения  $t$ , на которых определенное решение устойчиво. Для решения параметрических задач используется двойственный симплекс-метод.

#### Алгоритм решения параметрической задачи

1. Определение оптимального опорного плана при некотором фиксированном значении  $t_0 \in [t_1; t_2]$ . Для этого применяется двойственный симплекс-метод.

2. Определение множества значений параметра, содержащее  $t_0$ . Для выделения этого множества (интервала) используются условия оптимальности и условия допустимости опорного плана.

3. Задание нового фиксированного значения параметра  $t$ , не принадлежащего исследованным множествам параметра  $t$ . Решение продолжается, начиная с шага 1. Этот процесс продолжается до тех пор, пока не будут исследованы решения задачи при всех значениях параметра  $t \in [t_1; t_2]$ .

**Замечание.** Множество подынтервалов  $T_l$ ,  $l = \overline{1, k}$  должно удовлетворять условию  $\bigcup_{l=1}^k T_l = [t_1; t_2]$  и конец предыдущего подынтервала  $T_l$  должен совпадать с началом последующего  $T_{l+1}$  при  $l = \overline{1, k-1}$ .

### Пример 2.11

Решить параметрическую задачу

$$f(x, t) = (-8 + t)x_1 + (2 + t)x_2 \rightarrow \max ;$$

$$\begin{cases} -3x_1 + 2x_2 \leq 6, \\ x_1 + 2x_2 \geq 4, \\ x_1 - x_2 \leq 2; \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0; \quad t \in (-\infty; +\infty).$$

### Решение

Для решения используем алгоритм двойственного симплекс-метода. При этом значение параметра  $t$  выбирается произвольно, но так, чтобы в исходной таблице существовала возможность преобразования базисного решения (т.е. не имели бы место признаки пустой или разомкнутой области допустимых решений при наличии отрицательных оценок оптимальности). В исходной табл. 2.17 принято значение параметра  $t = -10$ . Выполняем преобразования симплекс-таблицы для определения оптимального решения с учетом значения  $t = -10$ .

Таблица 2.17

Баз. пер.	$c_i$	$b_i$	$-8+t$	$2+t$	0	0	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_3$	0	6	-3	2	1	0	0
$x_4$	0	-4	-1	-2	0	1	0
$x_5$	0	2	1	1	0	0	1
$f$	-	0	$8-t$	$-2-t$	0	0	0

Исходное решение:  $x = (0, 0, 6, -4, 2)$ . Разрешающий элемент, согласно алгоритму двойственного симплекс-метода,  $a_{rk} = a_{22} = -2$ . Результат преобразований представлен в табл. 2.19.

Таблица 2.18

Баз. пер.	$c_i$	$b_i$	$-8+t$	$2+t$	0	0	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_3$	0	2	-4	0	1	1	0
$x_2$	$2+t$	2	0.5	1	0	-0.5	0
$x_5$	0	4	1.5	0	0	-0.5	1
$f$	-	$4+2t$	$9-0.5t$	0	0	$-1-0.5t$	0

При  $t = -10$  в табл. 2.18 получено оптимальное решение:

$$x^{(1)*} = (0, 2, 2, 0, 4); \quad f_{\max 1} = 4 + 2t.$$

Для определения интервала изменения параметра  $t$ , в котором оптимальное решение устойчиво, используем условие оптимальности опорного плана (т.е. неотрицательность оценок оптимальности):

$$\begin{cases} 9 - 0.5t \geq 0, \\ -1 - 0.5t \geq 0. \end{cases} \quad \text{Откуда следует, что } t \in (-\infty; -2].$$

Новое фиксированное значение параметра  $t$  выбираем за пределами выделенного интервала, но так, чтобы нарушалось только одно условие неотрицательности. Выбираем, например,  $t = 0$ . Тогда в табл. 2.18 появляется отрицательная оценка  $\Delta_4 = -1$ . Разрешающий элемент  $a_{rk} = a_{14} = 1$ . Результат преобразования представлен в табл. 2.19.

Таблица 2.19

Баз. пер.	$c_i$	$b_i$	$-8+t$	$2+t$	0	0	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_4$	0	2	-4	0	1	1	0
$x_2$	$2+t$	3	-1.5	1	0.5	0	0
$x_5$	0	5	-0.5	0	0.5	0	1
$f$	-	$6+3t$	$5-2.5t$	0	$1+0.5t$	0	0

При  $t = 0$  в табл. 2.19 получено оптимальное решение:

$$x^{(2)*} = (0, 3, 0, 2, 5); \quad f_{\max 2} = 6 + 3t.$$

Для определения интервала изменения параметра  $t$ , в котором это оптимальное решение устойчиво, используем условие оптимальности опорного плана в виде:

$$\begin{cases} 5 - 2.5t \geq 0, \\ 1 + 0.5t \geq 0. \end{cases} \quad \text{Отсюда следует, что } t \in [-2; +2].$$

Выбирая новое значение параметра  $t > 2$ , видим, что оценка  $\Delta_1 < 0$ , и при этом разрешающий столбец не содержит положительных элементов (признак неограниченности области допустимых решений). Следовательно, при  $t \in [2; +\infty)$  задача не имеет оптимального решения из-за того, что  $f_{\max} \rightarrow \infty$ .

Таким образом, получено следующее решение задачи:

$$t \in (-\infty; -2]: \quad x^{(1)*} = (0, 2, 2, 0, 4), \quad f_{\max 1} = 4 + 2t.$$

$$t \in [-2; +2]: \quad x^{(2)*} = (0, 3, 0, 2, 5), \quad f_{\max 2} = 6 + 3t.$$

$$t \in [2; +\infty): \quad f_{\max} \rightarrow \infty.$$

### Пример 2.12

Решить параметрическую задачу

$$f(x, t) = (6 - t)x_1 + (3 + t)x_2 \rightarrow \max;$$

$$\begin{cases} x_1 + x_2 \leq (8 + t), \\ x_1 - x_2 \geq (2 - t), \\ -x_1 + x_2 \leq 4; \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0; \quad t \in [-\infty; +\infty].$$

**Решение**

Для решения применяется алгоритм двойственного симплекс-метода. В исходной табл. 2.20 принято значение параметра  $t = 0$ . Учитывая это значение, находим оптимальное решение. Оптимальное решение получено в табл. 2.23.

Таблица 2.20

Баз. пер.	$c_i$	$b_i$	$6-t$	$3+t$	0	0	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_3$	0	$8+t$	1	1	1	0	0
$x_4$	0	$-2+t$	-1	1	0	1	0
$x_5$	0	4	-1	1	0	0	1
$f$	-	0	$-6+t$	$-3-t$	0	0	0

Таблица 2.21

Баз. пер.	$c_i$	$b_i$	$6-t$	$3+t$	0	0	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_3$	0	$6+2t$	0	2	1	1	0
$x_1$	$6-t$	$2-t$	1	-1	0	-1	0
$x_5$	0	$6-t$	0	0	0	-1	1
$f$	-	$12-8t+t^2$	0	-9	0	$-6+t$	0

Таблица 2.22

Баз. пер.	$c_i$	$b_i$	$6-t$	$3+t$	0	0	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_2$	$3+t$	$3+t$	0	1	0.5	0.5	0
$x_1$	$6-t$	5	1	0	0.5	-0.5	0
$x_5$	0	$6-t$	0	0	0	-1	1
$f$	-	$39+t+t^2$	0	0	4.5	$-1.5+t$	0

Пояснение к табл. 2.22: разрешающий элемент  $a_{14} = 0.5$  выбран при  $t = 0$ ; разрешающий элемент  $a_{34} = -1$  выбран при  $t = 7$ .

Таблица 2.23

Баз. пер.	$c_i$	$b_i$	$6-t$	$3+t$	0	0	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_4$	0	$6+2t$	0	2	1	1	0
$x_1$	$6-t$	$8+t$	1	1	1	0	0
$x_5$	0	$12+t$	0	2	1	0	1
$f$	-	$48-2t-t^2$	0	$3-2t$	$6-t$	0	0

В табл. 2.23 при  $t = 0$  получено оптимальное решение:

$$x^{(1)*} = (8+t; 0; 0; 6+2t; 12+t); \quad f_{\max 1} = 48 - 2t - t^2.$$

Из условий допустимости и оптимальности опорного плана определяем интервал изменения параметра  $t$ , в котором оптимальное решение устойчиво:

$$\begin{cases} 6+2t \geq 0, \\ 8+t \geq 0, \\ 12+t \geq 0, \\ 3-2t \geq 0, \\ 6-t \geq 0. \end{cases} \quad \text{Решение этой системы: } t \in [-3; 1.5].$$

Новое значение параметра задается за пределами интервала, например,  $t = 2$ . Тогда разрешающим элементом в табл. 2.23 становится элемент  $a_{12} = 2$ . В результате преобразований возвращаемся к таблице 2.22, решение в которой оптимально при новом значении параметра:  $x^{(2)*} = (5; 3+t; 0; 0; 6-t)$ ,  $f_{\max 2} = 39 + t + t^2$ . Из условий допустимости и оптимальности этого решения определяем интервал устойчивости опорного плана:  $t \in [1.5; 6]$ .

В табл. 2.23 выбираем новое значение параметра  $t < -3$ , например,  $t = -4$ . Тогда  $x_4 < 0$ , а в строке этой переменной нет отрицательных коэффициентов, поэтому, согласно признаку пустой области допустимых решений двойственного симплекс-метода, при всех  $t < -3$  задача не имеет решений.

Возвращаясь к табл. 2.22, выбираем новое значение параметра  $t > 6$  (например,  $t = 7$ ). Тогда разрешающий элемент выбирается в табл. 2.22 на пересечении третьей строки (т.к.  $x_5 < 0$ ) и четвертого столбца (см. алгоритм двойственного симплекс-метода). В результате преобразований получаем таблицу 2.24 с оптимальным решением:

$x^{(3)*} = (2 + 0.5t; 6 + 0.5t; 0; -6 + t; 0)$ ;  $f_{\max 3} = 30 + 8.5t$  в интервале  $t \in [6; \infty)$ .

Таблица 2.24

Баз. пер.	$c_i$	$b_i$	$6-t$	$3+t$	0	0	0
			$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_2$	$3+t$	$6+0.5t$	0	1	0.5	0	0.5
$x_1$	$6-t$	$2+0.5t$	1	0	0.5	0	-0.5
$x_4$	0	$-6-t$	0	0	0	1	-1
$f$	-	$30+8.5t$	0	0	4.5	0	$-1.5+t$

Таким образом, получено следующее решение задачи по интервалам:

$$t \in (-\infty; -3): \quad \text{ОДР} = \emptyset .$$

$$t \in [-3; 1.5]: \quad x^{(1)*} = (8+t; 0; 0; 6+2t; 12+t); \quad f_{\max 1} = 48 - 2t - t^2 .$$

$$t \in [1.5; 6]: \quad x^{(2)*} = (5; 3+t; 0; 0; 6-t); \quad f_{\max 2} = 39 + t + t^2 .$$

$$t \in [6; \infty): \quad x^{(3)*} = (2 + 0.5t; 6 + 0.5t; 0; -6 + t; 0);$$

$$f_{\max 3} = 30 + 8.5t .$$

Значения параметра  $t$  на концах подынтервалов называются *критическими точками*. Определим решения в критических точках и сравним решения в этих точках смежных подынтервалов:

$$t = 1.5: \quad x^{(1)*} = (9.5; 0; 0; 9; 13.5), \quad f_{\max 1} = 42.75;$$

$$x^{(2)*} = (5; 4.5; 0; 0; 4.5), \quad f_{\max 2} = 42.75;$$

– решение в этой точке альтернативное.

$$t = 6: \quad x^{(2)*} = (5; 9; 0; 0; \underline{0}), \quad f_{\max 2} = 81;$$

$$x^{(3)*} = (5; 9; 0; \underline{0}; 0), \quad f_{\max 3} = 81;$$

– решение в этой точке вырожденное. Базисные переменные, равные нулю, подчеркнуты.

### Задача дробно-линейного программирования

Это задача, в которой целевая функция дробно-линейна, а допустимое множество  $U$  определяется линейными ограничениями и условиями неотрицательности переменных:

$$f(x) = \frac{\langle c, x \rangle + c_0}{\langle d, x \rangle + d_0} = \frac{\sum_{j=1}^n c_j x_j + c_0}{\sum_{j=1}^n d_j x_j + d_0} \rightarrow \max ; \quad (2.58)$$

$$\begin{cases} \sum_{j=1}^n a_j^{(i)} x_j = \langle a^{(i)}, x \rangle = b_i, & i = \overline{1, k}, \\ \sum_{j=1}^n a_j^{(i)} x_j = \langle a^{(i)}, x \rangle \leq b_i, & i = \overline{k+1, m}; \end{cases} \quad (2.59)$$

$$x_j \geq 0, \quad j = \overline{1, n}, \quad (2.60)$$

где  $c$  и  $d$  – фиксированные векторы в  $R^n$ . Предполагается, что знаменатель целевой функции в  $U$ :  $\sum_{j=1}^n d_j x_j + d_0 > 0$ . Такое условие не нарушает общности задачи, поскольку в том случае, когда эта величина отрицательна, знак минус можно отнести к числителю.

Задача (2.58) – (2.60) сводится к задаче линейного программирования. При этом существует два метода преобразования исходной модели:

- 1) с помощью введения новых переменных;
- 2) с помощью преобразования целевой функции к параметрическому виду (назовем этот подход к решению задачи *прямым методом*).

**1.** Будем считать, что на допустимом множестве  $U$  знаменатель целевой функции (2.58) сохраняет знак, т.е.  $\langle d, x \rangle + d_0 > 0$  для всех  $x \in U$ . Обозначим  $y_0 = 1/(\langle d, x \rangle + d_0)$ . Очевидно, что  $y_0 > 0$  при всех  $x \in U$ . Введем новые переменные:  $y_j = \overline{y_0} x_j, \quad j = \overline{1, n}$ .

В этих переменных задача (2.58) – (2.60) принимает следующий вид:

$$\overline{f}(x) = \sum_{j=0}^n c_j y_j \rightarrow \max ;$$

$$\begin{cases} \sum_{j=1}^n a_{ij} y_j - b_i y_0 = 0, & i = \overline{1, k}, \\ \sum_{j=1}^n a_{ij} x_j - b_i y_0 \leq 0, & i = \overline{k+1, m}, \\ \sum_{j=0}^n d_j y_j = 1; \\ y_j \geq 0, & j = 0, 1, \dots, n, \end{cases}$$

т.е. превращается в задачу линейного программирования в пространстве  $R^n$ . Симплекс–методом определяется оптимальное решение  $y^* = (y_0^*, \dots, y_n^*)$ ,  $\bar{f}_{\max}(y^*)$  и с помощью равенств  $x_j^* = y_j^* / y_0^*$ ,  $j = \overline{1, n}$ ,  $f_{\max}(x^*) = \bar{f}_{\max}(y^*)$  получим решение исходной задачи дробно–линейного программирования.

Если  $y_0^* = 0$ , то допустимое множество  $U$  задачи (2.58) – (2.60) не ограничено и максимум целевой функции  $f(x)$  на нем не достигается.

**Пример 2.13.**

Решить следующую задачу дробно–линейного программирования:

$$f(x) = \frac{2x_1 - x_2}{x_1 + 2x_2 + 1} \rightarrow \max ;$$

$$\begin{cases} x_1 - 2x_2 + x_3 = 2, \\ 2x_1 + x_2 + x_4 = 6; \end{cases} \quad x_j \geq 0, \quad j = \overline{1, 4}.$$

**Решение.**

Вводим дополнительную переменную  $y_0 = 1/(x_1 + 2x_2 + 1)$  и переходя к новым переменным  $y_j = y_0 x_j$ ,  $j = \overline{1, 4}$ , получим задачу линейного программирования:

$$f(x) = 2y_1 - y_2 \rightarrow \max ;$$

$$\begin{cases} y_1 - 2y_2 + y_3 - 2y_0 = 0, \\ 2y_1 + y_2 + y_4 - 6y_0 = 0, \\ y_1 + 2y_2 + y_0 = 1; \end{cases} \quad y_j \geq 0, \quad j = 0, 1, \dots, n.$$

Решая полученную задачу линейного программирования одним из вариантов симплекс метода, находим оптимальное решение

$$y_0^* = 1/3, \quad y_1^* = 2/3, \quad y_2^* = 0, \quad y_3^* = 0, \quad y_4^* = 2/3; \quad f_{\max} = 4/3.$$

Переходя к первоначальным переменным, получим оптимальное решение исходной задачи дробно–линейного программирования:

$$x^* = (2, 0, 0, 2), \quad f_{\max} = 4/3.$$

2. При решении задачи дробно–линейного программирования прямым методом преобразуется только целевая функция в линейно–параметрическую форму, т.е. задача (2.58) – (2.60) приводится к следующему виду:

$$f(x) = \frac{1}{d_0} \sum_{j=1}^n (c_j - f(x)) x_j + \frac{c_0}{d_0} \rightarrow \max, \quad d_0 \neq 0;$$

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

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

Полученная параметрическая задача решается двойственным симплекс–методом. При этом оценки оптимальности определяются с учетом текущего значения целевой функции.

### Пример 2.14

Решить следующую задачу дробно–линейного программирования *прямым методом*:

$$f(x) = \frac{3x_1 - x_2}{x_1 + x_2 + 2} \rightarrow \max;$$

$$\begin{cases} -x_1 - x_2 + x_3 = -3, \\ 2x_1 + x_2 + x_4 = 10; \end{cases}$$

$$x_j \geq 0, \quad j = \overline{1, 4}.$$

### Решение

Приведем исходную задачу к следующему виду:

$$f = (-f/2 + 3/2)x_1 + (-f/2 - 1/2)x_2 \rightarrow \max;$$

$$\begin{cases} -x_1 - x_2 + x_3 = -3, \\ 2x_1 + x_2 + x_4 = 10; \end{cases}$$

$$x_j \geq 0, \quad j = \overline{1, 4}.$$

Решим задачу двойственным симплекс–методом в модифицированной форме. В исходной таблице 2.25 значение целевой функции  $f = 0$ . Следовательно, оценки оптимальности  $\Delta_1 = -3/2$  и  $\Delta_2 = 1/2$ . Кроме того, базисная переменная  $x_3 = -3$  не удовлетворяет условию неотрицательности. Таким образом, согласно алгоритму двойственного симплекс-метода, выбираем разрешающий элемент  $a_{rk} = a_{11} = -1$ . Относительно этого элемента выполняем преобразования таблицы 2.25.

Таблица 2.25

Б.п. \ С.п.	$x_1$	$x_2$	$b_i$
$x_3$	-1	-1	-3
$x_4$	2	1	10
$f$	-3/2	1/2	0
–	$f/2$	$f/2$	0

В результате преобразования получаем таблицу 2.26 с текущим опорным планом  $x = (3, 0, 0, 4)$  и целевой функцией  $f = 9/2 - 3f/2$ . Из последнего уравнения  $f = 9/5$ . С учетом этого значения целевой функции определяем оценки  $\Delta_3 = -3/2 + f/2 = -3/5$ ,  $\Delta_2 = 2$ .

Таблица 2.26

Б.п. \ С.п.	$x_3$	$x_2$	$b_i$
$x_1$	-1	1	3
$x_4$	2	-1	4
$f$	-3/2	2	9/2
–	$f/2$	0	$-(3f)/2$

Таблицу 2.26, согласно двойственному симплекс-методу, преобразуем относительно разрешающего элемента  $a_{rk} = a_{21} = 2$ .

Получаем таблицу 2.27. Из уравнения  $f = 15/2 - 5f/2$  относительно целевой функции находим  $f = 15/7$ . Следовательно, оценки оптимальности  $\Delta_4 = 3/4 - f/2 = 3/14$  и  $\Delta_2 = 5/4 + f/4 = 25/14$ .

Таблица 2.27.

Б.п. \ С.п.	$x_4$	$x_2$	$b_i$
$x_1$	1/2	1/2	5
$x_3$	1/2	-1/2	2
$f$	3/4	5/4	15/2
-	$-f/4$	$f/4$	$-(5f)/2$

Отрицательных оценок нет. В таблице 2.27 получено следующее оптимальное решение исходной задачи дробно-линейного программирования:  $x^* = (5, 0, 2, 0)$ ,  $f_{\max} = 15/7$ .

## Глава 3

### МЕТОДЫ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

Многие содержательные задачи оптимизации приводят к математическим моделям, в которых все переменные или некоторые из них принимают целочисленные значения. Такие задачи оптимизации называют *дискретными*. В общем случае целочисленными могут быть не все, а часть переменных. В частности, множества решений могут быть не только конечными, но и состоящими всего из двух значений: 0 и 1. В этом случае имеет место дискретная оптимизация с бинарными переменными, такие задачи называются *булевыми*.

Как правило, методы, разработанные для решения непрерывных задач оптимизации, не могут быть использованы применительно к дискретным задачам. В этой главе рассмотрены некоторые из методов, рассчитанных на решение дискретных задач оптимизации: метод отсечений, метод ветвей и границ, аддитивный метод Балаша.

#### 3.1. Примеры дискретных оптимизационных задач

Широкий класс дискретных задач оптимизации составляют *целочисленные задачи математического программирования*. В этих задачах наряду с линейными ограничениями на множество допустимых решений содержатся требования целочисленности, налагаемые на какие-либо переменные  $x_j$ . Такие задачи возникают, когда управляемые переменные  $x_j$  по своему смыслу могут принимать только целочисленные значения (например,  $x_j$  – это количество единиц используемого оборудования или число резервных блоков в схеме электронного устройства и т.п.).

Требование целочисленности будем записывать в виде  $x_j \in Z$ ,  $j \in J$ , где  $J \subset \{1, 2, \dots, n\}$  – множество индексов, соответствующих целочисленным переменным,  $Z$  – множество целых чисел.

**Пример 3.1.** Полностью целочисленная задача линейного программирования.

В цехе площадью  $b_1$  предстоит установить станки, на приобретение которых выделено денежных средств  $b_2$ . Существует  $n$  типов станков подходящего назначения с различными характеристиками: один станок  $j$ -го типа ( $j = \overline{1, n}$ ) требует  $a_{1j}$  производственных площадей, стоит  $a_{2j}$  и имеет производительность  $c_j$  изделий в смену.

Построить математическую модель задачи оптимизации с точки зрения максимальной производительности цеха.

**Решение.** В качестве управляемых переменных выберем количества закупаемых станков  $j$ -го типа  $x_j, j = \overline{1, n}$ . Очевидно, все  $x_j$  могут принимать только целочисленные значения. С учетом ограниченности производственных площадей и средств на приобретение станков в качестве целевой функции возьмем суммарную производительность закупленных станков. Получим следующую математическую модель:

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \max;$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2;$$

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

которая представляет собой полностью целочисленную задачу линейного программирования.

Во многих практических случаях возникает задача оптимизации принимаемого решения, каждая из составных частей которого представляет собой выбор одного из двух возможных вариантов. Приписав  $j$ -му выбору варианта управляемую переменную  $x_j$ , принимающую значения 0 или 1 в зависимости от того, какой из двух вариантов выбран, получим частный случай целочисленной задачи оптимизации – задачу с *булевыми переменными*.

**Пример 3.2.** Задача математического программирования с булевыми переменными.

Электронная система состоит из  $n$  функционально законченных блоков, каждый из которых допускает два варианта реализации. Эти варианты для  $j$ -го блока различаются вероятностями отказа и массами (соответственно и  $q_j^{(1)}$  и  $m_j^{(1)}$  в первом варианте и  $q_j^{(2)}$  и  $m_j^{(2)}$  во втором,  $j = \overline{1, n}$ ), причем  $q_j^{(1)}, q_j^{(2)} \ll 1$ . Отказ любого из блоков ведет

к отказу всей системы. Масса системы не должна превышать величину  $M$ .

Составить математическую модель задачи оптимизации с точки зрения ее надежности (вероятности отказа).

**Решение.** Введем управляемые переменные  $x_j, j = \overline{1, n}$  следующим образом. Будем считать, что  $x_j = 0$ , если выбран первый вариант реализации  $j$ -го блока, и  $x_j = 1$ , если выбран второй вариант.

Положим:

$$q_j(x_j) = \begin{cases} q_j^{(1)} & \text{при } x_j = 0, \\ q_j^{(2)} & \text{при } x_j = 1, \end{cases} \quad m_j(x_j) = \begin{cases} m_j^{(1)} & \text{при } x_j = 0, \\ m_j^{(2)} & \text{при } x_j = 1. \end{cases}$$

Тогда целевая функция (вероятность отказа системы) записывается в виде:

$$f(x) = 1 - \prod_{j=1}^n (1 - q_j(x_j)),$$

откуда с учетом условия  $q_j(x_j) \ll 1$  получаем:

$$f(x) = \sum_{j=1}^n q_j(x_j).$$

Учитывая, что общая масса системы равна  $\sum_{j=1}^n m_j(x_j)$ , записываем математическую модель рассматриваемой задачи:

$$f(x) = \sum_{j=1}^n q_j(x_j) \rightarrow \min, \tag{3.1}$$

$$\sum_{j=1}^n m_j(x_j) \leq M, \quad x_j \in \{0, 1\}, \quad j = \overline{1, n}.$$

Это задача нелинейного программирования с булевыми переменными.

Отметим, что задачу (3.1) можно записать и в виде задачи линейного программирования с булевыми переменными. Для этого обозначения  $a_j = m_j(1) - m_j(0)$ ,  $c_j = q_j(1) - q_j(0)$ . Тогда  $m_j(x_j) = m_j(0) + a_j x_j$ ,  $q_j(x_j) = q_j(0) + c_j x_j$ . Подставив эти зависимости в (3.1), получим линейную математическую модель

$$f(x) = \sum_{j=1}^n c_j x_j + \sum_{j=1}^n q_j(0) \rightarrow \min,$$

$$\sum_{j=1}^n a_j x_j \leq M - \sum_{j=1}^n m_j(0), \quad x_j \in \{0,1\}, \quad j = \overline{1,n}.$$

Как уже отмечалось, методы, разработанные для решения непрерывных задач оптимизации, в большинстве случаев неприменимы к дискретным задачам. Так, если целочисленную задачу линейного программирования решить симплекс-методом, а затем округлить найденные значения переменных до ближайших допустимых целых чисел, то полученное решение может значительно отличаться от оптимального решения исходной задачи.

В качестве примера рассмотрим задачу линейного программирования с булевыми переменными

$$f(x) = -10x_1 - 8x_2 - 7x_3 \rightarrow \min, \\ 9x_1 + 8x_2 + 8x_3 \leq 16, \quad (3.2)$$

$$x_j \in \{0,1\}, \quad j = 1,2,3. \quad (3.3)$$

Заменяя ограничение (3.3) неравенствами  $0 \leq x_j \leq 1, j = 1,2,3$ , и решив полученную задачу симплекс-методом, найдем:  $x_1 = 1, x_2 = 7/8, x_3 = 0, f(x) = -17$ . Округление  $x_2$  до 1 недопустимо, так как при этом нарушится ограничение (3.2). Если округлить  $x_2$  до 0, то получим  $x = (1,0,0), f(x) = -10$ . Это решение заметно отличается от оптимального решения исходной задачи  $x^* = (0,1,1), f_{\min} = -15$  (его можно найти, например, полным перебором).

### 3.2. Метод отсекающих плоскостей (метод Гомори)

Метод отсекающих плоскостей рассчитан на решение целочисленных задач линейного программирования и основан на последовательном исключении из рассмотрения (*отсечении*) некоторых частей допустимого множества нецелочисленной задачи.

Рассмотрим полностью целочисленную задачу линейного программирования:

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \max, \quad (3.4)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1,m}, \quad x_j \geq 0, \quad j = \overline{1,n} \quad (3.5)$$

$$x_j \in Z, \quad j = \overline{1,n}. \quad (3.6)$$

В двумерном случае допустимое множество  $U$  этой задачи легко представить геометрически как совокупность точек целочисленной координатной сетки, которые принадлежат допустимому множеству  $\overline{U}$  задачи без требования целочисленности (т.е. выпуклому многоугольнику).

Если требованию целочисленности подчинены все  $n$  переменных задачи, то она называется *полностью целочисленной*, а если только часть из них, то говорят о *частично целочисленной* задаче.

Основная идея метода отсекающих плоскостей состоит в том, что к ограничениям задачи линейного программирования (3.4) – (3.5) добавляются линейные ограничения (*отсечения*), не исключающие целочисленных допустимых точек, т.е. не изменяющие решения целочисленной задачи. Отсечения вводятся в условие задачи (3.4) – (3.5) до тех пор, пока ее решение не станет целочисленным, т.е. совпадет с искомым решением задачи (3.4) – (3.6).

Рассмотрим алгебраический метод построения отсекающих плоскостей (*метод Гомори*).

Пусть  $\overline{x}^* \in \overline{U}$  – решение задачи (3.4) – (3.5). Дополнительное ограничение

$$\sum_{j=1}^n \{\alpha_{kj}\} x_j \geq \{\beta_k\}, \quad (3.7)$$

где  $\{\alpha_{kj}\}$  и  $\{\beta_k\}$  – дробные части соответствующих коэффициентов, будем называть *правильным отсечением* допустимого множества этой задачи если:

- 1) точка  $\overline{x}^*$  не удовлетворяет этому неравенству (*условие отсечения*);
- 2) любая точка допустимого множества целочисленной задачи (3.4) – (3.6) ему удовлетворяет (*условие правильности*).

В основе метода Гомори лежит двойственный симплекс-метод. Правильное отсечение строится следующим образом:

1. Двойственным симплекс-методом находится оптимальное решение  $\overline{x}^* = (\beta_1, \dots, \beta_m, 0, \dots, 0)$  *нецелочисленной* задачи (3.4) – (3.5). Пусть переменная  $x_k$  в этом решении не является целой, т.е. дробная часть числа  $\beta_k$   $\{\beta_k\} > 0$ . Напомним, что всякое действительное число, например  $\beta_k$ , представимо в виде  $\beta_k = [\beta_k] + \{\beta_k\}$ , где  $[\beta_k]$  – его целая часть. Целой частью любого дробного числа является наибольшее целое число, не превосходящее  $\beta_k$ , тогда  $\{\beta_k\} = \beta_k - [\beta_k]$  – дробная часть числа  $\beta_k$ . Следовательно, дробная часть любого дроб-

ного числа всегда положительна, т.е.  $\{\beta_k\} > 0$ . Например,  $\beta_k = -3.2$ . Тогда  $[\beta_k] = -4$  и  $\{\beta_k\} = \beta_k - [\beta_k] = -3.2 - (-4) = 0.8$ .

2. В симплекс-таблице, определяющей решение  $\bar{x}^*$ ,  $k$ -ой строке соответствует уравнение, где  $\beta_k$  – нецелое число:

$$x_k + \sum_{j=m+1}^n \alpha_{kj} x_j = \beta_k. \quad (3.8)$$

3) Используя (3.8) и согласно (3.7), строим дополнительное ограничение:

$$- \sum_{j=m+1}^n \{\alpha_{kj}\} x_j \leq -\{\beta_k\}, \quad (3.9)$$

которое с помощью новой переменной  $x_{n+1} \geq 0$  можно представить в виде равенства:

$$x_{n+1} - \sum_{j=m+1}^n \{\alpha_{kj}\} x_j = -\{\beta_k\}. \quad (3.10)$$

Покажем, что ограничение (3.9) (или (3.10)) является правильным отсечением.

**Теорема 3.1.** (О правильном отсечении). Пусть задача линейного программирования (3.4) – (3.5) имеет решение  $\bar{x}^*$ , причем его  $k$ -я координата не является целой, а допустимое множество целочисленной задачи (3.4)-(3.6) не пусто. Тогда ограничение (3.9) (или (3.10)) есть правильное отсечение допустимого множества  $\bar{U}$  задачи (3.4) – (3.5).

**Доказательство**

1. Покажем, что неравенство (3.9) определяет отсечение, т.е. решение  $\bar{x}^* = (\beta_1, \dots, \beta_m, 0, \dots, 0)$  нецелочисленной задачи (3.4) – (3.5) ему не удовлетворяет. В самом деле, так как координаты  $x_j$ ,  $j = \overline{m+1, n}$  точки  $\bar{x}^*$  равны нулю, то и левая часть неравенства (3.9) при  $x = \bar{x}^*$  обращается в нуль. С другой стороны,  $\{\beta_k\} > 0$ , поэтому условие (3.9) для  $x = \bar{x}^*$  не выполняется.

2. Убедимся в том, что отсечение (3.10) (или (3.9)) – правильное, т.е. любая целочисленная точка  $x' = (x'_1, \dots, x'_n)$  допустимого множества  $\bar{U}$  задачи (3.4) – (3.5) удовлетворяет ограничению (3.10) при некотором  $x_{n+1} \geq 0$ . Для этого покажем сначала, что переменная  $x_{n+1}$

в равенстве (3.10) при  $x_j = x'_j$  принимает целочисленное значение  $x'_{n+1}$ . В самом деле, так как точка  $x' \in \bar{U}$ , то она удовлетворяет уравнению (3.8), поэтому с учетом (3.10) можно записать:

$$\begin{aligned} x'_k &= \beta_k - \sum_{j=1}^n \alpha_{kj} x'_j = [\beta_k] + \{\beta_k\} - \sum_{j=1}^n [\alpha_{kj}] x'_j - \sum_{j=1}^n \{\alpha_{kj}\} x'_j = \\ &= [\beta_k] - \sum_{j=1}^n [\alpha_{kj}] x'_j - x'_{n+1}, \end{aligned}$$

откуда

$$x'_{n+1} = [\beta_k] - \sum_{j=1}^n [\alpha_{kj}] x'_j - x'_k.$$

Так как все числа в правой части последнего равенства – целые, то

$$x'_{n+1} \in Z. \quad (3.11)$$

Теперь докажем, что  $x'_{n+1} \geq 0$ . Предположим обратное, т.е. что  $x'_{n+1} < 0$ . Тогда из (3.10) получим:

$$-\{\beta_k\} + \sum_{j=m+1}^n \{\alpha_{kj}\} x'_j < 0. \quad (3.12)$$

Так как  $-\{\beta_k\} > -1$ , а  $\sum_{j=m+1}^n \{\alpha_{kj}\} x'_j \geq 0$ , то неравенство (3.12)

можно дополнить:

$$-1 < -\{\beta_k\} + \sum_{j=m+1}^n \{\alpha_{kj}\} x'_j < 0,$$

откуда с учетом (3.10) следует, что  $-1 < x'_{n+1} < 0$ , а это противоречит доказанному утверждению (3.11).

Таким образом, предположение  $x_{n+1} < 0$  приводит к противоречию, поэтому  $x_{n+1} \geq 0$ , и произвольная точка  $x' \in \bar{U}$  с целочисленными координатами удовлетворяет ограничению (3.10). Это и доказывает, что отсечение (3.9) или (3.10) является правильным.

Таким образом, при реализации метода отсечений в условии нецелочисленной задачи линейного программирования последовательно добавляются новые ограничения вида (3.10) до тех пор, пока ее решение не станет целочисленным или не будет обнаружено, что целочисленное решение отсутствует.

З а м е ч а н и я :

1. При использовании симплекс-метода добавлению ограничения (3.10) в условие задачи (3.4)–(3.5) соответствует ввод дополнительной строки в симплекс-таблицу с решением  $\bar{X}^*$  этой задачи

$$x_{n+1} \mid \alpha_{n+1,m+1}, \dots, \alpha_{n+1,n} \mid \beta_{n+1}, \quad (3.13)$$

где  $\alpha_{n+1,j} = -\{\alpha_{kj}\}$ ,  $j = m+1, n$ ;  $\beta_{n+1} = -\{\beta_k\}$ .

2. После добавления к симплекс-таблице дополнительной строки (3.13) теряется опорный план поскольку  $b_{n+1} = -\{\beta_k\} < 0$ . Поэтому для решения вновь полученной задачи следует совершить необходимое число шагов двойственного симплекс-метода для получения опорного плана или убедиться в ее неразрешимости.

При добавлении очередного ограничения вида (3.10) описанная ситуация и последовательность действий повторяются.

*Алгоритм метода Гомори* включает следующие основные шаги:

1. Определение оптимального опорного плана исходной задачи без учета требования целочисленности переменных.

2. Составление дополнительного ограничения (3.10) для базисной переменной, которая в полученном опорном плане задачи имеет максимальное дробное значение, а в оптимальном опорном плане исходной задачи должна быть целочисленной.

3. Определение оптимального опорного плана задачи с учетом присоединенного дополнительного ограничения. При этом используется двойственный симплекс-метод.

4. Анализ полученного результата. Процесс решения заканчивается, если получено оптимальное решение исходной задачи, удовлетворяющее требованиям целочисленности, или устанавливается ее неразрешимость. В противном случае процесс решения продолжается, начиная с шага 2.

З а м е ч а н и е .

Чтобы избежать накопления неопределенного числа строк в симплекс-таблицах, из них обычно исключают дополнительную переменную, соответствующую какому-либо отсечению, введенному ранее, если в некоторый момент выполнения алгоритма эта переменная переходит из свободных в базисные. Это означает, что число строк симплекс-таблиц никогда не превосходит числа исходных переменных  $n$ , а число используемых на любом шаге алгоритма дополнительных ограничений не превосходит  $n - m$ .

**Пример 3.3.**

Методом Гомори найти оптимальное решение следующей задачи:

$$f(x) = -4x_1 - 5x_2 \rightarrow \max; \quad (3.14)$$

$$\begin{cases} 3x_1 + 5x_2 \geq 6, \\ x_1 + x_2 \leq 5; \end{cases} \quad (3.15)$$

$$x_j \geq 0, x_j \in Z, j = 1, 2. \quad (3.16)$$

Дать геометрическую интерпретацию.

**Решение.**

Для определения оптимального решения задачи (3.14) – (3.16) в соответствии с алгоритмом Гомори сначала находим оптимальное решение задачи без учета целочисленности. А затем последовательно вводим дополнительные ограничения. Для выполнения преобразований используем двойственный симплекс–метод в форме *модифицированных преобразований* метода Жордана-Гаусса.

Строим каноническую форму математической модели:

$$f(\mathbf{x}) = -4x_1 - 5x_2 \rightarrow \max; \quad (3.17)$$

$$\begin{cases} -3x_1 - 5x_2 + x_3 = -6, \\ x_1 + x_2 + x_4 = 5; \end{cases} \quad (3.18)$$

$$x_j \geq 0, x_j \in Z, j = \overline{1, 4}. \quad (3.19)$$

Результаты решения задачи (3.17) – (3.19) представлены в табл. 3.1 и табл. 3.2.

Таблица 3.1

Св.п. Баз.п.	$x_1$	$x_2$	$b_i$
$x_3$	-3	-5	-6
$x_4$	1	1	5
$f$	4	5	0

Таблица 3.2

Св.п. Баз.п.	$x_1$	$x_3$	$b_i$
$x_2$	0.6	-0.2	1.2
$x_4$	0.4	0.2	3.8
$f$	1	1	-6

Как следует из табл.3.2, найденный оптимальный опорный план  $\bar{\mathbf{x}}^* = (0; 1.2; 0; 3.8)$ ,  $\bar{f}_{\max} = -6$  не является решением исходной задачи, т.к. решение не удовлетворяет условию целочисленности. В соответствии с (3.7) строим дополнительное ограничение для базисной переменной  $x_4$ , имеющей наибольшую дробную часть, и преобразуем его в уравнение, используя дополнительную неотрицательную переменную  $x_5$ :

$$-0.4x_1 - 0.2x_3 + x_5 = -0.8. \quad (3.20)$$

Поиск оптимального решения исходной задачи продолжаем с учетом ограничения (3.20). Выполненные преобразования представлены в табл. 3.3 и табл. 3.4.

Таблица 3.3

Св.п. Баз.п.	$x_1$	$x_3$	$b_i$
$x_2$	0.6	-0.2	1.2
$x_4$	0.4	0.2	3.8
$x_5$	-0.4	-0.2	-0.8
$f$	1	1	6

Таблица 3.4

Св.п. Баз.п.	$x_5$	$x_3$	$b_i$
$x_2$	1.5	-1	0
$x_4$	1	0	3
$x_1$	-2.5	0.5	2
$f$	2.5	0.5	-8

Решение, полученное в табл. 3.4, является оптимальным и целочисленным. Таким образом, исходная задача (3.14) – (3.16) имеет следующее оптимальное решение:  $x^* = (2, 0, 2, 3, 0)$ ,  $f_{\max} = -8$ .

Рассмотрим геометрическую интерпретацию решения исходной задачи методом Гомори. Приведем ограничение (3.20) к переменным  $x_1$  и  $x_2$ , отбрасывая при этом  $x_3 \geq 0$  и исключая  $x_3 \geq 0$ . Полученное неравенство добавляем к системе ограничений исходной задачи:

$$f(x) = -4x_1 - 5x_2 \rightarrow \max ; \quad (3.21)$$

$$\begin{cases} 3x_1 + 5x_2 \geq 6, & (1) \\ x_1 + x_2 \leq 5, & (2) \\ x_1 + x_2 \geq 2; & (3) \end{cases} \quad (3.22)$$

$$x_j \geq 0, j = 1, 2. \quad (3.23)$$

Таким образом, задаче (3.14) – (3.16) эквивалентна задача (3.21) – (3.23). Третье неравенство в системе ограничений (3.22) является отсекающей полуплоскостью Гомори (в многомерном пространстве это было бы отсекающее полупространство). Решение этой задачи графическим методом представлено на рис. 3.1.

Пунктирной линией на рис. 3.1 показана граница дополнительного ограничения (отсекающей плоскости).

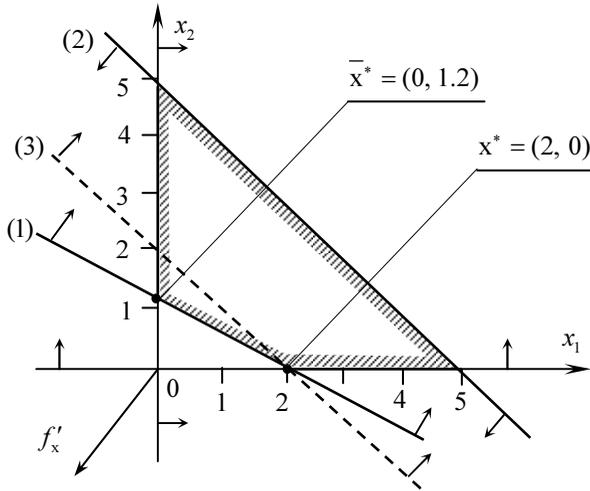


Рис. 3.1.

Несмотря на то, что метод Гомори является надежным средством решения задач целочисленного программирования, его применение нецелесообразно, если исходная задача имеет большую размерность. Более того, встречаются примеры, когда изменение порядка выбора дополнительных ограничений при решении задачи целочисленного программирования средней размерности приводит к неоправданному возрастанию объема вычислений при поиске оптимального решения.

### 3.3. Метод ветвей и границ

Метод ветвей и границ применяется для решения разнообразных задач дискретной оптимизации. Его идея состоит в последовательном разбиении допустимого множества  $U_0$  исходной задачи

$$f(x) \rightarrow \min, \quad x \in U_0, \quad U_0 - \text{дискретно} \quad (3.24)$$

на взаимно непересекающиеся подмножества  $U_k$  (этот процесс называется *ветвлением*) и получении оценок (*границ*)  $f_k$  значений целевой функции  $f(x)$  на этих подмножествах ( $\min_{U_k} f(x) \geq f_k$ ). При

выполнении определенных условий процесс ветвления завершается и решение  $x^* = x^{(k)}$  задачи на одном из подмножеств  $U_k$  оказывается решением исходной задачи (3.24). Сказанное выше и объясняет название метода.

Схему поиска решения методом ветвей и границ в каждом конкретном случае можно наглядно представить в виде некоторого *дерева*, состоящего из множества *вершин* и соединяющих их *ветвей* (рис. 3.2).

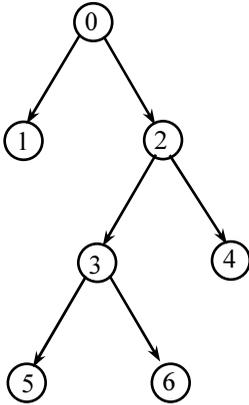


Рис. 3.2.

Начальной вершине 0 соответствует исходное допустимое множество  $U_0$  или исходная задача (3.24), а любой другой вершине  $k$  – подмножество  $U_k$ , полученное в результате ветвления, или подзадача  $k$  :

$$f(x) \rightarrow \min, x \in U_k. \quad (3.25)$$

Если при ветвлении из каждой вершины происходит разбиение соответствующего ей подмножества на две части, то схема метода изображается *бинарным* деревом (рис. 3.2).

Процесс ветвления из данной вершины  $k$  не производится, если выполнено одно из следующих условий:

1. Граница  $f_k$  найдена точно:  $f_k = \min_{U_k} f(x)$ , т.е. получено целочисленное решение  $(x^{(k)}, f_k)$  подзадачи (3.25). Говорят, что в этом случае вместо оценки решения в вершине  $k$  найдено *полное решение*, соответствующее этой вершине.
2. Множество  $U_k$  является пустым.
3. Из полученных до этого *полных* решений найдется такое  $(x^{(l)}, f_l)$ , что граница  $f_k$  удовлетворяет неравенству

$$f_k \geq f_l. \quad (3.26)$$

В этом случае дальнейший поиск решения исходной задачи на подмножестве  $U_k$  не имеет смысла. В самом деле, из (3.26) следует, что  $f_k^* = \min_{U_k} f(x) \geq f_k \geq f_l$  т.е. минимальное значение  $f_k^*$  функции  $f(x)$  на  $U_k$  не может быть меньше, чем в уже найденной точке  $x^{(l)*} \in U_l \subset U_0$ .

Вершины, удовлетворяющие одному из условий 1–3, называются *прозондированными*. Непрозондированные вершины, из которых ветвление еще не произведено, называются *активными*.

Процесс ветвления продолжается до тех пор, пока остается хотя бы одна активная вершина. По окончании процедуры ветвления можно

указать решение исходной задачи (3.24) – это то из найденных полных решений  $(x^{(k)*}, f_k^*)$ , для которого значение  $f_k^*$  минимально.

Метод ветвей и границ обладает большой универсальностью и может применяться для решения не только целочисленных задач. Однако он оказывается достаточно эффективным лишь в том случае, когда используемый конкретный алгоритм учитывает специфику решаемой задачи.

Рассмотрим применение метода ветвей и границ к решению целочисленной задачи линейного программирования

$$f(x) = \langle c, x \rangle \rightarrow \min, \quad Ax = b, \quad x \geq 0; \quad (3.27)$$

$$x_j \in Z, \quad j \in I, \quad (3.28)$$

где  $I$  – заданное множество индексов, соответствующих целочисленным переменным,  $I \subset \{1, 2, \dots, n\}$ .

Алгоритм метода ветвей и границ предполагает задание способа получения нижних границ  $f_k$  значений  $f(x)$  на подмножествах  $U_k$  и правила ветвления, т.е. разбиения  $U_k$  на непересекающиеся подмножества. Для задачи (3.27), (3.28) упомянутые операции можно осуществлять следующим образом.

Пусть активной вершине  $k$  соответствует подмножество  $U_k$  допустимых решений, которые, разумеется, удовлетворяют условию (3.28). Рассмотрим наряду с  $U_k$  множество  $\bar{U}_k$ , определенное теми же ограничениями, что и  $U_k$ , за исключением требования целочисленности (3.28). Найдем решение  $(x^{(k)}, f_k)$  нецелочисленной задачи линейного программирования

$$f(x) \rightarrow \min, \quad x \in \bar{U}_k. \quad (3.29)$$

Очевидно,  $U_k \subset \bar{U}_k$ , поэтому  $\min_{U_k} f(x) = f_k^* \geq \min_{\bar{U}_k} f(x) = f_k$ ,

т.е.  $f_k$  является нижней границей для вершины  $k$ . Таким образом, способ получения нижних границ описан.

Ветвление будем производить следующим образом. Пусть  $x^{(k)}$  не является полным решением, т.е. условие (3.28) для  $x^{(k)}$  не выполняется. Выберем нецелочисленную компоненту  $x_m^{(k)}$  с номером  $m \in I$  и разобьем  $U_k$  на подмножества  $U_{k_1}$  и  $U_{k_2}$  следующим образом:

$$U_{k_1} = \{x \mid x \in U_k, x_m \leq [x_m^{(k)}]\}, \quad U_{k_2} = \{x \mid x \in U_k, x_m \geq [x_m^{(k)}] + 1\}.$$

Таким образом, для получения множеств  $U_{k_1}$  и  $U_{k_2}$  к ограничениям, определяющим  $U_k$ , добавляются соответственно неравенства  $x_m \leq [x_m^{(k)}]$  и  $x_m \geq [x_m^{(k)}] + 1$ . Каждому множеству  $U_{k_1}$  и  $U_{k_2}$  соответствует своя задача (*подзадача*) линейного программирования. При этом, очевидно, потери допустимых решений из множества  $U_k$  не происходит, т.е.  $U_k = U_{k_1} \cup U_{k_2}$ . Итак, правило ветвления также описано.

В общем случае каждая подзадача решается двойственным симплекс-методом. При этом дерево решений строится до тех пор, пока есть хотя бы одна перспективная вершина в том смысле, что имеется возможность улучшить текущее целочисленное решение. Если образуется вершина (подзадача) с нецелочисленным решением, которому соответствует значение целевой функции большее, чем для текущего целочисленного решения, то дальнейшее ветвление из этой вершины не имеет смысла.

В том случае, когда задача содержит только две переменные, то для наглядности решения можно использовать графический метод.

#### Пример 3.4.

Решить следующую целочисленную задачу линейного программирования

$$f(x) = -3x_1 - x_2 \rightarrow \min ;$$

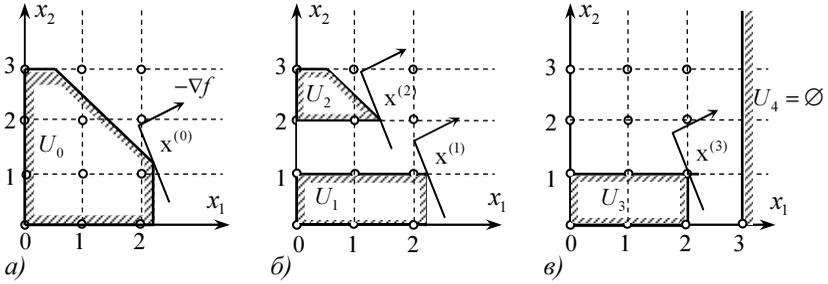
$$\begin{cases} x_1 + x_2 = 3.5, \\ x_1 \leq 2.25, x_2 \leq 3, \\ x_1, x_2 \geq 0; x_1, x_2 \in Z \end{cases}$$

методом ветвей и границ.

#### *Решение.*

Задача содержит две переменные, поэтому решения всех вспомогательных задач (подзадач) будем находить графически, не приводя их в каноническую форму.

Решив исходную задачу без учета требования целочисленности (рис. 3.3, а), получим:  $x^{(0)} = (2.25; 1.25)$ ,  $f_0 = -8$ .



Таким образом, нижняя граница  $f_0$  для вершины 0 равна  $-8$  и, так как решение  $x^{(0)}$  не является целочисленным, т.е. полным решением, считаем эту вершину активной и производим ветвление. В качестве нецелочисленной компоненты решения  $x^{(0)}$  выберем, например,  $x_2^{(0)} = 1.25$ . Задачи вида (3.29), соответствующие новым вершинам 1 и 2, имеют вид:

*Задача 1*

$$f(x) = -3x_1 - x_2 \rightarrow \min;$$

$$\begin{cases} x_1 + x_2 = 3.5; \\ x_1 \leq 2.25; \quad x_2 \leq 3; \\ x_2 \leq [1.25] = 1; \quad x_1, x_2 \geq 0. \end{cases}$$

*Задача 2*

$$f(x) = -3x_1 - x_2 \rightarrow \min;$$

$$\begin{cases} x_1 + x_2 = 3.5; \\ x_1 \leq 2.25; \quad x_2 \leq 3; \\ x_2 \geq [1.25] + 1 = 2; \quad x_1, x_2 \geq 0. \end{cases}$$

Решив эти задачи (рис. 3.3, б), получим:  $x^{(1)} = (2.25; 1)$ ,  $f_1 = -7.75$ ;  $x^{(2)} = (1.5; 2)$ ,  $f_2 = -6.5$ . Таким образом, нижние границы в вершинах 1 и 2 равны соответственно  $-7.75$  и  $-6.5$ , а решения  $x^{(1)}$  и  $x^{(2)}$  не являются целочисленными, т.е. полными. Поэтому обе эти вершины являются активными. Выберем для следующего ветвления вершину 1 с нецелочисленной компонентой  $x_1^{(1)} = 2.25$ . Получим две новые вершины 3 и 4, которым соответствуют следующие задачи вида (3.29):

*Задача 3*

$$f(x) = -3x_1 - x_2 \rightarrow \min;$$

$$\begin{cases} x_1 + x_2 = 3.5; \\ x_1 \leq 2.25; \quad x_2 \leq 1; \\ x_1 \leq [2.25] = 2; \quad x_1, x_2 \geq 0. \end{cases}$$

*Задача 4*

$$f(x) = -3x_1 - x_2 \rightarrow \min;$$

$$\begin{cases} x_1 + x_2 = 3.5; \\ x_1 \leq 2.25; \quad x_2 \leq 1; \\ x_1 \geq [2.25] + 1 = 3; \quad x_1, x_2 \geq 0. \end{cases}$$

Из рис. 3.3,в следует, что  $x^{(3)} = (2; 1)$ ,  $f_3 = -7$ , т.е. решение для вершины 3 является полным (так как оно целочисленно) и можно записать:  $x^{*(3)} = (2; 1)$ ,  $f_3^* = -7$ . Таким образом, вершина 3 является прозондированной, т.к. она удовлетворяет условию 1.

Далее  $U_4 = \emptyset$ , т.к. задача 4 содержит взаимно исключающие условия  $x_1 \leq 2.25$  и  $x_1 \geq 3$ , см. также рис. 3.3,в. Поэтому и вершина 4 прозондирована как удовлетворяющая условию 2.

Из активных вершин осталась только вершина 2. Однако  $f_2 = -6.5 > f_3^* = -7$ , т.е. она исключается вершиной 3, и ее также следует считать прозондированной по условию 3.

Итак, множество активных вершин пусто, поэтому в качестве решения исходной задачи следует взять лучшее из полных решений. Таким образом, имеем единственное полное решение  $(x^{*(3)}, f_3^*)$ ,  $f_3^* = -7$ . Следовательно,  $x^* = x^{*(3)} = (2; 1)$ ,  $f_{\min} = f_3^* = -7$ .

Схема решения задачи представляет собой бинарное дерево, показанное на рис. 3.4.

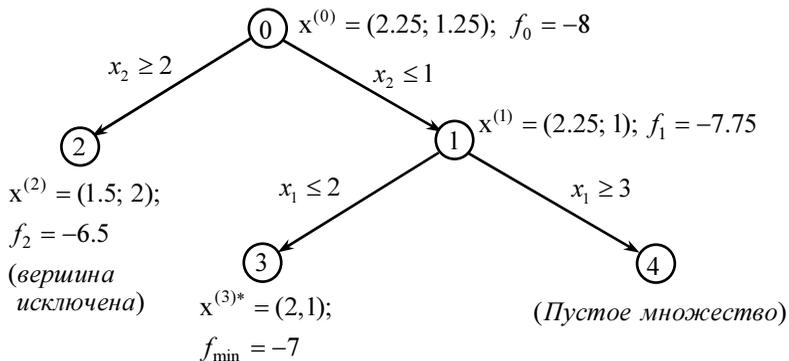


Рис. 3.4

При использовании метода ветвей и границ для решения целочисленной задачи линейного программирования возникают следующие вопросы:

- 1) как выбирать очередную вершину ветвления?
- 2) как определить нецелочисленную компоненту  $x_m^{(k)}$ , по которой будет производиться ветвление?

Универсальных ответов на эти вопросы, определяющих наилучший способ ветвления, не существует. Поэтому на практике используются какие-либо из следующих *правил выбора*:

1. Выбор вершины ветвления  $k$ .

а) выбирается та вершина  $k$ , которой соответствует минимальная из найденных нижних границ  $f_i$ . Некоторым обоснованием этого правила служит то соображение, что допустимое множество задачи с минимальной границей  $f_k$  может содержать и хорошее целочисленное решение. А чем раньше будет найдено такое решение, тем вероятнее, что меньше ветвлений придется совершить в дальнейшем из других полученных к данному моменту вершин, так как часть из них будет исключена вершиной с этим решением;

б) выбираемой вершине  $k$  соответствует вспомогательная задача линейного программирования, решавшаяся последней. Это несколько уменьшает объем памяти, необходимой для хранения промежуточных симплекс-таблиц.

2. Выбор нецелочисленной компоненты  $x_m^{(k)}$  ветвления из вершины  $k$ :

а)  $x_m^{(k)}$  выбирается в соответствии с заранее приписанными всем переменным  $x_j$  приоритетами. Можно использовать различные критерии приоритетности. Например, считать, что чем больше модуль коэффициента  $c_j$  в выражении  $f(x) = \sum_j c_j x_j$ , тем существеннее вклад переменной  $x_j$  в показатель качества решения  $f(x)$  и, следовательно, выше приоритет  $x_j$ ;

б) используется произвольное правило выбора. Например, выбирается компонента с минимальным номером или с максимальной дробной частью.

**Алгоритм метода ветвей и границ** решения целочисленной задачи линейного программирования:

1. Определение решения  $(x^{(0)}, f_0)$  исходной задачи (3.27) без учета требования целочисленности, т.е. задачи 0. Задача 0 считается разрешимой. Пусть  $V$  – множество активных вершин.

Если  $(x^{(0)}, f_0)$  является полным решением, т.е. если  $x^{(0)}$  удовлетворяет условию целочисленности, то решение является оптимальным, т.е.  $x^* = x^{(0)}$ ,  $f_{\min} = f_0$ ,  $V = \emptyset$ , задача решена. Иначе – поло-

жить  $f_* = +\infty$  (равным достаточно большому числу) и включить вершину 0 (задачу 0) в множество  $V$ . Перейти к шагу 2.

2. Если  $V = \emptyset$ , то перейти к шагу 8, иначе – выбрать вершину  $k \in V$ , исключить ее из  $V$  и перейти к шагу 3.

3. Выбрать переменную ветвления  $x_m^{(k)}$  из вершины  $k$  и породить две новые вершины (задачи)  $k_1$  и  $k_2$ , которым соответствуют допустимые подмножества  $U_{k_1}$  и  $U_{k_2}$ :

$$U_{k_1} = \{x \mid x \in U_k \text{ и } x_j \leq [x_j]\};$$

$$U_{k_2} = \{x \mid x \in U_k \text{ и } x_j \geq [x_j] + 1\}.$$

Положить  $i = 1$  и перейти к шагу 4.

4. Если допустимое множество  $U_{k_i}$  задачи  $k_i$  не пусто, то перейти к шагу 5, иначе – к шагу 7.

5. Найти решение  $(x^{(k_i)}, f_{k_i})$  задачи  $k_i$ . Если  $f_{k_i} \geq f_*$  (т.е. вершина  $k_i$  исключена), то перейти к шагу 7, иначе – к шагу 6.

6. Если  $x^{(k_i)}$  удовлетворяет условию целочисленности, т.е. является полным решением, то положить  $f_* = f_{k_i}$ ,  $x_* = x^{(k_i)}$  иначе – включить вершину  $k_i$  в активное множество  $V$ . Перейти к шагу 7.

7. Если  $i < 2$ , то положить  $i = 2$  и перейти к шагу 4, иначе – перейти к шагу 2.

8. Положить  $x^* = x_*$ ,  $f_{\min} = f_*$  и завершить поиск решения. Если  $f_* = +\infty$ , т.е. остается равным достаточно большому числу, то исходная задача не имеет решения.

### 3.4. Булева задача линейного программирования

Задачи целочисленной оптимизации, переменные которых могут принимать только два значения – 0 или 1, называются *булевыми* (или *бинарными*) *задачами*, а их переменные – *булевыми* (или *бинарными*) *переменными*. Значение 0 бинарной переменной  $x$ , как правило, означает отсутствие определенного свойства, а значение 1 – наличие этого свойства.

Общая форма математической модели булевой задачи имеет следующий вид:

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \min; \quad (3.30)$$

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i = \overline{1, m}; \quad (3.31)$$

$$x_j \in \{0, 1\}, \quad j = \overline{1, n}. \quad (3.32)$$

Рассмотрим один из методов: *аддитивный алгоритм* (метод Балаша) решения булевых задач.

Представленный ниже алгоритм опирается на идеи метода ветвей и границ. В этом алгоритме вычислительные операции включают лишь операции сравнения, сложения и вычитания.

Для применения алгоритма математическая модель (3.30) – (3.32) приводится к следующему каноническому виду :

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad c_j \geq 0, \quad j = \overline{1, n};$$

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

$$x_j \in \{0, 1\}, \quad j = \overline{1, n}; \quad y_i \geq 0, \quad i = \overline{1, m}.$$

Если в (3.30) есть  $c_j < 0$ , то с помощью замены  $x_j = 1 - x'_j$  (где  $x'_j$  – бинарная переменная) в целевой функции и ограничениях задача приводится к соответствующему виду.

Основная идея аддитивного алгоритма заключается в переборе возможных решений исходной задачи. В процедуре перебора используются специальные *тесты*, что позволяет многие решения отвергать без непосредственной проверки. В итоге для определения оптимального решения требуется непосредственное рассмотрение лишь части из возможных решений. В процессе решения строится дерево *подзадач*.

Бинарная переменная называется *свободной переменной* в некоторой вершине дерева, если ее значение не зафиксировано ни на одной из ветвей, ведущих в эту вершину, и принимается равным нулю. Обозначим через  $s$  – множество индексов свободных переменных, через  $u$  – множество индексов фиксированных переменных.

*Частичным решением* называется упорядоченное множество бинарных переменных, в котором значения одной или нескольких переменных фиксированы и равны нулю или единице. Каждому частичному решению соответствует определенная вершина дерева, порожденного исходной задачей. Каждое последующее частичное решение может быть получено из предыдущего. Частичное решение считается *прозондированным*, если оно не может привести к улучшению значения целевой функции и получению допустимого решения.

Зондирование частичных решений осуществляется с помощью приведенных ниже тестов.

*Тест 1.* Если  $x_k$  – свободная переменная, и хотя бы для одного значения  $y_i < 0$  соответствующий коэффициент  $a_{ik} \geq 0$ , то присвоение  $x_k$  значения 1 не может привести к получению допустимого решения, и значение  $x_k$  должно остаться равным нулю.

*Тест 2.* Если  $x_k$  – свободная переменная, и выполняется неравенство  $c_k + f \geq \bar{f}$ , где  $f$  – текущее значение целевой функции,  $\bar{f}$  – текущее значение верхней границы значения целевой функции, то присвоение  $x_k$  значения 1 не может уменьшить значения целевой функции и значение  $x_k$  должно остаться равным нулю. Этот тест проверяется, если текущее решение становится допустимым.

*Тест 3.* Пусть  $s$  – текущее множество индексов свободных переменных, для которых не выполняются условия тестов 1 и 2. Если, по крайней мере, для одного значения  $y_i < 0$  выполняется неравенство

$$\sum_{j \in s} \min\{0, a_{ij}\} > y_i,$$

то значения всех переменных из множества  $s$  должны остаться равными нулю, поскольку никакой другой допустимый набор их фиксированных значений не приведет к получению допустимого решения. Частичное решение в этом случае считается прозондированным.

*Тест 4.* Если  $s \neq \emptyset$  и  $v_k = \max\{v_j\}$ , где  $v_j = \sum_{i=1}^m \min\{0, y_i - a_{ij}\}$ ,

то переменную  $x_k$  следует выбирать в качестве переменной, инициирующей процесс ветвления. Если  $v_k = 0$ , то объединение условия  $x_k = 1$  с частичным решением  $u$  обеспечивает получение допустимого решения с меньшим значением целевой функции  $\bar{f}$ . При этом частичное решение  $u$ , построенное путем приписывания ему справа элемента  $\{k\}$ , считается прозондированным. В противном случае проверяются остальные тесты.

И, наконец, рассмотрим правило построения последующего частичного решения на основе прозондированного. Процесс решения задачи начинается с частичного решения при  $u^{(0)} = \emptyset$ , означающего, что все бинарные переменные являются свободными. Введение условия  $x_k = 1$  влечет за собой частичное решение с  $u^{(1)} = \{k\}$ , которое, например, может быть допустимым решением. Следующее час-

точное решение строится путем замены знака крайнего правого положительного элемента  $u^{(1)}$ , т.е.  $u^{(2)} = \{-k\}$ , что соответствует  $x_k = 0$ . Процедура замены знака крайнего правого положительного элемента частичного решения называется *процедурой возврата*.

Кратко схему работы алгоритма можно сформулировать следующим образом. На первом шаге все исходные переменные полагаются равными нулю. Это представляется логичным, поскольку все  $c_j \geq 0$ . Пусть исходное решение оказывается недопустимым, т.е. некоторые дополнительные переменные принимают отрицательные значения, тогда следует необходимость присвоения некоторым переменным значения 1.

На следующем шаге с использованием тестов осуществляется выбор тех переменных, которым надлежит присвоить значение 1.

Целью всей процедуры является получение допустимого решения  $x^*$ , которому соответствует  $f_{\min}$ .

Итак, аддитивный алгоритм основан на последовательной проверке представленных *тестов* 1 – 4 с использованием частичных решений.

### Пример 3.5.

Решить следующую бинарную задачу с использованием аддитивного алгоритма:

$$f(x) = 4x_1 + 8x_2 + 5x_3 + 2x_4 \rightarrow \min; \quad (3.33)$$

$$\begin{cases} -2x_1 + x_2 - x_3 + 3x_4 \leq 3, \\ 8x_1 - 6x_2 - 3x_3 - 4x_4 \leq -2, \\ 4x_1 - 4x_2 - x_3 - 3x_4 \leq -4; \end{cases} \quad (3.34)$$

$$x_j \in \{0, 1\}, \quad j = \overline{1, 4}. \quad (3.35)$$

**Решение.** После введения дополнительных переменных  $y_i \geq 0, i = 1, 2, 3$  исходная задача (3.33) – (3.35) приводится к виду

$$\begin{cases} f(x) = 4x_1 + 8x_2 + 5x_3 + 2x_4 \rightarrow \min \\ y_1 = 3 - (-2x_1 + x_2 - x_3 + 3x_4) \\ y_2 = -2 - (8x_1 - 6x_2 - 3x_3 - 4x_4) \\ y_3 = -4 - (4x_1 - 4x_2 - x_3 - 3x_4) \end{cases}$$

**Итерация 0.** Начальные условия:  $u^{(0)} = \emptyset$ ,  $s^{(0)} = \{1, 2, 3, 4\}$ ;  $x^{(0)} = (0, 0, 0, 0)$ ,  $\bar{f} = \infty$ ,  $y_1^{(0)} = 3$ ,  $y_2^{(0)} = -2$ ,  $y_3^{(0)} = -4$ .

Используя тесты 1–4 выбираем переменную, которой можно присвоить значение 1.

С помощью *теста 1* исключается из рассмотрения переменная  $x_1$ , т.к.  $y_2 < 0$  и  $a_{21} > 0$  (или  $y_3 < 0$  и  $a_{31} > 0$ ).

*Тест 2* не имеет силы, т.к. исходное решение не является допустимым.

*Тест 3* (анализ по строкам) свидетельствует о том, что остальные свободные переменные не исключаются из рассмотрения, т.к. при  $s^{(0)} = \{2, 3, 4\}$

$$y_2 = -2 < 0: \sum_{j \in s^{(0)}} \min\{0, a_{2j}\} = \min\{0, 8\} + \min\{0, -6\} + \\ + \min\{0, -3\} + \min\{0, -4\} = -13 < y_2;$$

$$y_3 = -4 < 0: \sum_{j \in s^{(0)}} \min\{0, a_{3j}\} = \min\{0, 4\} + \min\{0, -4\} + \\ + \min\{0, -1\} + \min\{0, -3\} = -8 < y_3;$$

Если хотя бы для одного  $y_i < 0$  выполнялось условие  $\sum_{j \in s^{(0)}} \min\{0, a_{ij}\} > y_i$ , то вершина считалась бы прозондированной и выполнялся бы возврат на один шаг.

*Тест 4* (анализ по столбцам) позволяет выбрать переменную, иницирующую ветвление из текущей вершины.

$$\text{Определим } v_k = \max\{v_j\} \text{ при } v_j = \sum_{i=1}^m \min\{0, y_i - a_{ij}\}:$$

$$v_2 = \min\{0, 3 - 1\} + \min\{0, -2 - (-6)\} + \min\{0, -4 - (-4)\} = 0;$$

$$v_3 = \min\{0, 3 - (-1)\} + \min\{0, -2 - (-3)\} + \min\{0, -4 - (-1)\} = -3;$$

$$v_4 = \min\{0, 3 - 3\} + \min\{0, -2 - (-4)\} + \min\{0, -4 - (-3)\} = -1;$$

$$v_k = \max\{v_j\} = \max\{0, -3, -1\} = v_2 = 0. \text{ Следовательно, } x_2 = 1.$$

Получено допустимое решение  $x^{(1)} = (0, 1, 0, 0)$ ,  $f = 8$ . Если тест 4 дает результат  $v_k < 0$ , то осуществляется ветвление относительно  $x_k$ , т.е. исследуются две подзадачи при  $x_k = 1$  и  $x_k = 0$  соответственно.

Итерация 1. Начальные условия:  $u^{(1)} = \{2\}$ ,  $s^{(1)} = \{1, 3, 4\}$ ,  $x^{(1)} = (0, 1, 0, 0)$ ;  $f = 8$ ;  $\bar{f} < f$ , следовательно,  $\bar{f} = 8$ . Значения дополнительных переменных:  $y_1^{(1)} = 2$ ,  $y_2^{(1)} = 4$ ,  $y_3^{(1)} = 0$ .

Так как все  $y_i^{(1)} \geq 0$ , то получено допустимое решение, вершина 1 (подзадача 1) прозондирована. Используется процедура возврата.

На рис.3.1 изображено дерево ветвления решения задачи. Каждой подзадаче (итерации) соответствует определенная вершина дерева

Итерация 2. Начальные условия:  $u^{(2)} = \{-2\}$ ,  $s^{(2)} = \{1, 3, 4\}$ ,  $x^{(2)} = (0, \underline{0}, 0, 0)$ ;  $f = 0$ ;  $\bar{f} = 8$ . Исследуется следующая подзадача:

$$\begin{cases} y_1 = 3 - (-2x_1 - x_3 + 3x_4); \\ y_2 = -2 - (8x_1 - 3x_3 - 4x_4); \\ y_3 = -4 - (4x_1 - x_3 - 3x_4); \end{cases}$$

где  $y_1^{(2)} = 3$ ,  $y_2^{(2)} = -2$ ,  $y_3^{(2)} = -4$ .

Согласно *тесту 1* исключается из рассмотрения переменная  $x_1$ , т.к.  $y_2 < 0$  и  $a_{21} > 0$

*Тест 2* не используется, т.к. исходное решение не является допустимым.

*Тест 3* (анализ по строкам) свидетельствует о том, что остальные свободные переменные не исключаются из рассмотрения, т.к. при  $s^{(2)} = \{1, 3, 4\}$

$$y_2 = -2 < 0:$$

$$\sum_{j \in s^{(2)}} \min\{0, a_{2j}\} = \min\{0, 8\} + \min\{0, -3\} + \min\{0, -4\} = -7 < y_2;$$

$$y_3 = -4 < 0:$$

$$\sum_{j \in s^{(2)}} \min\{0, a_{3j}\} = \min\{0, 4\} + \min\{0, -1\} + \min\{0, -3\} = -4 = y_3;$$

*Тест 4* (анализ по столбцам) позволяет выбрать переменную, иницирующую ветвление из текущей вершины.

$$\text{Определим } v_k = \max\{v_j\} \text{ при } v_j = \sum_{i=1}^m \min\{0, y_i - a_{ij}\}:$$

$$v_3 = \min\{0, 3 - (-1)\} + \min\{0, -2 - (-3)\} + \min\{0, -4 - (-1)\} = -3;$$

$$v_4 = \min\{0, 3 - 3\} + \min\{0, -2 - (-4)\} + \min\{0, -4 - (-3)\} = -1;$$

$v_k = \max\{v_j\} = \max\{-3, -1\} = v_4 = -1$ . Следовательно, ветвление осуществляется относительно переменной  $x_4$ .

Итерация 3. Начальные условия:  $u^{(3)} = \{-2; 4\}$ ,  $s^{(3)} = \{1, 3\}$ ,  $x^{(3)} = (0, \underline{0}, 0, 1)$ ;  $f = 2$ ;  $\bar{f} = 8$ . Исследуется следующая подзадача:

$$\begin{cases} y_1 = 0 - (-2x_1 - x_3); \\ y_2 = 2 - (8x_1 - 3x_3); \\ y_3 = -1 - (4x_1 - x_3); \end{cases}$$

где  $y_1^{(3)} = 0$ ,  $y_2^{(3)} = 2$ ,  $y_3^{(3)} = -1$ .

Согласно *тесту 1* исключается из рассмотрения переменная  $x_1$ , т.к.  $y_3 < 0$  и  $a_{31} > 0$ , поэтому  $x_1 = 0$

*Тест 2* не используется, т.к. исходное решение не является допустимым.

*Тест 3* (анализ по строкам) свидетельствует о том, что остальные свободные переменные не исключаются из рассмотрения, т.к. при  $s^{(3)} = \{1, 3\}$

$$y_3 = -1 < 0:$$

$$\sum_{j \in s^{(3)}} \min\{0, a_{3j}\} = \min\{0, 4\} + \min\{0, -1\} = -1 = y_3;$$

*Тест 4* (анализ по столбцам). Осталась только одна переменная  $x_3$ , которую проверяем на допустимость:

$$v_k = v_3 = \min\{0, 0 - (-1)\} + \min\{0, 2 - (-3)\} + \min\{0, -1 - (-1)\} = 0.$$

При  $x_3 = 1$  получено допустимое решение  $x^{(3)} = (0, \underline{0}, 1, 1)$ ,  $f = 7$ .

*Итерация 4.* Начальные условия:  $u^{(4)} = \{-2, 4, 3\}$ ,  $s^{(4)} = \{1\}$ ,  $x^{(4)} = (0, \underline{0}, 1, 1)$ ;  $f = 7$ ;  $\bar{f} = 8$ . Так как  $f < \bar{f}$ , то  $\bar{f} = 7$ . Все дополнительные переменные  $y_i^{(4)} \geq 0$ :  $y_1^{(4)} = 1$ ,  $y_2^{(4)} = 5$ ,  $y_3^{(4)} = 0$ . Следовательно, получено допустимое решение и вершина 4 прозондирована. Возврат на 1 шаг.

*Итерация 5.* Начальные условия:  $u^{(5)} = \{-2, 4, -3\}$ ,  $s^{(4)} = \{1\}$ ,  $x^{(4)} = (0, \underline{0}, \underline{0}, 1)$ ;  $f = 2$ ,  $\bar{f} = 7$ ; Исследуется следующая подзадача:

$$\begin{cases} y_1 = 0 - (-2x_1); \\ y_2 = 2 - (8x_1); \\ y_3 = -4 - (4x_1); \end{cases}$$

где  $y_1^{(5)} = 0$ ,  $y_2^{(5)} = 2$ ,  $y_3^{(5)} = -1$ .

Согласно *тесту 1* решение прозондировано, т.к.  $y_3 < 0$  и  $a_{31} > 0$ , значит переменная  $x_1$  не может равняться 1. Решение прозондировано, возврат в подзадачу 2 при условии  $x_4 = 0$ .

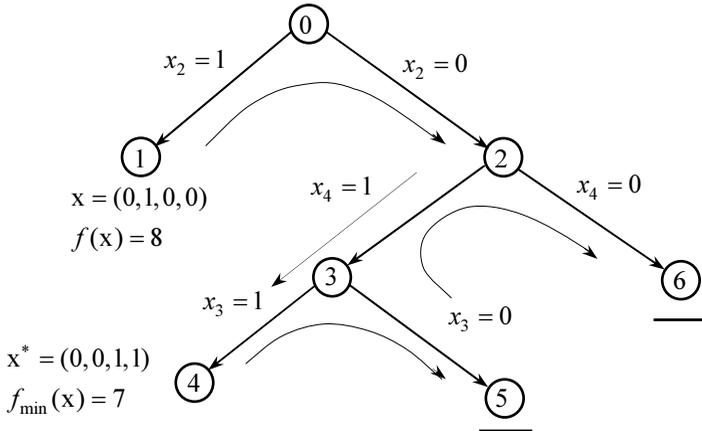


Рис. 3.5.

Итерация 6. Начальные условия:  $u^{(6)} = \{-2, -4\}$ ,  $s^{(6)} = \{1, 3\}$ ,  $x^{(6)} = (0, \underline{0}, 0, \underline{0})$ ;  $f = 0$ ;  $\bar{f} = 7$ . Исследуется следующая подзадача:

$$\begin{cases} y_1 = 3 - (-2x_1 - x_3); \\ y_2 = -2 - (8x_1 - 3x_3); \\ y_3 = -4 - (4x_1 - x_3); \end{cases}$$

где  $y_1^{(6)} = 3$ ,  $y_2^{(2)} = -2$ ,  $y_3^{(2)} = -4$ .

Согласно *тесту 1* исключается из рассмотрения переменная  $x_1$ , т.к.  $y_2 < 0$  и  $a_{21} > 0$ .

*Тест 2* не используется, т.к. исходное решение не является допустимым.

Проверяем условия по *тесту 3* (анализ по строкам):

$$y_2 = -2 < 0: \sum_{j \in s^{(6)}} \min\{0, a_{2j}\} = \min\{0, 8\} + \min\{0, -3\} = -3 < y_2;$$

$$y_3 = -4 < 0: \sum_{j \in s^{(6)}} \min\{0, a_{3j}\} = \min\{0, 4\} + \min\{0, -1\} = -1 > y_3.$$

Последнее соотношение свидетельствует о том, что решение прозондировано. Граф–дерево решений представлен на рис. 3.5.

Таким образом, оптимальное решение исходной задачи определяется нижней границей значения целевой функции:

$$x^* = (0, 0, 1, 1); f_{\min} = 7.$$

Из описания и применения метода ветвей и границ ясно, что для его применения существенным является выполнение только следующих условий:

1. Известно *правило ветвления*, т.е. разбиения множества допустимых решений, представляемого вершиной, на несколько попарно непересекающихся подмножеств.

2. Имеется *алгоритм получения нижней границы* целевой функции на любом допустимом подмножестве.

Поэтому метод ветвей и границ можно использовать для решения не только целочисленной задачи линейного программирования, но и многих других задач, для которых выполняются указанные условия.

Эффективное решение многих практических задач дискретного программирования большой размерности с помощью алгоритмов метода ветвей и границ проблематично. Поэтому при решении сложных задач имеет смысл использовать различные приемы для окончания вычислительного процесса с целью нахождения за некоторое фиксированное время допустимого приближенного решения. Если к тому же удастся приближенное решение сопровождать и определенной хорошей оценкой точности получаемого решения, то применение метода может иметь значительную практическую ценность.

В ряде работ были предприняты попытки теоретического обоснования метода ветвей и границ [3]. Исследования показали, что встречаются задачи дискретной оптимизации, для которых объем вычислений близок к полному перебору.

### 3.5. Метод динамического программирования

Многие математические модели дискретных и непрерывных задач оптимизации можно сформулировать в виде *многошаговой задачи оптимизации*, в рамках которой процесс поиска решения представляется как последовательность некоторых этапов (шагов). Для решения таких задач используется *метод динамического программирования*, рассматриваемый в данном разделе.

Динамическое программирование позволяет решать задачи, дискретные по самой своей природе. Например, в задачах оптимального выбора динамическое программирование выступает как метод оптимального перебора вариантов.

### ***Постановка многошаговой задачи оптимизации***

Предположим, что состояние некоторого *процесса* описывается вектором  $x = (x_1, \dots, x_n)$ , т.е. точкой  $x$  пространства  $R^n$ , которое называется *фазовым пространством*.

Будем считать, что процесс является  $N$ -шаговым, т.е. его эволюция происходит в  $N$  этапов по следующей схеме:

$$\begin{array}{ccccccc}
 x^{(0)} & \rightarrow & \dots & \rightarrow & x^{(k-1)} & \rightarrow & x^{(k)} & \rightarrow & \dots & \rightarrow & x^{(N)} \\
 \text{Начальное} & & & & \text{\textit{k}-ый шаг} & & & & & & \text{Конечное} \\
 \text{состояние} & & & & & & & & & & \text{состояние}
 \end{array}$$

Переход между состояниями на  $k$ -ом шаге совершается в соответствии с *уравнением состояний*

$$x^{(k)} = F^{(k)}(x^{(k-1)}, u^{(k)}),$$

где  $u^{(k)} \in R^m$  –  $m$ -мерный *вектор управления*, выбираемый на этом шаге, а  $F^{(k)}(x^{(k-1)}, u^{(k)})$  – заданная  $n$ -мерная вектор-функция аргументов  $x \in R^n$  и  $u \in R^m$ . Предполагается, что состояние, в которое переходит процесс на  $k$ -м шаге, полностью определяется только начальным состоянием  $x^{(k-1)}$  этого шага и выбранным на нем управлением  $u^{(k)}$ , но не зависит от «предыстории» процесса до  $k$ -го шага.

Числовым критерием оптимизации  $k$ -го шага является частичная целевая функция  $f_k(x^{(k-1)}, u^{(k)})$  этого шага, а весь процесс характеризуется целевой функцией

$$f(\hat{x}, \hat{u}) = \sum_{k=1}^N f_k(x^{(k-1)}, u^{(k)}),$$

где  $\hat{x} = \{x^{(0)}, \dots, x^{(N)}\}$  – набор состояний, называемый *фазовой траекторией процесса*, а  $\hat{u} = \{u^{(0)}, \dots, u^{(N)}\}$  – совокупность векторов управлений, которая называется *управлением процессом*.

На фазовую траекторию и управление процессом налагаются ограничения

$$x^{(k)} \in X_k, \quad k = 0, 1, \dots, N; \quad u^{(k)} \in U_k(x^{(k-1)}), \quad k = \overline{1, N},$$

где  $X_k \subset R^n$  и  $U_k(x^{(k-1)}) \subset R^m$  – заданные множества, причем множество  $U_k$  зависит, вообще говоря, от начального состояния  $k$ -го шага  $x^{(k-1)}$ . Ограничения на начальное и конечное состояния процесса  $x^{(0)} \in X_0$  и  $x^{(N)} \in X_N$  называют *начальными* и *конечными условиями*, причем во многих случаях множества  $X_0$  и  $X_N$  содержат по одной точке (начало и конец фазовой траектории).

Таким образом, многошаговая задача оптимизации формулируется в следующем виде:

$$f(\hat{x}, \hat{u}) = \sum_{k=1}^N f_k(x^{(k-1)}, u^{(k)}) \rightarrow \min (\max); \quad (3.36)$$

$$x^{(k)} = F^{(k)}(x^{(k-1)}, u^{(k)}), \quad k = 1, 2, \dots, N; \quad (3.37)$$

$$x^{(k)} \in X_k, \quad k = 0, 1, \dots, N; \quad (3.38)$$

$$u^{(k)} \in U_k(x^{(k-1)}), \quad k = 1, 2, \dots, N, \quad (3.39)$$

т.е. в ней требуется найти оптимальное управление процессом  $\hat{u}^* = \{u^{(0)*}, \dots, u^{(N)*}\}$  и соответствующую ему оптимальную фазовую траекторию  $\hat{x}^* = \{x^{(0)*}, \dots, x^{(N)*}\}$ , удовлетворяющие ограничениям (3.37) – (3.39) и доставляющие минимум (максимум) целевой функции (3.36).

Приведем пример построения математической модели, имеющей вид многошаговых задач оптимизации.

**Пример 3.6.** Целочисленная задача линейного программирования.

Сформулировать целочисленную задачу линейного программирования с одним линейным ограничением:

$$f(y) = \sum_{k=1}^N c_k y_k \rightarrow \min ,$$

$$\sum_{k=1}^N a_k y_k \leq b , \quad (3.40)$$

$$y_k \geq 0, \quad y_k \in Z, \quad k = \overline{1, N}$$

где  $a_k \geq 0$ ,  $k = \overline{1, N}$ ,  $b > 0$ , как многошаговую задачу оптимизации.

**Решение.** Представим процесс решения задачи в виде последовательного выбора переменных  $y_k$ , считая их управлениями:  $y_k = u^{(k)}$ ,

$k = \overline{1, N}$ . Введем одномерный вектор состояния  $x^{(k)}$  по формулам  $x^{(0)} = b$ ,  $x^{(k)} = b - \sum_{j=1}^k a_j u^{(j)}$ ,  $k = \overline{1, N}$ . Тогда уравнение состояний (3.40) примет вид  $x^{(k)} = x^{(k-1)} - a_k u^{(k)}$ .

Из ограничения (3.40) с учетом неотрицательности  $a_k$  следует:  $x^{(k)} \geq 0$ . Кроме того, по условию исходной задачи  $u^{(k)} \geq 0$ ,  $u^{(k)} \in Z$ .

Таким образом, получаем следующую  $N$ -шаговую задачу оптимизации:

$$\begin{aligned} f(\hat{x}, \hat{u}) &= \sum_{k=1}^N c_k u^{(k)} \rightarrow \min ; \\ x^{(k)} &= x^{(k-1)} - a_k u^{(k)}, \\ x^{(0)} &= b, \quad x^{(k)} \geq 0, \\ u^{(k)} &\geq 0, \quad u^{(k)} \in Z, \quad k = \overline{1, N}. \end{aligned}$$

В примере 3.6 к многошаговой схеме (3.36) – (3.39) сведена дискретная задача оптимизации. Однако дискретность не является существенным обстоятельством при построении многошаговых математических моделей. Так, если в дискретной задаче линейного программирования из примера 3.6 отбросить условие целочисленности, то она станет непрерывной, а соответствующая многошаговая модель почти не изменится.

Из примера видно, что для построения многошаговой математической модели необходимо представить (во многих случаях искусственно) поиск решения исходной задачи как последовательность некоторых этапов выбора частичных решений.

### ***Принцип оптимальности Беллмана***

Может показаться, что при решении многошаговой задачи (3.36)–(3.38) достаточно на каждом шаге выбирать управление  $u^{(k)}$  из условия оптимальности данного шага, и тогда аддитивная целевая функция  $f(\hat{x}, \hat{u})$  всего процесса примет экстремальное значение.

Однако такой подход является ошибочным, так как выбор управления  $u^{(k)}$ , направленный на оптимизацию только  $k$ -го шага, определяет, как правило, не лучшее начальное состояние  $x^{(k)}$  для следующих за ним шагов, и эти шаги не будут оптимальными, как бы ни выбирались на них управления  $u^{(k+1)}, \dots, u^{(N)}$ .

Из приведенных рассуждений следует вывод о том, что *при поиске оптимального решения многошаговой задачи оптимизации (3.36) –*

(3.39) выбор управления  $u^{(k)}$  на каждом шаге, независимо от его начального состояния  $x^{(k-1)}$ , должен быть направлен на оптимизацию не только данного, но и всех оставшихся шагов. Этот вывод является одной из формулировок так называемого принципа оптимальности и связан с важным свойством оптимальной фазовой траектории многошагового процесса, вытекающим из следующей теоремы.

**Теорема 3.2.** Пусть  $u^* = \{u^{(1)*}, \dots, u^{(N)*}\}$  – оптимальное управление, а  $\hat{x}^* = \{x^{(0)*}, \dots, x^{(N)*}\}$  – оптимальная фазовая траектория в задаче (3.36) – (3.39). Тогда при  $k = 2, \dots, N$  управление  $\hat{u}^*(k) = \{u^{(k)*}, \dots, u^{(N)*}\}$  и траектория  $\hat{x}^*(k) = \{x^{(k-1)*}, \dots, x^{(N)*}\}$  будут оптимальными для последних  $N - k + 1$  шагов, начинающихся из состояния  $x^{(k-1)*}$ .

**Доказательство.** Предположим обратное, т.е. что  $\hat{u}^*(k)$  не является оптимальным управлением завершающих  $N - k + 1$  шагов. Тогда существует такое управление  $\hat{u}_*(k) = \{u_*^{(k)}, \dots, u_*^{(N)}\}$  с соответствующей фазовой траекторией  $\hat{x}_*(k) = \{x_*^{(k-1)}, x_*^{(k)}, \dots, u_*^{(N)}\}$  на этих шагах, что целевая функция последних  $N - k + 1$  шагов  $f^{(k)}(\hat{x}(k), \hat{u}(k)) = \sum_{i=k}^N f_i(x^{(i-1)}, u^{(i)})$  принимает для этих управления и фазовой траектории меньшее значение, т.е.  $f^{(k)}(\hat{x}_*(k), \hat{u}_*(k)) < f^{(k)}(\hat{x}^*(k), \hat{u}^*(k))$ .

Положим:

$$\begin{aligned} \hat{u}_* &= \{u_*^{(1)}, \dots, u_*^{(k-1)}, u_*^{(k)}, \dots, u_*^{(N)}\}, \\ \hat{x}_* &= \{x_*^{(0)}, \dots, x_*^{(k-1)}, x_*^{(k)}, \dots, u_*^{(N)}\}. \end{aligned}$$

Тогда:

$$\begin{aligned} f^{(k)}(\hat{x}_*, \hat{u}_*) &= \sum_{i=1}^{k-1} f_i(x^{(i-1)*}, u^{(i)*}) + f^k(\hat{x}_*(k), \hat{u}_*(k)) < \\ &< \sum_{i=1}^{k-1} f_i(x^{(i-1)*}, u^{(i)*}) + f^k(\hat{x}^*(k), \hat{u}^*(k)), \end{aligned}$$

т.е. управление  $\hat{u}^*$  и фазовая траектория  $\hat{x}^*$  не являются оптимальными, что противоречит условию теоремы.

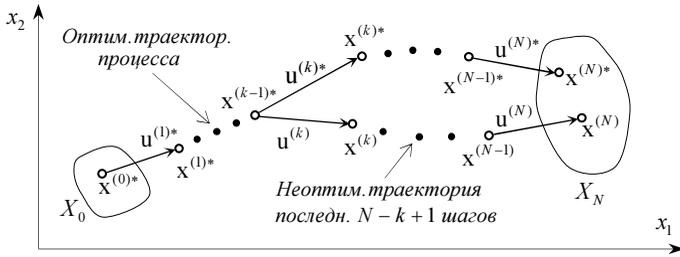


Рис. 3.6

Геометрическая интерпретация сформулированного в теореме 3.2 свойства оптимальной фазовой траектории дана на рис. 3.6. Предположим, что состояние многошагового процесса характеризуется точкой  $x = (x_1, x_2)$  плоскости и эта точка в соответствии с выбираемыми управлениями перемещается из начального состояния  $x^{(0)} \in X_0$  в конечное состояние  $x^{(N)} \in X_N$ , проходя промежуточные состояния  $x^{(1)}, \dots, x^{(N-1)}$ . Пусть фазовая траектория  $\hat{x}^* = \{x^{(0)*}, \dots, x^{(N)*}\}$  является оптимальной, тогда и любая ее завершающая часть, начинающаяся из промежуточного состояния  $x^{(k-1)*}$ , будет оптимальной на последних  $N - k + 1$  шагах. А всякая другая траектория из того же состояния, во-обще говоря, не является оптимальной на этих шагах и, следовательно, *не может быть частью оптимальной траектории всего процесса.*

### **Метод динамического программирования**

Рассмотренное выше свойство оптимальной траектории  $N$ -шагового процесса подсказывает следующую стратегию его оптимизации: сначала оптимизируется последний,  $N$ -ый шаг, затем два последних шага, затем три последних шага и т.д., вплоть до первого, начального шага. Такой подход лежит в основе метода динамического программирования. Опишем его дискретный вариант.

Прежде всего отметим, что в формулировке многошаговой задачи оптимизации (3.36) – (3.39) ограничения (3.38) при  $k = \overline{1, N}$  на фазовую траекторию можно включить в ограничения на выбор управлений, заменив соотношения (3.38) для  $k \neq 0$  и (3.39) следующим эквивалентным ограничением:

$$u^{(k)} \in \overline{U}_k(x^{(k-1)}) = \{u^{(k)} \in \overline{U}_k(x^{(k-1)}) \mid F^{(k)}(x^{(k-1)}, u^{(k)}) \in X_k\},$$

$$k = \overline{1, N}.$$

Будем последовательно строить оптимальное управление и траекторию  $N$ -шагового процесса от конца к началу, т.е. рассмотрим сначала оптимизацию только  $N$ -го шага, затем только шагов  $N-1$  и  $N$  и т.д., пока не дойдем до первого шага процесса.

При реализации такого подхода возникает следующее затруднение. Приступая к оптимизации завершающих шагов процесса, начиная с  $k$ -го мы не знаем, из какого состояния  $x^{(k-1)}$  эти шаги должны начинаться, так как искомая оптимальная траектория  $\hat{x}^*$  а значит, и ее промежуточные состояния еще неизвестны. Поэтому приходится просматривать все возможные начальные состояния  $x^{(k-1)} \in X_{k-1}$  завершающей части процесса и для каждого из них строить оптимальное управление  $\hat{u}^*(k) = \{u^{(k)*}, \dots, u^{(N)*}\}$  и оптимальную траекторию  $\hat{x}^*(k) = \{x^{(k-1)*}, x^{(k)*}, \dots, x^{(N)*}\}$  последних  $N-k+1$  шагов.

Рассмотрим подробнее определение  $\hat{u}^*(k)$  и  $\hat{x}^*(k)$  для произвольного начального состояния  $x^{(k-1)} \in X_{k-1}$  завершающих шагов процесса, начинающихся с  $k$ -го шага. При  $\hat{u}(k) = \hat{u}^*(k)$ ,  $\hat{x}(k) = \hat{x}^*(k)$  целевая функция  $f^{(k)}(\hat{x}(k), \hat{u}(k)) = \sum_{i=k}^N f_i(x^{(i-1)}, u^{(i)})$  рассматриваемых шагов принимает минимальное значение  $f^{(k)*}$ , зависящее от  $x^{(k-1)}$ :  $f^{(k)*} = f^{(k)*}(x^{(k-1)})$ . Функцию  $f^{(k)*}(x^{(k-1)})$  называют *функцией Беллмана* последних  $N-k+1$  шагов процесса и обозначают через  $B_k(x^{(k-1)})$ .

Для  $k = N$  имеем

$$B_N(x^{(N-1)}) = \min_{u^{(N)} \in \bar{U}_N(x^{(N-1)})} f_N(x^{(N-1)}, u^{(N)}). \quad (3.41)$$

Найдем рекуррентное соотношение, связывающее между собой функции Беллмана  $B_k(x^{(k-1)})$  и  $B_{k+1}(x^{(k)})$ ,  $k = \overline{1, N-1}$ . Для этого оптимизируем  $k$ -ый шаг для начального состояния  $x^{(k-1)}$  согласно принципу оптимальности. Пусть на последних  $N-k$  шагах при всех  $x^{(k)} \in X_k$  подобная оптимизация уже проведена, и функция Беллмана  $B_{k+1}(x^{(k)})$  этих шагов известна. Если на  $k$ -м шаге с начальным состоянием  $x^{(k-1)}$  выбрать управление  $u^{(k)}$ , то процесс перейдет в состояние  $x^{(k)}$  (начальное для последних  $N-k$  шагов) в соответствии

с уравнением (3.37):  $x^{(k)} = F^{(k)}(x^{(k-1)}, u^{(k)})$ , а минимум  $f^{(k+1)*}(x^{(k)})$  целевой функции последних  $N - k$  шагов станет равным  $B_{k+1} \left[ F^{(k)}(x^{(k-1)}, u^{(k)}) \right]$ .

Учитывая вклад и  $k$ -го шага в значение целевой функции заключительных  $N - k + 1$  шагов, соответствующее оптимальному завершению процесса из состояния  $x^{(k-1)}$ , получаем:

$$f^{(k)*}(x^{(k-1)}) = B_k(x^{(k-1)}) =$$

$$= \min_{u_k \in \bar{U}_k(x^{(k-1)})} \left\{ B_{k+1} \left[ F^{(k)}(x^{(k-1)}, u^{(k)}) \right] + f_k(x^{(k-1)}, u^{(k)}) \right\}, \quad (3.42)$$

$$k = \overline{1, N-1}.$$

Иллюстрация к выводу этой формулы приведена на рисунке:

$$\begin{array}{c} \underbrace{f_k(x^{(k-1)}, u^{(k)})}_{\text{}} \quad \underbrace{B_{k+1} \left[ F^{(k)}(x^{(k-1)}, u^{(k)}) \right]}_{\text{}} \\ \underbrace{\dots x^{(k-1)} \xrightarrow{u^{(k)}} x^{(k)}}_{\text{}} = F^{(k)}(x^{(k-1)}, u^{(k)}) \xrightarrow{\text{}} x^{(k+1)} \xrightarrow{\text{}} \dots \xrightarrow{\text{}} x^{(N)} \end{array}$$

Соотношения (3.41) и (3.42), позволяющие последовательно найти функции  $B_N(x^{(N-1)})$ ,  $B_{N-1}(x^{(N-2)})$ , ...,  $B_1(x^{(0)})$ , называются *уравнениями Беллмана*.

Находя функции Беллмана  $B_k(x^{(k-1)})$ ,  $k = N, N-1, \dots, 1$  из (3.41) и (3.42), мы одновременно определим и управления  $u^{(k)*}(x^{(k-1)})$ , которым отвечают минимальные значения соответствующих величин  $Z_N = f_N(x^{(N-1)}, u^N)$  и

$$Z_k = B_{k+1} \left[ F^{(k)}(x^{(k-1)}, u^{(k)}) \right] + f_k(x^{(k-1)}, u^{(k)}), \quad k = N, N-1, \dots, 1$$

из правых частей равенств (3.41) и (3.42):

$$Z_N \left[ x^{(N-1)}, u^{(N)*}(x^{(N-1)}) \right] = \min_{u^{(N)} \in \bar{U}_N(x^{(N-1)})} Z_N \left[ x^{(N-1)}, u^{(N)} \right], \quad (3.43)$$

$$Z_k \left[ x^{(k-1)}, u^{(k)*}(x^{(k-1)}) \right] = \min_{u^{(k)} \in \bar{U}_k(x^{(k-1)})} Z_k \left[ x^{(k-1)}, u^{(k)} \right],$$

$$k = N, N-1, \dots, 1. \quad (3.44)$$

Управления  $u^{(k)*}(x^{(k-1)})$ ,  $k = \overline{1, N}$  называются *условными оптимальными управлениями*, а процесс их нахождения – *условной оптимизацией*. Отметим, что управления  $u^{(k)*}(x^{(k-1)})$ , найденные в соответствии с (3.44), удовлетворяют принципу оптимальности, т.е. с

учетом начального состояния  $x^{(k-1)}$   $k$ -го шага управление  $u^{(k)*}(x^{(k-1)})$  является оптимальным не только для этого шага, но и для следующих за ним  $N - k$  шагов.

Таким образом, в результате условной оптимизации находятся функции Беллмана  $B_k(x^{(k-1)})$  и условные оптимальные управления  $u^{(k)*}(x^{(k-1)})$ ,  $k = \overline{1, N}$ . После этого можно осуществить *безусловную оптимизацию*, т.е. определить искомым оптимальное управление процессом  $\hat{u}^* = \{u^{(1)*}, \dots, u^{(N)*}\}$  и оптимальную фазовую траекторию  $\hat{x}^* = \{x^{(0)*}, \dots, x^{(N)*}\}$  следующим образом.

Так как функция Беллмана  $B_1(x^{(0)})$  для каждого начального состояния  $x^{(0)} \in X_0$  равна минимальному значению целевой функции  $N$  шагов, т.е. всего процесса, начатого из этого состояния, то оптимальное начальное условие процесса  $x^{(0)*} \in X_0$  находится из соотношения:

$$B_1(x^{(0)*}) = \min_{x^{(0)} \in X_0} B_1(x^{(0)}) \quad (3.45)$$

(если множество  $X_0$  состоит из единственной точки  $x^{(0)}$ , то  $x^{(0)*} = x^{(0)}$ ).

Далее, используя найденные условные оптимальные управления, а также уравнения состояний (3.37), последовательно находим  $u^{(1)*}$ ,  $x^{(1)*}$ ,  $u^{(2)*}$ ,  $x^{(2)*}$ , ...,  $u^{(N)*}$ ,  $x^{(N)*}$  из следующих соотношений:

$$\begin{aligned} u^{(1)*} &= u^{(1)*}(x^{(0)*}), & x^{(1)*} &= F^{(1)}(x^{(0)*}, u^{(1)*}); \\ u^{(2)*} &= u^{(2)*}(x^{(1)*}), & x^{(2)*} &= F^{(2)}(x^{(1)*}, u^{(2)*}); \\ &\dots & & \dots \\ u^{(N)*} &= u^{(N)*}(x^{(N-1)*}), & x^{(N)*} &= F^{(N)}(x^{(N-1)*}, u^{(N)*}). \end{aligned} \quad (3.46)$$

На основании приведенных преобразований строится алгоритм метода динамического программирования.

Рассмотрим пример применения дискретного варианта метода динамического программирования.

**Пример 3.7.** Дискретная задача о распределении средств.

Записать математическую модель задачи о распределении средств в виде многошаговой задачи оптимизации.

Найти оптимальное с точки зрения суммарного дохода распределение средств предприятиям.

Средства в размере  $S$  усл. ед. распределяются между  $N$  предприятиями в количествах, кратных  $a$  усл. ед., где  $S/a = m$  – целое число. При выделении  $k$  – му предприятию средств  $u$  усл. ед. ( $u = ai$ ,  $i = \overline{1, m}$ ) оно приносит доход  $f_k(u)$  усл. ед.,  $k = \overline{1, n}$  (функции  $f_k(u)$  заданы таблично). В соответствии с постановкой математическая модель задачи записывается в следующем виде:

$$f(u) = \sum_{k=1}^N f_k(u_k) \rightarrow \max ;$$

$$\sum_{k=1}^N u_k = S, \quad u_k = \{0, a, 2a, \dots, ma\}.$$

Задачу решить при следующих исходных данных:  $S = 4$  усл. ед.,  $N = 3$ ,  $a = 1$  усл. ед.. Целевые функции заданы таблично (табл. 3.5.).

Таблица 3.5

Доход $f_i(u)$ усл. ед.	Выделенные средства $u$ , усл. ед.				
	0	1	2	3	4
$f_1(u)$	0	8	14	15	16
$f_2(u)$	0	6	8	9	10
$f_3(u)$	0	7	9	10	11

### Решение

Будем считать, что из общей суммы  $S$  усл. ед. средства предприятиям выделяются по очереди в  $N$  этапов: сначала первому предприятию  $u^{(1)}$  усл. ед., затем второму  $u^{(2)}$  усл. ед. и т.д. и, наконец,  $N$ –му  $u^{(N)}$  усл. ед. Обозначим сумму выделенных к началу  $k$ –го этапа средств через  $x^{(k-1)}$ . Очевидно,  $0 \leq x^{(k)} \leq S$  и  $x^{(k)} = x^{(k-1)} + u^{(k)}$ ,  $k = \overline{1, N}$ . Отсюда следует, что  $0 \leq u^{(k)} \leq S - x^{(k-1)}$ . Кроме того,  $u_k = \{0, a, 2a, \dots, ma\}$ .

С учетом всего сказанного получаем трехшаговую задачу оптимизации:

$$f(\hat{x}, \hat{u}) = \sum_{k=1}^3 f_k(u^{(k)}) \rightarrow \max ;$$

$$x^{(k)} = x^{(k-1)} + u^{(k)}, \quad k = 1, 2, 3 ;$$

$$x^{(0)} = 0, \quad x^{(3)} = 4, \quad x^{(k)} \in [0; 4], \quad k = 1, 2;$$

$$u^{(k)} \in [0; 4 - x^{(k-1)}] \cap Z,$$

где частичные целевые функции  $f_k(u)$  заданы таблично (табл. 3.5).

Рассмотрим реализацию метода динамического программирования.

**Этап 1** (условная оптимизация)

1. Найдем  $B_3(x^{(2)}) = \max_{u^{(3)} \in [0; 4-x^{(2)}] \cap Z} f_3(u^{(3)})$ . Так как  $Z_3(x^{(2)}, u^{(3)}) = f_3(u^{(3)})$  – возрастающая функция аргумента  $u^{(3)}$  (табл. 3.5), то ее максимум достигается при максимальном допустимом значении  $u^{(3)}$ , т.е.

$$u^{(3)*}(x^{(2)}) = [4 - x^{(2)}]. \quad (3.47)$$

Отсюда  $B_3(x^{(2)}) = Z_3(x^{(2)}, u^{(3)*}(x^{(2)})) = f_3([4 - x^{(2)}])$ . Значения  $B_3(x^{(2)})$ , найденные с помощью табл. 3.5, представлены в табл. 3.6.

Таблица 3.6

$x^{(2)}$	0	1	2	3	4
$B_3(x^{(2)})$	11	10	9	7	0

2. Найдем  $B_2(x^{(1)}) = \max_{u^{(2)} \in [0; 4-x^{(1)}] \cap Z} \{B_3(x^{(1)} + u^{(2)}) + f_2(u^{(2)})\}$ .

Для определения максимума функции  $Z_2(x^{(1)}, u^{(2)}) = B_3(x^{(1)} + u^{(2)}) + f_2(u^{(2)})$  составим таблицу 3.7 значений этой функции, используя табл. 3.5 и 3.6.

Таблица 3.7

$x^{(1)} \backslash u^{(2)}$	0	1	2	3	4
0	11	16	17	16	10
1	10	15	15	9	–
2	9	13	8	–	–
3	7	6	–	–	–
4	0	–	–	–	–

В табл. 3.7 выделены максимальные по  $u^{(2)}$  значения функции  $Z_2(x^{(1)}, u^{(2)})$ , соответствующие различным значениям  $x^{(1)}$ .

С помощью табл. 3.7 находим функции  $B_2(x^{(1)})$  и  $u^{(2)*}(x^{(1)})$ , представив их значения в табл. 3.8 и 3.9.

Таблица 3.8

$x^{(1)}$	0	1	2	3	4
$B_2(x^{(1)})$	17	15	13	7	0

Таблица 3.9

$x^{(1)}$	0	1	2	3	4
$u^{(2)*}(x^{(1)})$	2	1; 2	1	0	0

3. Так как множество  $X_0$  состоит из единственной точки  $x_0 = 0$ , то найдем только  $u^{(1)*}(0)$  и  $B_1(0)$ :

$$B_1(0) = \max_{u^{(1)} \in \{0; 4\} \cap \mathbb{Z}} \{B_2(0 + u^{(1)}) + f_1(u^{(1)})\}.$$

Для определения максимума в правой части последнего равенства составим табл. 3.10 значений функции  $Z_2(0, u^{(1)}) = B_2(u^{(1)}) + f_1(u^{(1)})$ , которые найдем с помощью табл. 3.5 и 3.8.

Таблица 3.10

$u^{(1)}$	0	1	2	3	4
$Z_2(0, u^{(1)})$	17	23	27	22	16

Из табл. 3.10 видно, что

$$u^{(1)*}(0) = 2, \quad (3.48)$$

$$B_1(0) = 27. \quad (3.49)$$

**Этап 2** (безусловная оптимизация).

0. Оптимальное начало фазовой траектории определяется начальным условием  $x^{(0)*} = 0$ .

1. С помощью (3.48) находим:  $u^{(1)*} = u^{(1)*}(0) = 2$ ,  $x^{(1)*} = x^{(0)*} + u^{(1)*} = 2$ .

2. Из таблицы 3.9 получаем  $u^{(2)*} = u^{(2)*}(2) = 1$ ,  $x^{(2)*} = x^{(1)*} + u^{(2)*} = 3$ .

3. Используя (3.47), находим:  $u^{(3)*} = u^{(3)*}(3) = 1$ ,  $x^{(3)*} = x^{(2)*} + u^{(3)*} = 4$ .

Окончательно получаем:  $\hat{u}^* = \{2, 1, 1\}$ ,  $\hat{x}^* = \{0, 2, 3, 4\}$ . Таким образом, первому предприятию следует выделить 2 усл. ед., а второму и третьему – по 1 усл. ед.. При этом суммарный доход предприятий достигнет максимального значения и составит  $B_1(0) = 27$  усл. ед..

Метод динамического программирования, как было сказано, является следствием так называемого *принципа оптимальности*, сформулированного Р. Беллманом. На основе этого принципа определяются оптимальные решения для достаточно широкого класса задач, как дискретных, так и непрерывных.

С одной стороны, с помощью метода динамического программирования решают задачи, связанные с непрерывными процессами оптимального управления, и тогда он представляет собой один из приемов решения *вариационных* задач. В этом методе органически присутствует способ численного решения непрерывных вариационных задач: необходимо лишь заменить непрерывный процесс дискретным с соответствующим малым интервалом дискретности. С другой стороны, метод динамического программирования в применении к решению дискретных задач позволяет достичь существенной экономии вычислений по сравнению с полным перебором вариантов.

Поскольку общей канонической формы представления многошаговых задач оптимизации не существует, невозможно составить общую эффективную программу решения всех таких задач методом динамического программирования. Тем не менее, составление программ решения конкретных задач, как правило, не вызывает затруднений.

Большую ценность метод динамического программирования приобретает при принятии оперативных решений в реальном времени, поскольку после того как проведена условная оптимизация, появляется возможность выбора оптимальной стратегии на любом этапе процесса, независимо от предыдущих (возможно и неоптимальных) решений, принятых на начальных этапах.

Методы отсечений, ветвей и границ и динамического программирования, рассмотренные в этой главе, являются основными методами дискретной оптимизации.

## Глава 4

### МЕТОДЫ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ

В этой главе рассматривается простейшая математическая модель оптимизации, в которой целевая функция зависит от одной переменной, а допустимым множеством является отрезок вещественной оси.

К таким моделям приводят прикладные задачи оптимизации с одной управляемой переменной. Кроме того, и это главное, необходимость в методах одномерной оптимизации возникает при реализации многих итерационных методов решения многомерных и других более сложных задач оптимизации.

Рассмотрены некоторые известные методы, хорошо проявившие себя на практике.

#### 4.1. Предварительные сведения

Пусть математическая модель задачи оптимизации имеет вид:

$$f(x) \rightarrow \min, \quad x \in U \subset R^1, \quad (4.1)$$

где  $U = [a; b]$  – произвольный интервал (отрезок), он может быть открытым или полуоткрытым.

Как уже отмечалось в главе 1, максимизация целевой функции ( $f(x) \rightarrow \max$ ) эквивалентна минимизации противоположной величины ( $-f(x) \rightarrow \min$ ), поэтому, не умаляя общности, будем рассматривать только задачи минимизации.

Широкий класс функций  $f(x)$ , для которых множество точек минимума этой функции  $U^* \subset U$  не пусто, определяет хорошо известная из математического анализа *теорема Вейерштрасса*, согласно которой непрерывная на замкнутом ограниченном множестве функция достигает на этом множестве своих минимального и максимального значений. Таким образом, задача (4.1) с непрерывной целевой функцией  $f(x)$  всегда имеет решение.

Если функция  $f(x)$  на множестве  $U$  имеет, кроме глобального, локальные минимумы, отличные от него, то минимизация  $f(x)$ , как правило, сильно затрудняется. В частности, многие методы поиска

точки минимума  $f(x)$  приспособлены только для функций, у которых каждый локальный минимум является одновременно и глобальным. Этим свойством обладают *унимодальные функции*.

**Определение 4.1.** Функция  $f(x)$  называется *унимодальной* на отрезке  $[a; b]$ , если она непрерывна на этом отрезке и существуют числа  $\alpha$  и  $\beta$ ,  $a \leq \alpha \leq \beta \leq b$  такие, что:

- 1) если  $a < \alpha$ , то на отрезке  $[a; \alpha]$  функция  $f(x)$  монотонно убывает;
- 2) если  $\beta < b$ , то на отрезке  $[\beta; b]$  функция  $f(x)$  монотонно возрастает;
- 3) при  $x \in [\alpha; \beta]$   $f(x) = f_{\min} = \min_{[a; b]} f(x)$ .

Множество унимодальных на отрезке  $[a; b]$  функций обозначают через  $Q[a; b]$ .

Отметим, что возможно вырождение в точку одного или двух отрезков из  $[a; \alpha]$ ,  $[\alpha; \beta]$  и  $[\beta; b]$ . Некоторые варианты расположения и вырождения в точку отрезков монотонности и постоянства унимодальной функции показаны на рис. 4.1.

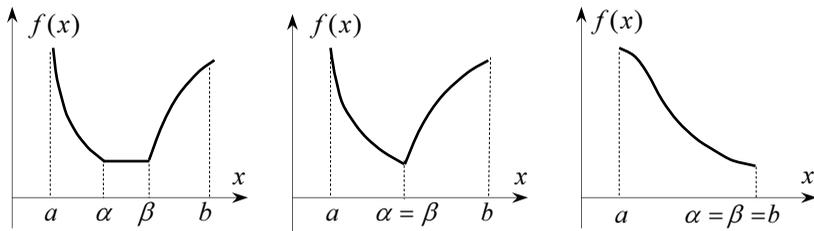


Рис. 4.1.

Из определения 4.1 вытекают следующие *основные свойства унимодальных функций*:

1. Любая из точек локального минимума унимодальной функции является и точкой ее глобального минимума на отрезке  $[a; b]$ .
2. Функция, унимодальная на отрезке  $[a; b]$  является унимодальной и на любом меньшем отрезке  $[c; d] \subset [a; b]$ .
3. Пусть  $f(x) \in Q[a; b]$  и  $a \leq x_1 < x_2 \leq b$ . Тогда
 
$$\begin{aligned} \text{если } f(x_1) \leq f(x_2), \text{ то } x^* \in [a; x_2]; \\ \text{если } f(x_1) \geq f(x_2), \text{ то } x^* \in [x_1; b]; \end{aligned} \quad (4.2)$$

где  $x^*$  – одна из точек минимума  $f(x)$  на отрезке  $[a; b]$ .

Частным случаем унимодальных функций являются *выпуклые функции*, которые удовлетворяют более жестким требованиям.

**Определение 4.2.** Функция  $f(x)$ , заданная на отрезке  $[a; b]$ , называется *выпуклой* на этом отрезке, если для всех  $x_1, x_2 \in [a; b]$  и произвольного числа  $\alpha \in [0; 1]$  выполняется неравенство

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2). \quad (4.3)$$

*Выпуклые функции обладают следующими основными свойствами:*

1. Если функция  $f(x)$  выпукла на  $[a; b]$ , то на любом отрезке  $[x_1; x_2] \subset [a; b]$  ее график расположен не выше хорды, проведенной через точки графика с абсциссами  $x_1$  и  $x_2$ .

2. Из курса математического анализа известны следующие условия выпуклости функции:

а) для того чтобы дифференцируемая на отрезке  $[a; b]$  функция  $f(x)$  была выпуклой на этом отрезке, необходимо и достаточно, чтобы производная  $f'(x)$  не убывала на  $[a; b]$ ;

б) для того чтобы дважды дифференцируемая на отрезке  $[a; b]$  функция  $f(x)$  была выпуклой на этом отрезке, необходимо и достаточно, чтобы при всех  $x \in [a; b]$  выполнялось неравенство  $f''(x) \geq 0$ .

3. Условие выпуклости для дифференцируемой на отрезке  $[a; b]$  функции  $f(x)$  означает, что на этом отрезке любая касательная к графику  $f(x)$  лежит не выше этого графика.

4. Если  $f(x)$  – выпуклая дифференцируемая на отрезке  $[a; b]$  функция и в точке  $x^* \in [a; b]$  выполняется равенство

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

то  $x^*$  является точкой *глобального минимума*  $f(x)$  на  $[a; b]$ .

5. Можно показать, что всякая выпуклая непрерывная на отрезке  $[a; b]$  функция является и унимодальной на этом отрезке. Обратное, вообще говоря, неверно.

Таким образом, кроме перечисленных свойств, выпуклые функции обладают также и всеми свойствами унимодальных функций.

При исследовании выпуклости функций на практике неравенство (4.3) удается использовать только в редких случаях. Поэтому для дифференцируемых достаточное число раз функций обычно применяют дифференциальные критерии выпуклости (см. свойство 2 выпуклых функций).

Непосредственная проверка унимодальности с помощью определения 4.1 также в большинстве случаев вызывает затруднения, и для обоснования унимодальности достаточно гладких функций часто используют те же критерии выпуклости. Разумеется, при отрицательном результате проверки функции на выпуклость нельзя сделать вывод о том, что она не унимодальная.

Применение некоторых методов одномерной минимизации возможно только в случае, если скорость изменения функции  $f(x)$  на любом участке отрезка  $[a; b]$  ограничена некоторым числом, одним и тем же для всех участков. В этом случае говорят, что  $f(x)$  удовлетворяет на  $[a; b]$  условию Липшица. Целевые функции большинства практических задач оптимизации указанным свойством обладают.

**Определение 4.3.** Функция  $f(x)$  удовлетворяет на отрезке  $[a; b]$  условию Липшица, если существует такое число  $L > 0$  (константа Липшица), что

$$|f(x_1) - f(x_2)| \leq L |x_1 - x_2| \quad (4.4)$$

для всех  $x_1, x_2 \in [a; b]$ .

З а м е ч а н и я

1. Если неравенство (4.4) выполняется с константой  $L$ , то оно справедливо и при всех  $L' > L$ . Поэтому для функции, удовлетворяющей условию Липшица, существует бесконечное множество констант  $L$  из (4.4).

При использовании алгоритмов минимизации с параметром  $L$ , наилучшие результаты достигаются, как правило тогда, когда в качестве  $L$  берется минимальная из констант Липшица.

2. Из условия (4.4) непосредственно следует непрерывность  $f(x)$  на отрезке  $[a; b]$ . Поэтому, согласно теореме Вейерштрасса, функция  $f(x)$ , удовлетворяющая на отрезке  $[a; b]$  условию Липшица, имеет на нем хотя бы одну точку минимума.

3. Условие (4.4) означает, что модуль углового коэффициента любой хорды графика  $f(x)$  не превосходит  $L$ .

Переходя в (4.4) к пределу при  $|x_1 - x_2| \rightarrow 0$ , убеждаемся, что если в некоторой точке существует касательная к графику функции  $f(x)$ , то модуль ее углового коэффициента также не может превышать  $L$ . Так, функция  $f(x) = \sqrt{x}$  на отрезке  $[0; 1]$  условию Липшица не удовлетворяет, потому что при  $x \rightarrow +0$  угловой коэффициент касательной к ее графику  $k$  неограниченно возрастает.

4. Если функция  $f(x)$  имеет на отрезке  $[a; b]$  непрерывную производную, то она удовлетворяет на этом отрезке условию Липшица с константой  $L = \max_{[a; b]} |f'(x)|$ .

5. Если  $a = x_0 < x_1 < \dots < x_n = b$ , а функция  $f(x)$  непрерывна на  $[a; b]$  и удовлетворяет условию (4.4) на каждом из отрезков  $[x_i; x_{i+1}]$ ,  $i = 0, 1, \dots, n-1$ , с константой  $L_i$ , то она удовлетворяет условию Липшица и на всем отрезке  $[a; b]$  с константой  $L = \max_{0 \leq i \leq n-1} L_i$ .

И наконец, из *математического анализа* известны следующие условия локального экстремума функции  $f(x)$ , дифференцируемой достаточное число раз.

1. Если функция  $f(x)$  дифференцируема в точке  $x^*$  и достигает в этой точке локального экстремума, то  $f'(x^*) = 0$  (*необходимое условие экстремума*).

Пусть функция  $f(x)$   $n$  раз дифференцируема в точке  $x^*$  и в этой точке все производные  $f(x)$  до  $(n-1)$ -го порядка включительно равны нулю, а  $f^{(n)}(x^*) \neq 0$ . Тогда, если  $n$  – нечетно, то  $x^*$  не является точкой локального экстремума функции  $f(x)$ . Если же  $n$  – четное число, то (*достаточное условие экстремума*):

а) при  $f^{(n)}(x^*) > 0$   $x^*$  – точка локального минимума  $f(x)$ ;

б) при  $f^{(n)}(x^*) < 0$   $x^*$  – точка локального максимума  $f(x)$ .

Перечисленные условия позволяют предложить следующий путь решения задачи минимизации (4.1):

1) с помощью условия 1 находим все точки возможного экстремума функции  $f(x)$  на интервале  $(a; b)$ , т.е. корни уравнения

$$f'(x) = 0$$

(стационарные точки), принадлежащие интервалу  $(a; b)$ ;

2) найденные стационарные точки исследуем в соответствии с условием 2, выделяя из них только точки локальных минимумов  $f(x)$ ;

3) значения  $f(x)$  в точках локальных минимумов и на концах отрезка  $[a; b]$  сравниваем между собой. Наименьшему из этих значений соответствует точка глобального минимума  $f(x)$  на  $[a; b]$ .

**З а м е ч а н и е .** Применение условия 2 требует вычисления высших производных функции  $f(x)$ , поэтому в большинстве случаев бывает проще сравнить значения  $f(x)$  во всех стационарных точках, не интересуясь их характером.

## 4.2. Прямые методы минимизации

Для решения задачи (4.1) минимизации функции  $f(x)$  на отрезке  $[a; b]$  на практике, как правило, применяют приближенные методы. Они позволяют найти решение этой задачи с необходимой точностью в результате определения конечного числа значений функции  $f(x)$  и ее производных в некоторых точках отрезка  $[a; b]$ . Методы, использующие только значения функции и не требующие вычисления ее производных, называются *прямыми методами* минимизации.

Достоинством прямых методов является то, что от целевой функции не требуется дифференцируемости и, более того, она может быть не задана в аналитическом виде. Единственное, на чем основаны алгоритмы прямых методов минимизации, это возможность определения значений  $f(x)$  в заданных точках.

Рассмотрим наиболее распространенные на практике прямые методы поиска точки минимума. Самым слабым требованием к функции  $f(x)$ , позволяющим использовать эти методы, является ее унимодальность. Поэтому далее будем считать функцию  $f(x)$  унимодальной на отрезке  $[a; b]$ .

### Метод перебора.

Метод перебора (или равномерного поиска) является простейшим из прямых методов минимизации и состоит в следующем.

Разобьем отрезок  $[a; b]$  на  $n$  равных частей точками деления  $x_i = a + i(b - a)/n$ ,  $i = 0, 1, \dots, n$ . Вычислив значения  $f(x)$  в точках  $x_i$ , путем сравнения найдем точку  $x_m$ ,  $0 \leq m \leq n$ , для которой

$$f(x_m) = \min_{0 \leq i \leq n} f(x_i). \quad (4.5)$$

Далее, положим  $x^* = x_m$ ,  $f_{\min} = f(x_m)$ .

Погрешность определения точки минимумах  $x^*$  функции  $f(x)$  методом перебора не превосходит величины  $\varepsilon_n = (b - a)/n$ .

Действительно, предположим, что  $x_m$  из (4.5) является внутренней точкой разбиения отрезка  $[a; b]$ , т.е.  $1 \leq m \leq n - 1$ . Тогда из соотношения (4.5) с учетом свойства (4.2) унимодальных функций следует что:

а)  $f(x_{m-1}) \geq f(x_m)$ , т.е.  $x^* \in [x_{m-1}; b]$ ;

б)  $f(x_m) \leq f(x_{m+1})$ , т.е.  $x^* \in [a; x_{m+1}]$ .

Отсюда получаем, что  $x^* \in [x_{m-1}; b] \cap [a; x_{m+1}] = [x_{m-1}; x_{m+1}]$ . Длина последнего отрезка равна  $2(b - a)/n$ , а точка  $x_m$  является его серединой. Поэтому  $|x_m - x^*| \leq (b - a)/n = \varepsilon_n$ .

Таким образом, чтобы обеспечить требуемую точность  $\varepsilon$  определения точки  $x^*$ , число отрезков разбиения  $n$  необходимо выбрать из условия  $\varepsilon_n = (b - a)/n \leq \varepsilon$ , т.е.  $n \geq (b - a)/\varepsilon$ .

Пусть реализация метода перебора потребовала  $N$  вычислений функции  $f(x)$ . Это означает, что отрезок  $[a; b]$  был разбит на  $n = N - 1$  частей и достигнутая точность определения  $x^*$  составила  $\varepsilon_n = \varepsilon_{N-1} = (b - a)/(N - 1)$ . Поэтому точность решения  $\varepsilon(N)$ , которую обеспечивает метод перебора в результате  $N$  вычислений  $f(x)$ , будет

$$\varepsilon(N) = \frac{b - a}{N - 1}.$$

### Метод поразрядного поиска.

Рассмотрим возможности усовершенствования метода перебора с целью уменьшения количества значений  $f(x)$ , которые необходимо находить в процессе минимизации.

Во-первых, если оказывается, что  $f(x_i) \leq f(x_{i+1})$ , то отпадает необходимость вычислять  $f(x)$  в точках  $x_{i+2}, x_{i+3}$  и т.д., так как  $x^* \leq x_{i+1}$  (см. (4.2)).

Во-вторых, разумно было бы сначала определить отрезок, содержащий  $x^*$ , грубо, т.е. найти точку  $x^*$  с небольшой точностью, а затем искать ее на этом отрезке с меньшим шагом дискретизации, повышая точность.

Указанные возможности улучшения метода перебора реализованы в методе *порядкового поиска*. В этом методе перебор точек отрезка происходит сначала с шагом  $\Delta = x_{i+1} - x_i > \varepsilon$  до тех пор, пока выполнится условие  $f(x_i) \leq f(x_{i+1})$  или пока очередная из этих точек не совпадет с концом отрезка. После этого шаг уменьшается (обычно в 4 раза), и перебор точек с новым шагом производится в противоположном направлении до тех пор, пока значения  $f(x)$  снова не перестанут уменьшаться или очередная точка не совпадет с другим концом отрезка и т.д. Описанный процесс завершается, когда перебор в данном направлении закончен, а использованный при этом шаг дискретизации не превосходит  $\varepsilon$ .

**Алгоритм метода порядкового поиска:**

1. Выбрать начальный шаг  $\Delta = (b - a)/4$ . Положить  $x_0 = a$ . Вычислить  $f(x_0)$ .
2. Положить  $x_1 = x_0 + \Delta$ . Вычислить  $f(x_1)$ .
3. Сравнить  $f(x_0)$  и  $f(x_1)$ . Если  $f(x_0) > f(x_1)$ , то перейти к шагу 4, иначе – к шагу 5
4. Положить  $x_0 = x_1$  и  $f(x_0) = f(x_1)$ . Проверить условие  $x_0 \in (a; b)$ . Если  $a < x_0 < b$ , то перейти к шагу 2, иначе – к шагу 5.
5. Проверить окончание поиска: если  $|\Delta| \leq \varepsilon$ , то вычисления завершить, полагая  $x^* = x_0$ ,  $f^* = f(x_0)$ , иначе перейти к шагу 6.
6. Изменить направление и шага поиска: положить  $x_0 = x_1$ ,  $f(x_0) = f(x_1)$ ,  $\Delta = \Delta/4$ . Перейти к шагу 2.

**Методы исключения отрезков.**

В методе перебора, рассмотренном выше, точки  $x_i$ , в которых определяются значения  $f(x)$ , выбираются заранее. Если же для выбора очередной точки вычисления (измерения)  $f(x)$  использовать информацию, содержащуюся в уже найденных значениях  $f(x)$ , то поиск точки минимума можно сделать более эффективным, т.е. сократить число определяемых для этого значений  $f(x)$ , как, например, в методе порядкового поиска.

Один из путей такого более эффективного поиска точки  $x^*$  указывает свойство 3 унимодальных функций (см. (4.2)).

Пусть  $a < x_1 < x_2 < b$ . Сравнив значения  $f(x)$  в точках  $x_1$  и  $x_2$  (пробных точках), можно сократить отрезок поиска точки  $x^*$ , перейдя к отрезку  $[a; x_2]$ , если  $f(x_1) \leq f(x_2)$ , или к отрезку  $[x_1; b]$ , если  $f(x_1) > f(x_2)$  (рис. 4.2).

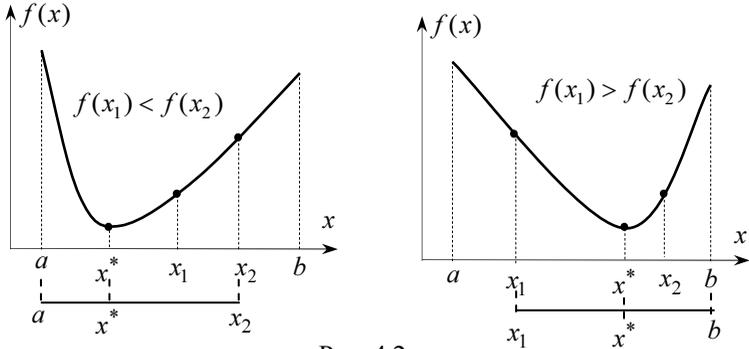


Рис. 4.2.

Описанную процедуру можно повторить необходимое число раз, последовательно уменьшая отрезок, содержащий точку минимума. Когда длина последнего из найденных отрезков станет достаточно малой, следует положить  $x^* = \bar{x}$  где  $\bar{x}$  – одна из точек этого отрезка, например, его середина. Методы минимизации, основанные на этом принципе, называются *методами исключения отрезков*.

Чтобы относительное уменьшение отрезка на каждой итерации не зависело от того, какая из его частей исключается из дальнейшего рассмотрения, пробные точки следует располагать симметрично относительно середины исходного отрезка. В зависимости от способа выбора пробных точек получаются различные методы исключения отрезков. На практике чаще всего используются *метод деления отрезка пополам (дихотомии)*, *метод золотого сечения*, *метод парабол*.

### Метод дихотомии

В этом методе точки  $x_1$  и  $x_2$  располагаются близко к середине очередного отрезка  $[a; b]$ , т.е.

$$x_1 = \frac{b+a-\delta}{2}, \quad x_2 = \frac{b+a+\delta}{2}, \quad (4.6)$$

где  $\delta > 0$  – малое число. При этом отношение длин нового и исходного отрезков  $\tau = \frac{b-x_1}{b-a} = \frac{x_2-a}{b-a}$  близко к  $\frac{1}{2}$ , этим и объясняется название метода.

Отметим, что для любых точек  $x_1$  и  $x_2$  величина  $\tau > 1/2$ , поэтому указанный выбор пробных точек объясняется стремлением обеспечить максимально возможное относительное уменьшение отрезка на каждой итерации поиска  $x^*$ .

В конце вычислений по методу дихотомии в качестве приближенного значения  $x^*$  берут середину последнего из найденных отрезков  $[a; b]$ , убедившись предварительно, что достигнуто неравенство  $(b - a)/2 \leq \varepsilon$ .

#### *Алгоритм метода дихотомии*

1. Определить  $x_1$  и  $x_2$  по формулам (4.6). Вычислить  $f(x_1)$  и  $f(x_2)$ .
2. Сравнить  $f(x_1)$  и  $f(x_2)$ . Если  $f(x_1) \leq f(x_2)$ , то перейти к отрезку  $[a; x_2]$ , положив  $b = x_2$ , иначе – к отрезку  $[x_1; b]$ , положив  $a = x_1$ .
3. Найти достигнутую точность  $\varepsilon_n = (b - a)/2$ . Если  $\varepsilon_n > \varepsilon$ , то перейти к следующей итерации, вернувшись к шагу 1. Если  $\varepsilon_n \leq \varepsilon$ , то завершить поиск  $x^*$ , перейдя к шагу 4.
4. Положить  $x^* = \bar{x} = (a + b)/2$ ,  $f^* = f(\bar{x})$ .

#### *Замечание 1*

Число  $\delta$  из (4.6) выбирается на интервале  $(0; 2\varepsilon)$  с учетом следующих соображений:

а) чем меньше  $\delta$ , тем больше относительное уменьшение длины отрезка на каждой итерации, т.е. при уменьшении  $\delta$  достигается более высокая скорость сходимости метода дихотомии;

б) при чрезмерно малом  $\delta$  сравнение значений  $f(x)$  в точках  $x_1$  и  $x_2$ , отличающихся на величину  $\delta$ , становится затруднительным. Поэтому выбор  $\delta$  должен быть согласован с точностью определения  $f(x)$  и с количеством верных десятичных знаков при задании аргумента  $x$ .

#### *Замечание 2*

Число  $n$  итераций метода дихотомии, необходимое для определения точки  $x^*$  с точностью  $\varepsilon$ , определяется неравенством

$$n \geq \log_2 \frac{b - a - \delta}{2\varepsilon - \delta}. \quad (4.7)$$

Действительно, обозначив длину исходного отрезка  $[a; b]$  через  $\Delta_0$ , рассмотрим изменение отрезка по итерациям. Длина отрезка, полученного после первой итерации, будет

$$\Delta_1 = (\Delta_0 + \delta)/2,$$

после второй итерации

$$\Delta_2 = (\Delta_1 + \delta)/2 = (b - a)/4 + \delta(1/4 + 1/2),$$

после третьей итерации

$$\Delta_3 = (\Delta_2 + \delta)/2 = (b - a)/8 + \delta(1/8 + 1/4 + 1/2) \text{ и т.д.}$$

Таким образом, в результате  $n$  итераций длина отрезка поиска точки  $x^*$  станет

$$\Delta_n = (b - a)/2^n + \delta(1/2^n + 1/2^{n-1} + \dots + 1/2) = (b - a)/2^n + \delta(1 - 1/2^n).$$

При этом будет достигнута точность определения точки минимума  $\varepsilon_n = \Delta_n/2$ . Находя  $n$  из условия

$$\varepsilon_n = \frac{b - a}{2^{n+1}} + \left(1 - \frac{1}{2^n}\right) \frac{\delta}{2} \leq \varepsilon,$$

получаем неравенство (4.7).

**Замечание 3**

Величина  $\delta$  может быть выбрана достаточно малой, поэтому, пренебрегая ею в (4.7), получаем:  $\varepsilon_n = (b - a)/2^{n+1}$ . На каждой итерации метода дихотомии вычисляются два значения  $f(x)$ . Поэтому после  $N$  вычислений  $f(x)$  производится  $n = N/2$  итераций и достигается точность определения  $x^*$ :

$$\varepsilon(N) = \varepsilon_{N/2} = \frac{b - a}{2^{N/2+1}}.$$

### Метод золотого сечения

Рассмотрим такое симметричное расположение точек  $x_1$  и  $x_2$  на отрезке  $[a; b]$ , при котором одна из них становится пробной точкой и на новом отрезке, полученном после исключения части исходного отрезка. Использование таких точек позволяет на каждой итерации метода исключения отрезков, кроме первой, ограничиться определением только одного значения  $f(x)$ , так как другое значение уже найдено на одной из предыдущих итераций.

Найдем точки  $x_1$  и  $x_2$ , обладающие указанным свойством.

Рассмотрим сначала отрезок  $[0; 1]$  и для определенности предположим, что при его уменьшении исключается правая часть этого отрезка. Пусть  $x_2 = \tau$ , тогда симметрично расположенная точка  $x_1 = 1 - \tau$  (рис. 4.3).

Пробная точка  $x_1$  отрезка  $[0; 1]$  перейдет в пробную точку  $x_2' = 1 - \tau$  нового отрезка  $[0; \tau]$ . Чтобы точки  $x_2 = \tau$  и  $x_2' = 1 - \tau$  делили отрезки  $[0; 1]$  и  $[0; \tau]$  в одном и том же отношении, должно выполняться равенство  $1/\tau = \tau/(1 - \tau)$  или  $\tau^2 + \tau - 1 = 0$ , откуда находим положительное значение  $\tau = (\sqrt{5} - 1)/2 = 0.61803\dots$ . Таким образом,  $x_1 = 1 - \tau = (3 - \sqrt{5})/2$ ,  $x_2 = \tau = (\sqrt{5} - 1)/2$ .

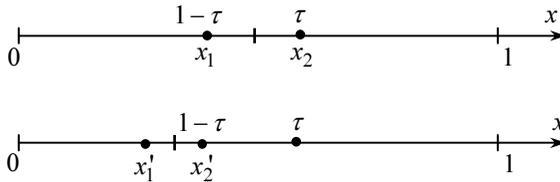


Рис. 4.3.

Для произвольного отрезка  $[a; b]$  выражения для пробных точек примут вид

$$x_1 = a + \frac{3 - \sqrt{5}}{2}(b - a); \quad x_2 = a + \frac{\sqrt{5} - 1}{2}(b - a) \quad (4.8)$$

Точки  $x_1$  и  $x_2$ , из (4.8) обладают следующим свойством: каждая из них делит отрезок  $[a; b]$  на две неравные части так, что отношение длины всего отрезка к длине его большей части равно отношению длин большей и меньше частей отрезка. Точки с таким свойством называются *точками золотого сечения* отрезка  $[a; b]$ . Это и объясняет название рассматриваемого метода.

На каждой итерации исключения отрезков с пробными точками (4.8) одна из них переходит на следующий отрезок и значение  $f(x)$  в этой точке вычислять не следует. Если новым отрезком становится  $[a; x_2]$ , то на него переходит пробная точка  $x_1$  исходного отрезка, становясь его второй пробной точкой ( $x_2' = x_1$ ) (рис. 4.3). В случае

перехода к отрезку  $[x_1; b]$  пробная точка  $x_2$  исходного отрезка становится первой пробной точкой отрезка  $[x_1; b]$ .

Легко проверить, что  $x_1 = a + b - x_2$  и  $x_2 = a + b - x_1$ . Поэтому на каждой итерации метода золотого сечения недостающую пробную точку нового отрезка можно найти по перешедшей на него пробной точке с помощью сложения и вычитания, не используя формул (4.8).

В конце вычислений по методу золотого сечения в качестве приближенного значения  $x^*$  можно взять середину последнего из полученных отрезков  $\bar{x} = (a + b)/2$ .

На каждой итерации отрезок поиска точки минимума уменьшается в одном и том же отношении  $\tau = (\sqrt{5} - 1)/2$ , поэтому в результате  $n$  итераций его длина становится  $\Delta_n = \tau^n (b - a)$ . Таким образом, точность  $\varepsilon_n$  определения точки  $x^*$  после  $n$  итераций находится из равенства

$$\varepsilon_n = \frac{\Delta_n}{2} = \frac{1}{2} \left( \frac{\sqrt{5} - 1}{2} \right)^n (b - a),$$

а условием окончания поиска точки  $x^*$  с точностью  $\varepsilon$  служит неравенство  $\varepsilon_n \leq \varepsilon$ .

*Алгоритм метода золотого сечения:*

1. Найти  $x_1$  и  $x_2$  по формулам (4.8). Вычислить  $f(x_1)$  и  $f(x_2)$ . Положить  $\tau = (\sqrt{5} - 1)/2$ ,  $\varepsilon_n = (b - a)/2$ .

2. Проверить окончание поиска: если  $\varepsilon_n > \varepsilon$ , то перейти к шагу 3, иначе – к шагу 4.

3. Перейти к новому отрезку и новым пробным точкам. Если  $f(x_1) \leq f(x_2)$ , то положить  $b = x_2$ ,  $x_2 = x_1$ ,  $f(x_2) = f(x_1)$ ,  $x_1 = b - \tau(b - a)$  и вычислить  $f(x_1)$ , иначе – положить  $a = x_1$ ,  $x_1 = x_2$ ,  $f(x_1) = f(x_2)$ ,  $x_2 = a + \tau(b - a)$  и вычислить  $f(x_2)$ .

Положить  $\varepsilon_n = \tau \varepsilon_n$  и перейти к шагу 2.

4. Окончание поиска: положить  $x^* = \bar{x} = (a + b)/2$ ,  $f^* = f(\bar{x})$ .

### Метод парабол

Поиск точки минимума методами исключения отрезков основан на сравнении значений функции в двух точках. При таком сравнении

разности значений  $f(x)$  в этих точках не учитываются, важны только их знаки.

Учесть информацию, содержащуюся в относительных изменениях значений  $f(x)$  в пробных точках, позволяют *методы полиномиальной аппроксимации*, основная идея которых состоит в том, что для функции  $f(x)$  строится аппроксимирующий многочлен и его точка минимума служит приближением к  $x^*$ . Для эффективного использования этих методов на функцию  $f(x)$ , кроме унимодальности, налагается дополнительное требование достаточной гладкости (по крайней мере, непрерывности).

Обоснованием указанных методов является известная из математического анализа теорема Вейерштрасса об аппроксимации, согласно которой непрерывную на отрезке функцию можно с любой точностью приблизить на этом отрезке некоторым полиномом.

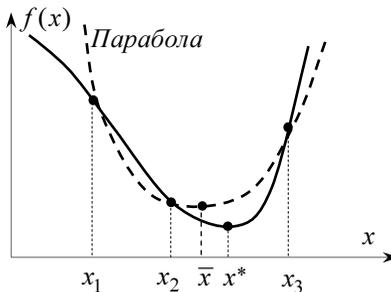


Рис. 4.4

Для повышения точности аппроксимации можно, во-первых, увеличивать порядок полинома и, во-вторых, уменьшать длину отрезка аппроксимации. Первый путь приводит к быстрому усложнению вычислительных процедур, поэтому на практике используются аппроксимирующие полиномы не выше третьего порядка. В то же время

уменьшение отрезка, содержащего точку минимума унимодальной функции, не представляет особого труда.

В простейшем методе полиномиальной аппроксимации – *методе парабол* используются полиномы второго порядка. На каждой итерации этого метода строится квадратный трехчлен, график которого (парабола) проходит через три выбранные точки графика функции  $f(x)$  (рис.4.4).

Опишем метод парабол. Рассмотрим унимодальную на отрезке  $[a; b]$  функцию  $f(x)$ , достигающую минимума во внутренней точке

этого отрезка. Выберем три точки  $x_1, x_2$  и  $x_3$  отрезка  $[a; b]$ , для которых выполняются неравенства

$$x_1 < x_2 < x_3, \quad f(x_1) \geq f(x_2) \leq f(x_3). \quad (4.9)$$

Из унимодальности функции  $f(x)$  следует, что  $x^* \in [x_1; x_3]$ .

Построим квадратный трехчлен

$$q(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2),$$

график которого проходит через точки  $(x_1, f(x_1))$ ,  $(x_2, f(x_2))$ ,  $(x_3, f(x_3))$  графика функции  $f(x)$ . Будем считать, что хотя бы одно из неравенств (4.9) для  $f(x_i)$  является строгим (если  $f(x_1) = f(x_2) = f(x_3)$ , то поиск точки  $x^*$  на этом закончен, так как из унимодальности функции  $f(x)$  следует, что она достигает минимума в каждой точке отрезка  $[x_1; x_3]$ ). Тогда из (4.9) следует, что ветви искомой параболы направлены вверх, а точка минимума  $\bar{x}$  трехчлена  $q(x)$  принадлежит отрезку  $[x_1; x_3]$ .

Определяя коэффициенты  $a_0, a_1$  и  $a_2$  из системы уравнений

$$\begin{cases} q(x_1) = f(x_1) = f_1, \\ q(x_2) = f(x_2) = f_2, \\ q(x_3) = f(x_3) = f_3, \end{cases}$$

находим:

$$a_0 = f_1, \quad a_1 = \frac{f_2 - f_1}{x_2 - x_1}, \quad a_2 = \frac{1}{x_3 - x_2} \left( \frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right).$$

Точку минимума  $\bar{x}$  квадратного трехчлена  $q(x)$  вычислим, приравняв его производную к нулю. Получим:

$$\begin{aligned} \bar{x} &= \frac{1}{2} \left( x_1 + x_2 - \frac{a_1}{a_2} \right) = \\ &= \frac{1}{2} \left[ x_1 + x_2 - \frac{(f_2 - f_1)(x_3 - x_2)}{(x_2 - x_1)} \middle/ \left( \frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right) \right]. \quad (4.10) \end{aligned}$$

Число  $\bar{x}$  из (4.10) служит очередным приближением метода парабол к  $x^*$ . Далее описанная процедура повторяется для новых точек  $x_1, x_2, x_3$ , удовлетворяющих неравенствам (4.9).

Выбрать эти точки среди  $x_1, x_2, x_3$  и  $\bar{x}$  можно с помощью перехода от исходного к новому отрезку  $[x_1; x_3]$ , содержащему точку  $x^*$ , методом исключения отрезков. Для этого перехода используются пробные точки  $x_2$  и  $\bar{x}$  и сравниваются значения  $f(x)$  в этих точках. Если окажется, что  $\bar{x} = x_2$ , то в качестве второй пробной точки (помимо  $x_2$ ) можно взять любую другую точку интервала  $(x_1; x_3)$ , например,  $(x_1 + x_2)/2$  или  $(x_2 + x_3)/2$ . Начало и конец нового отрезка, а также пробная точка, попавшая на него, образуют тройку точек, обладающих свойством (4.9).

Заметим, что на каждой итерации метода парабол, кроме первой, определяется только одно новое значение  $f(x)$ .

Условием окончания поиска служит близость к нулю разности  $\Delta$  чисел  $\bar{x}$ , найденных на данной и предыдущей итерациях, т.е. неравенство  $|\Delta| \leq \varepsilon$ , где  $\varepsilon$  – заданное число, характеризующее точность определения  $x^*$ .

#### *Алгоритм метода парабол*

1. Выбрать точки  $x_1, x_2, x_3$ , удовлетворяющие условиям (4.9).  
Перейти к шагу 2.
2. Найти  $\bar{x}$  по формуле (2.10). На первой итерации перейти к шагу 4, на остальных – к шагу 3.
3. Проверить окончание поиска. Если  $|\Delta| \leq \varepsilon$ , то поиск завершить, полагая  $x^* = \bar{x}$ ,  $f^* = f(\bar{x})$ , иначе – перейти к шагу 4.
4. Вычислить значение  $f^* = f(\bar{x})$ . Перейти к шагу 5.
5. Определить новую тройку  $x_1, x_2, x_3$ . Присвоить  $f(x_1)$ ,  $f(x_2)$  и  $f(x_3)$  соответствующие значения  $f(x)$ , найденные ранее.  
Перейти к шагу 2.

### **4.3. Методы, использующие производные функции**

Прямые методы, рассмотренные в разд. 4.2, используются при минимальных требованиях к целевой функции  $f(x)$  – она считается унимодальной и вычислению (измерению) подлежат значения только самой функции, но не ее производных. Если усилить эти требования, предположив, что  $f(x)$  является дифференцируемой или дважды дифференцируемой выпуклой функцией и считать, что возможно вы-

числение производных  $f'(x)$  в произвольно выбранных точках, то эффективность процедур поиска точки минимума можно существенно повысить.

Рассмотрим методы минимизации, в которых используются значения производных целевой функции. Напомним, что для выпуклой дифференцируемой функции равенство  $f'(x) = 0$  является не только необходимым, но и достаточным условием глобального минимума.

Поэтому если известно, что  $x^*$  является внутренней точкой отрезка  $[a; b]$ , то приближенное равенство  $f'(x) \approx 0$  или  $|f'(x)| \leq \varepsilon$ , где  $\varepsilon > 0$  – достаточно малое число, может служить условием остановки вычислений в рассматриваемых ниже методах.

### Метод средней точки

Если определение значений производной  $f'(x)$  не представляет затруднений, то в процедуре исключения отрезков метода деления отрезка пополам вычисление двух значений  $f'(x)$  вблизи середины очередного отрезка можно заменить вычислением одного значения  $f'(x)$  в его средней точке  $\bar{x} = (a + b)/2$ .

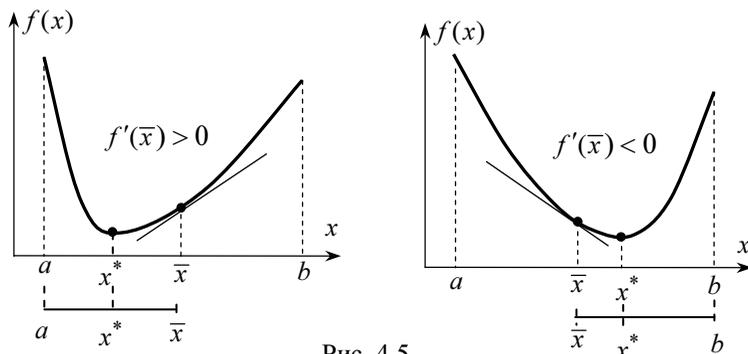


Рис. 4.5

В самом деле, если  $f'(\bar{x}) > 0$ , то точка  $\bar{x}$  лежит на участке монотонного возрастания  $f(x)$ , поэтому  $x^* < \bar{x}$ , и точку минимума следует искать на отрезке  $[a, \bar{x}]$ . При  $f'(\bar{x}) < 0$  имеем противоположную ситуацию и переходим к отрезку  $[\bar{x}, b]$  (рис. 4.5). Равенство  $f'(\bar{x}) = 0$  означает, что точка минимума найдена точно:  $x^* = \bar{x}$ .

Такое исключение отрезков требует на каждой итерации только одного вычисления  $f'(x)$  и уменьшает отрезок поиска точки  $x^*$  ровно вдвое.

### Метод хорд

Как уже отмечалось, равенство  $f'(x) = 0$  является необходимым и достаточным условием глобального минимума выпуклой дифференцируемой функции  $f(x)$ . Поэтому если на концах отрезка  $[a, b]$  производная  $f'(x)$  имеет разные знаки, т.е.  $f'(a) \cdot f'(b) < 0$ , то на интервале  $[a, b]$  найдется точка, в которой  $f'(x)$  обращается в нуль, и поиск точки минимума  $f(x)$  на  $[a, b]$  эквивалентен решению уравнения

$$f'(x) = 0, \quad x \in (a, b) \quad (4.11)$$

Отсюда следует, что при  $f'(a) \cdot f'(b) < 0$  любой приближенный метод решения уравнения (4.11) можно рассматривать как метод минимизации выпуклой дифференцируемой функции  $f(x)$  на отрезке

$[a, b]$ .

В курсе *математического анализа* рассматривается метод хорд приближенного решения уравнения  $F(x) = 0$  на отрезке  $[a, b]$  при  $F(x) \cdot F(b) < 0$ . Этот метод основан на исключении отрезков путем определения точки  $\tilde{x}$  пересечения с осью  $Ox$  хорды графика функции  $F(x)$  на очередном

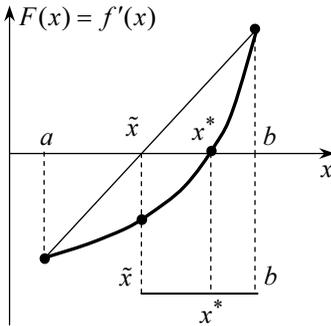


Рис. 4.6

отрезке (рис. 4.6).

Полагаем, что  $F(x) = f'(x)$ . На основании уравнения отрезка прямой, соединяющего две точки  $(a, f'(a))$  и  $(b, f'(b))$ , записываем координату точки  $\tilde{x}$ , в которой  $f'(\tilde{x}) = 0$ :

$$\tilde{x} = a - \frac{f'(a)}{f'(a) - f'(b)}(a - b). \quad (4.12)$$

Отрезок дальнейшего поиска точки  $x^*$   $[a, \tilde{x}]$  или  $[\tilde{x}, b]$  выбирается в зависимости от знака  $f'(\tilde{x})$  так же, как в методе средней точки. На каждой итерации, кроме первой, следует вычислять одно новое значение  $f'(x)$ .

#### Алгоритм метода хорд

1. Найти  $\tilde{x}$  по формуле (4.12). Вычислить значение  $f'(\tilde{x})$ . Перейти к шагу 2.
2. Проверка на окончание поиска: если  $|f'(\tilde{x})| \leq \varepsilon$ , то положить  $x^* = \tilde{x}$ ,  $f^* = f(\tilde{x})$  и поиск завершить, иначе – перейти к шагу 3.
3. Переход к новому отрезку. Если  $f'(\tilde{x}) > 0$ , то положить  $b = \tilde{x}$ ,  $f'(b) = f'(\tilde{x})$ , иначе – положить  $a = \tilde{x}$ ,  $f'(a) = f'(\tilde{x})$ . Перейти к шагу 1.

Замечание. До сих пор предполагалось, что  $f'(a) \cdot f'(b) < 0$ , т.е. производная  $f'(x)$  на концах отрезка имеет разные знаки. При нарушении этого условия точку  $x^*$  можно указать сразу. Так, если  $f'(a), f'(b) > 0$ , то  $f(x)$  возрастает на  $[a, b]$ , следовательно,  $x^* = a$ , а при  $f'(a), f'(b) < 0$  она убывает и  $x^* = b$ . В случае  $f'(a) \cdot f'(b) = 0$ ,  $x^* = a$  или  $x^* = b$  в зависимости от того, на каком из концов отрезка  $[a, b]$   $f'(x) = 0$ .

#### Метод Ньютона

Предположим, что  $f(x)$  – дважды дифференцируемая функция, причем  $f''(x) > 0$  (это гарантирует выпуклость  $f(x)$ ). Тогда корень

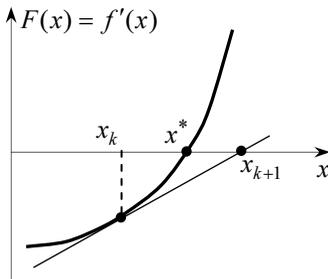


Рис. 4.7.

уравнения  $f'(x) = 0$  можно искать приближенно, используя *метод касательных*. Этот метод решения уравнения  $F(x) = 0$  состоит в построении последовательных приближений  $x_k$ ,  $k = 0, 1, \dots$  следующим образом. В очередной точке  $x_k$  строится линейная аппроксимация функции  $F(x)$  (каса-

тельная к графику  $F(x)$ ) и точка, в которой линейная аппроксимирующая функция обращается в нуль, используется в качестве следующего приближения  $x_{k+1}$  (рис. 4.7).

Уравнение касательной к графику  $F(x)$  в точке  $x = x_k$ , как известно, имеет вид  $F(x) = F(x_k) + F'(x_k)(x - x_k)$ , поэтому точка  $x = x_{k+1}$ , найденная из условия  $F(x) = 0$ , определяется формулой  $x_{k+1} = x_k - F(x_k)/F'(x_k)$ .

Положим  $F(x) = f'(x)$ , тогда для решения уравнения  $f'(x) = 0$  необходимо построить последовательность

$$x_{k+1} = x_k - f'(x_k)/f''(x_k), \quad k = 0, 1, \dots, \quad (4.13)$$

где  $x_0$  – точка, выбранная в качестве начального приближения.

Вычисления по формуле (4.13) производят до тех пор, пока не выполнится неравенство  $|f'(x)| \leq \varepsilon$ , после чего полагают  $x^* \approx x_k$ . Описанная процедура поиска точки минимума называется *методом Ньютона*.

Для функций  $f(x)$ , удовлетворяющих определенным требованиям (например, квадратичных функций) метод Ньютона гарантирует высокую скорость сходимости к точке минимума. В общем случае, если начальное приближение  $x_0$  выбрано не достаточно близким к точке  $x^*$ , то последовательность (4.13) метода Ньютона может расходиться. В подобных случаях необходимо найти лучшее начальное приближение  $x_0$ , например, с помощью нескольких итераций метода золотого сечения.

#### 4.4. Методы минимизации многомодальных функций

При использовании методов численной минимизации существенную роль играет унимодальность целевой функции  $f(x)$ . Если  $f(x)$  этим свойством не обладает, т.е. является многомодальной, то поиск ее точки минимума с помощью упомянутых методов может оказаться безрезультатным или привести к ошибке (будет найдена точка локального экстремума, отличного от глобального минимума  $f(x)$ ). Кроме того, во многих практических случаях исследование унимодальности целевой функции вызывает затруднения или известно, что она унимодальной не является.

## Метод перебора

Применение метода перебор строго обосновано лишь для унимодальной на отрезке  $[a, b]$  функции  $f(x)$ . Однако, если вместо унимодальности потребовать, чтобы функция  $f(x)$  удовлетворяла на  $[a, b]$  условию Липшица (4.4), то можно гарантировать определение минимального значения  $f^*$  методом перебора с любой заданной точностью. Сформулируем более строго это утверждение.

Пусть функция  $f(x)$  удовлетворяет на отрезке  $[a, b]$  условию Липшица (4.4) с константой  $L$  и приближенные значения  $x^* \approx x_m$ ,  $f^* \approx f(x_m)$  найдены методом перебора с разбиением отрезка  $[a, b]$  на  $n$  частей. Тогда для погрешности  $\varepsilon_n$  определения минимального значения  $f^*$  справедлива оценка

$$\varepsilon_n = f(x_m) - f^* \leq L \frac{b-a}{2n}. \quad (4.14)$$

Действительно, пусть  $x^*$  – точка минимума  $f(x)$  на  $[a, b]$ . Существование точки минимума  $x^* \in [a, b]$  вытекает из теоремы Вейерштрасса, так как условие Липшица гарантирует непрерывность  $f(x)$  на  $[a, b]$ . Очевидно, среди пробных точек  $x_i = a + i(b-a)/n$ ,  $i = 0, 1, \dots, n$  метода перебора найдется такая точка  $x_i$ , что  $|x_i - x^*| \leq (b-a)/2n$ . Поэтому с учетом условия Липшица можно записать:

$$0 \leq f(x_m) - f^* \leq f(x_i) - f^* \leq L |x_i - x^*| \leq L \frac{b-a}{2n}.$$

### Замечания

1. Из оценки (4.14) следует, что в результате  $N$  вычислений  $f(x)$  метод перебора позволяет найти минимум  $f^*$  с точностью

$$\varepsilon(N) = L \frac{b-a}{2(N-1)}.$$

2. Для обеспечения точности  $\varepsilon$  определения минимального значения  $f^*$  методом перебора число  $n$  должно быть выбрано из условия

$$\varepsilon_n = L \frac{(b-a)}{2n} \leq \varepsilon, \text{ т.е. } n \geq L \frac{(b-a)}{2\varepsilon}.$$

3. Если функция  $f(x)$  многомодальна, то погрешность определения ее точки минимума  $x^*$  методом перебора может быть значительной, несмотря на то, что минимум функции  $f^*$  найден достаточно точно (рис. 4.8). Однако во многих случаях практический интерес представляют те значения аргумента  $x$  (возможно, и далекие от  $x^*$ ), при которых целевая функция принимает значения, достаточно близкие к минимальному. Это обуславливает использование метода перебора и некоторых других методов для минимизации многомодальных функций.

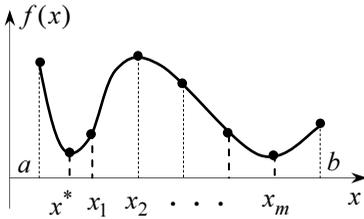


Рис. 4.8.

Однако во многих случаях практический интерес представляют те значения аргумента  $x$  (возможно, и далекие от  $x^*$ ), при которых целевая функция принимает значения, достаточно близкие к минимальному. Это обуславливает использование метода перебора и некоторых других методов для минимизации многомодальных функций.

### Метод ломаных

Этот прямой метод также рассчитан на минимизацию многомодальных функций, удовлетворяющих условию Липшица. В нем используются *кусочно-линейные аппроксимации* функции  $f(x)$ , графиками которых являются ломаные, что и объясняет название метода.

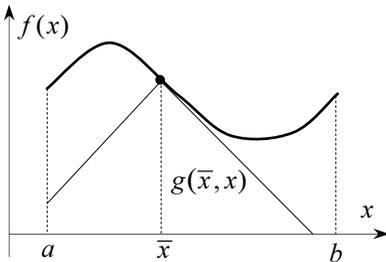


Рис. 4.9

Опишем построение аппроксимирующих кусочно-линейных функций  $p_k(x)$ .

Пусть функция  $f(x)$  удовлетворяет на отрезке  $[a, b]$  условию Липшица (4.4) с константой  $L$ . Зафиксируем точку  $\bar{x} \in [a, b]$  и рассмотрим вспомогательную функцию одной переменной

$$g(\bar{x}, x) = f(\bar{x}) - L|\bar{x} - x|.$$

Ее график показан на рис. 4.9.

Аппроксимирующие кусочно-линейные функции  $p_k(x)$ ,  $k = 0, 1, \dots$  строятся следующим образом.

Рассмотрим прямые

$$y = f(a) - L(x - a) \quad \text{и} \quad y = f(b) + L(x - b).$$

Они пересекаются в точке  $(x_0, y_0)$  координатами

$$x_0 = \frac{1}{2L}[f(a) - f(b) + L(a+b)], \quad y_0 = \frac{1}{2}[f(a) + f(b) + L(a-b)].$$

(4.15)

Положим

$$p_0(x) = \begin{cases} f(a) - L(x-a) & \text{при } x \in [a; x_0], \\ f(b) - L(x-b) & \text{при } x \in [x_0; b]. \end{cases}$$

График функции  $p_0(x)$  показан на рис. 4.10,а, ее точка минимума  $x_0^* = x_0$ , а минимальное значение  $p_0^* = y_0$ .

Используя вспомогательную функцию  $g(x_0^*, x)$ , определяем следующую аппроксимирующую функцию:

$$p_1(x) = \max[p_0(x), g(x_0^*, x)],$$

у которой по сравнению с  $p_0(x)$  исчезла точка минимума  $x_0^*$ , но появились две новые точки локального минимума  $x_1'$  и  $x_1''$  (рис. 4.10,б):

$$x_1' = x_0^* - \Delta_1, \quad x_1'' = x_0^* + \Delta_1, \quad \text{где } \Delta_1 = \frac{1}{2L}[f(x_0^*) - p_0^*], \quad (4.16)$$

причем  $p_1(x_1') = p_1(x_1'') = p_1 = \frac{1}{2}[f(x_0^*) - p_0^*]$ .

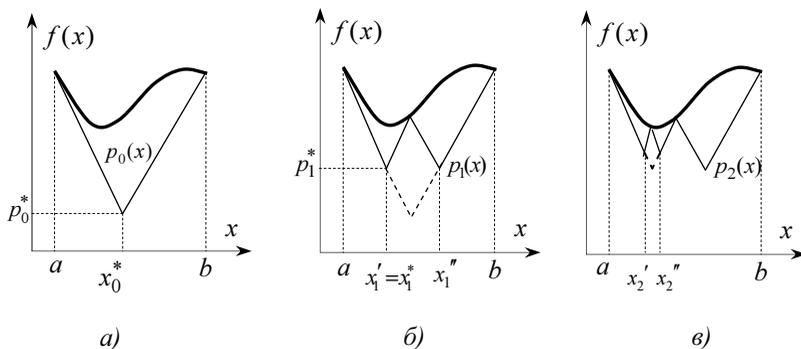


Рис. 4.10.

Формулы (4.16) легко получаются с помощью элементарной геометрии с учетом того, что тангенс угла наклона каждого звена ломаной  $p_1(x)$  к горизонтальной оси по модулю равен  $L$ .

Выберем любую из точек глобального минимума  $p_1^*$  функции  $p_1(x)$ , обозначим ее через  $x_1^*$  (в данном случае это  $x_1'$  или  $x_1''$ , считаем, например, что  $x_1^* = x_1'$ ) и положим

$$p_2(x) = \max[p_1(x), g(x_1^*, x)].$$

У функции  $p_2(x)$  по сравнению с  $p_1(x)$  вместо  $x_1^*$  появились две новые точки локального минимума  $x_2'$  и  $x_2''$  (рис. 4.10, в), которые находятся аналогично (4.16):

$$x_2' = x_1^* - \Delta_2, \quad x_2'' = x_1^* + \Delta_2, \quad \text{где } \Delta_2 = \frac{1}{2L} [f(x_1^*) - p_1^*], \quad (4.17)$$

причем  $p_2(x_2') = p_2(x_2'') = p_2 = \frac{1}{2} [f(x_1^*) - p_1^*]$

Пусть функция  $p_{k-1}(x)$  построена. Выбрав произвольную точку глобального минимума  $p_{k-1}^*$  функции  $p_{k-1}(x)$  и обозначив ее через  $x_{k-1}^*$ , определим функцию

$$p_k(x) = \max[p_{k-1}(x), g(x_{k-1}^*, x)].$$

Новые точки локального минимума  $x_k'$  и  $x_k''$  функции  $p_k(x)$ , появившиеся взамен  $x_{k-1}^*$ , а также значения  $p_k(x)$  в этих точках находятся по формулам, аналогичным (4.16) и (4.17):

$$x_k' = x_{k-1}^* - \Delta_k, \quad x_k'' = x_{k-1}^* + \Delta_k, \quad \text{где } \Delta_k = \frac{1}{2L} [f(x_{k-1}^*) - p_{k-1}^*],$$

$$p_k(x_k') = p_k(x_k'') = p_k = \frac{1}{2} [f(x_{k-1}^*) - p_{k-1}^*]. \quad (4.18)$$

Перечислим некоторые свойства функций  $p_k(x)$ , вытекающие из способа их построения:

1. Функция  $p_k(x)$  – непрерывная кусочно-линейная, ее графиком является ломаная, состоящая из отрезков с угловыми коэффициентами  $\pm L$ . У функции  $p_k(x)$  существует ровно  $k+1$  локальных минимумов, точками которых являются абсциссы вершин ломаной, а сами эти минимумы равны ординатам вершин.

2. Для всех  $x \in [a, b]$  и  $k=1, 2, \dots$  справедливы неравенства  $p_{k-1}(x) \leq p_k(x) \leq f(x)$ .

Из этих свойств и метода построения функций  $p_k(x)$  следует, что с ростом  $k$  ломаные  $p_k(x)$  приближаются снизу к графику функции  $f(x)$  вблизи точек ее глобального минимума. Идея метода ломаных состоит в том, чтобы искать глобальный минимум не функции  $f(x)$ , а ломаных  $p_k(x)$ , что, вообще говоря, значительно проще, так как точками минимума  $p_k(x)$  могут быть только абсциссы вершин этой ломаной, определяемые в ходе построения последовательности  $\{p_k(x)\}$ . Для достаточно большого  $k$  полагают:  $x_k^* \approx x_{k-1}^*$ ,  $f^* \approx f(x_{k-1}^*)$ .

Обоснованием метода ломаных являются следующие факты, доказательство которых можно найти в [3].

Пусть функция  $f(x)$  удовлетворяет на отрезке  $[a, b]$  условию Липшица с константой  $L$ . Тогда последовательность  $\{x_k^*\}$  точек минимума функций  $p_k(x)$ , построенных по методу ломаных, обладает следующими свойствами:

1.  $\lim_{k \rightarrow \infty} f(x_k^*) = f^*$ .

2. Если точка глобального минимума  $x^*$  функции  $f(x)$  на  $[a, b]$  единственна, то  $\lim_{k \rightarrow \infty} x_k^* = x^*$ . Если множество  $U^*$  точек минимума  $x^*$  содержит более одной точки, то справедливо более общее соотношение:  $\lim_{k \rightarrow \infty} \rho_k = 0$ , где  $\rho_k$  – расстояние от точки  $x^*$  до ближайшей к ней точки множества  $U^*$ .

3. Справедлива следующая оценка достигнутой точности  $\varepsilon_k$  приближения  $f(x_{k-1}^*)$  к минимуму  $f^*$ :

$$0 \leq \varepsilon_k = f(x_{k-1}^*) - f^* \leq f(x_{k-1}^*) - p_{k-1}^* = 2L \Delta_k, \quad (4.19)$$

которая используется в качестве условия окончания поиска минимума методом ломаных.

Замечание. При построении функции  $p_k(x)$ ,  $k=1, 2, \dots$  используются только координаты  $(x, p)$  вершин ломаной  $p_{k-1}(x)$ , найденные ранее. Поэтому на  $k$ -й итерации построения ломаных следует из найденных до этого  $k$  пар чисел  $(x, p)$  (координат вершин, соответствующих локальным минимумам  $p_{k-1}(x)$ ) выбрать пару

$(x_{k-1}^*, p_{k-1}^*)$ , относящуюся к глобальному минимуму  $p_{k-1}(x)$  (это пара  $(x, p)$  с минимальным значением  $p$ ), и вместо  $(x_{k-1}^*, p_{k-1}^*)$  ввести в рассмотрение две новые пары  $(x_k', p_k)$  и  $(x_k'', p_k)$ , найденные по формулам (4.18).

*Алгоритм метода ломаных:*

1. Вычислить  $x_0$  и  $y_0$  по формулам (4.15) и определить пару  $(x, p)$ , полагая  $x = x_0$ ,  $p = y_0$ .

2. Сравнивая значения  $p$  введенных в рассмотрение пар  $(x, p)$ , определить пару  $(x_*, p_*)$ , у которой это значение минимально. Вычислить  $f(x_*)$ .

3. Проверить окончание поиска с помощью условия (4.19). Вычислить  $\Delta = [f(x_*) - p_*]/(2L)$  и положить  $\varepsilon_k = 2L\Delta$ . Если  $\varepsilon_k \leq \Delta$ , то поиск завершить, полагая  $x^* = x_*$  и  $f^* = f(x_*)$ , иначе – перейти к шагу 4.

4. Определить новые пары  $(x', p)$  и  $(x'', p)$ , полагая  $x' = x_* - \Delta$ ,  $x'' = x_* + \Delta$ ,  $\Delta = [f(x_*) + p_*]/2$  и ввести их в рассмотрение вместо пары  $(x_*, p_*)$ . Перейти к шагу 2.

Как видно из формул (4.18), на каждой итерации метода ломаных, кроме первой, вычисляется только одно значение  $f(x)$ .

Численные эксперименты, результаты которых приведены в специальной литературе, подтверждают преимущества методов полиномиальной аппроксимации и Ньютона по сравнению с другими методами минимизации, особенно когда необходима высокая точность определения точки минимума.

Однако при ухудшении свойств функции  $f(x)$  (например, если  $f(x)$  не выпукла, быстро изменяется, имеет разрывные производные и т.п.) алгоритмы минимизации, основанные на указанных методах, по эффективности и надежности могут уступать прямым методам, которые практически не чувствительны к свойствам функции, если она унимодальна.

## Глава 5

### МЕТОДЫ БЕЗУСЛОВНОЙ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ

Глава посвящена исследованию задач оптимизации, сводящихся к поиску точек минимума функции многих переменных на всем пространстве. С ростом размерности пространства переменных, как правило, возрастают объем вычислений и сложность алгоритмов, а также затрудняется анализ поведения целевой функции.

На основании общих принципов многомерной оптимизации рассмотрены прямые методы, методы первого и второго порядков безусловной оптимизации.

#### 5.1. Предварительные сведения

Будем рассматривать функции многих переменных  $f = f(x_1, \dots, x_n)$ , заданные в точках  $n$ -мерного евклидова пространства  $R^n$ :  $f = f(x)$ . Точки  $x \in R^n$  представляются векторами-столбцами координат:  $x = (x_1, \dots, x_n)^T$ , где символ « $T$ » – знак транспонирования. В дальнейшем, там, где это не приводит к недоразумениям, символ « $T$ » будем опускать.

В дополнение к разд. 2.2 рассмотрим некоторые понятия из курса линейной алгебры, которые будут использованы в дальнейшем.

Напомним определение длины (нормы) вектора  $x$ :

$$\|x\| = \sqrt{\langle x, x \rangle} = \left( \sum_{j=1}^n x_j^2 \right)^{1/2}$$

и расстояния между векторами  $x$  и  $y$  (точками пространства  $R^n$ ):

$$\rho(x, y) = \|x - y\| = \left[ \sum_{j=1}^n (x_j - y_j)^2 \right]^{1/2}.$$

Для норм произвольных векторов  $x, y \in R^n$  справедливо неравенство треугольника:

$$\|x - y\| \leq \|x\| + \|y\|.$$

Скалярное произведение оценивается по модулю *неравенством Коши-Буняковского*

$$|\langle x, y \rangle| \leq \|x\| \cdot \|y\|.$$

Напомним основные понятия, связанные с числовыми матрицами.

Матрица  $A = (a_{ij})$ ,  $i = \overline{1, m}$ ;  $j = \overline{1, n}$  представляет собой прямоугольный массив (таблицу) чисел, состоящий из  $m$  строк и  $n$  столбцов. Таким образом, вектор-столбец  $x$  является матрицей размера  $(n \times 1)$ .

Матрица  $A^T = (a_{ji})$ ,  $j = \overline{1, n}$ ;  $i = \overline{1, m}$ , которая получается из матрицы  $A = (a_{ij})$ , если поменять местами ее строки и столбцы, называется *транспонированной* по отношению к матрице  $A$ .

Квадратная матрица  $A$  называется *симметрической*, если  $A^T = A$ .

Матрицы одинакового размера  $A = (a_{ij})$  и  $B = (b_{ij})$  можно складывать:  $A + B = (a_{ij} + b_{ij})$ .

Результатом умножения матрицы  $A$  на число  $\lambda$  является матрица  $\lambda A = (\lambda a_{ij})$ .

Произведением  $Ax$  матрицы  $A = (a_{ij})$  размера  $(m \times n)$  на вектор-столбец  $x \in R^n$  называется вектор-столбец  $b \in R^m$ , координаты которого вычисляются по формуле  $b_i = \sum_{j=1}^n a_{ij} x_j = \langle a^{(i)}, x \rangle$ ,  $i = \overline{1, m}$ , где  $a^{(i)} = (a_{i1}, \dots, a_{in})$  – вектор коэффициентов  $i$ -й строки матрицы  $A$ .

Для матриц  $A = (a_{ij})$  и  $B = (b_{jk})$  соответственно размера  $(m \times n)$  и  $(n \times r)$  определено произведение  $AB = C = (c_{ik})$ , где элемент  $c_{ik}$  матрицы  $C$  размера  $(m \times r)$  определяется равенством

$$c_{ik} = \sum_{j=1}^n a_{ij} b_{jk}, \text{ где } i = \overline{1, m}, j = \overline{1, n}, k = \overline{1, r}.$$

Можно показать, что  $(AB)^T = B^T A^T$ .

Если рассматривать  $n$ -мерные векторы-столбцы  $x$  и  $y$  как матрицы размера  $(n \times 1)$ , то формулу для их скалярного произведения можно получить по правилу умножения матриц  $x^T$  и  $y$ :

$$\langle x, y \rangle = (x_1, \dots, x_n) \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix} = \sum_{j=1}^n x_j y_j .$$

Заметим, что для  $x, y \in R^n$  произведение  $x \cdot y^T$  задает квадратную матрицу (векторное произведение):

$$x \cdot y^T = \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix} (y_1, \dots, y_n) = \begin{pmatrix} x_1 y_1 & \dots & x_1 y_n \\ \dots & \dots & \dots \\ x_n y_1 & \dots & x_n y_n \end{pmatrix} .$$

Если  $A$  – квадратная симметрическая матрица размера  $(n \times n)$ , то для любых векторов  $x, y \in R^n$  имеет место  $\langle Ax, y \rangle = \langle x, Ay \rangle$ , так как  $\langle Ax, y \rangle = (Ax)^T y = x^T A^T y = x^T Ay = \langle x, Ay \rangle$ .

Каждой квадратной матрице размера  $(n \times n)$  можно поставить в соответствие число – *определитель* матрицы  $A$  (обозначается  $\det A$  или  $|A|$ ), который вычисляется по формуле

$$\det A = |A| = \sum_{i=1}^n a_{ij} A_{ij} ,$$

где *алгебраическое дополнение*  $A_{ij}$  элемента  $a_{ij}$  определяется соотношением  $A_{ij} = (-1)^{i+j} M_{ij}$  (*минор*  $M_{ij}$  – это определитель матрицы, полученной из матрицы  $A$  вычеркиванием  $i$ -й строки и  $j$ -го столбца). Квадратная матрица  $A$  называется *вырожденной*, если ее определитель равен нулю, и *невырожденной* в противном случае.

Для каждой невырожденной матрицы  $A$  существует *обратная матрица*  $A^{-1} = (\alpha_{ij})$  такая, что  $A^{-1}A = AA^{-1} = E$ , где  $E = (e_{ij})$  – *единичная матрица*:

$$e_{ij} = \begin{cases} 1, & \text{если } i = j, \\ 0, & \text{если } i \neq j, \quad i = 1, \dots, n; \quad j = 1, \dots, n. \end{cases}$$

Элементы обратной матрицы могут быть найдены по формуле:

$$\alpha_{ij} = \frac{A_{ji}}{\det A} ,$$

где  $A_{ji}$  – алгебраическое дополнение элемента  $a_{ji}$  матрицы  $A$ .

Пусть  $A = (a_{ij})$  – симметрическая матрица размера  $(n \times n)$ . Тогда функция  $n$  переменных  $h_1, \dots, h_n$

$$Q(\mathbf{h}) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} h_i h_j = \langle A\mathbf{h}, \mathbf{h} \rangle$$

называется *квадратичной формой* этих переменных, а матрица  $A$  – матрицей квадратичной формы.

Квадратичная форма  $Q(\mathbf{h})$  называется *положительно определенной*, если для всех  $\mathbf{h} \neq 0$  имеет место неравенство  $Q(\mathbf{h}) > 0$ .

Из курса линейной алгебры известен *критерий Сильвестра* положительной определенности квадратичной формы: для того, чтобы квадратичная форма  $Q(\mathbf{h}) = \langle A\mathbf{h}, \mathbf{h} \rangle$  была положительно определенной, необходимо и достаточно, чтобы ее матрица  $A = (a_{ij})$  была положительно определена, т.е. все ее угловые миноры были положительными:

$$\Delta_1 = a_{11} > 0; \quad \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0; \dots; \quad \Delta_n = \begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix} > 0. \quad (5.1)$$

Ненулевой вектор  $\xi$ , для которого  $A\xi = \lambda\xi$ , называется *собственным вектором* квадратной матрицы  $A$ , а число  $\lambda$  – соответствующим ему *собственным значением* этой матрицы.

Собственные значения находятся из характеристического уравнения  $\det(A - \lambda E) = 0$ . Если  $\lambda_i$  – собственное значение матрицы  $A$ , то нетривиальное решение однородной системы линейных уравнений  $\det(A - \lambda_i E)\xi = 0$  дает соответствующий ему собственный вектор. Собственные значения симметрической положительно определенной матрицы  $A$  положительны, и существует ортонормированный базис в  $R^n$  из собственных векторов  $\xi^{(1)}, \dots, \xi^{(n)}$  матрицы  $A$ . В этом базисе матрица  $A$  имеет диагональный вид: на ее главной диагонали стоят собственные значения  $\lambda_1, \dots, \lambda_n$  а на остальных местах – нули.

*Нормой матрицы*  $A$  размера  $(n \times n)$  называется число  $\|A\| = \max_{\|\eta\|=1} \|A\eta\|$ . Очевидно, для произвольного вектора  $x \in R^n$  выполняется неравенство:

$$\|Ax\| \leq \|A\| \cdot \|x\|.$$

Норма симметрической положительно определенной матрицы удовлетворяет двойному неравенству

$$l \leq \|A\| \leq L,$$

где  $l$  и  $L$  – ее наименьшее и наибольшее собственные значения.

Справедлива оценка  $\|A\| \leq \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2}$ .

### **Минимум функции многих переменных**

Обобщим некоторые определения, сформулированные в гл. 4 для функции одной переменной, на случай функций многих переменных. Пусть функция  $f(x)$ ,  $x \in R^n$  определена во всем пространстве  $R^n$ .

Точка  $x^* \in R^n$  называется точкой *глобального минимума* функции  $f(x)$ , если для всех  $x \in R^n$  выполняется неравенство  $f(x^*) \leq f(x)$ . Значение  $f(x^*) = \min_{x \in R^n} f(x) = f^*$  называется *минимумом* функции. Множество всех точек глобального минимума функции  $f(x)$  будем обозначать через  $U^*$ .

Если  $U^* = \emptyset$ , то вместо минимума функции  $f(x)$  иногда рассматривают ее точную нижнюю грань  $f_* = \inf_{x \in R^n} f(x)$ , определение которой в  $n$ -мерном случае практически не отличается от определения, данного в разд. 4.1.

Точка  $\tilde{x} \in R^n$  называется точкой *локального минимума* функции  $f(x)$ , если существует  $\varepsilon$ -окрестность точки  $\tilde{x}: U_\varepsilon(\tilde{x}) = \{x \mid \rho(x, \tilde{x}) < \varepsilon\}$  такая, что для всех  $x \in U_\varepsilon(\tilde{x})$  выполняется неравенство  $f(\tilde{x}) \leq f(x)$ .

Если допустимое множество  $U$  в задаче минимизации (максимизации) функции  $n$  переменных совпадает со всем пространством  $R^n$ , то говорят о задаче *безусловной оптимизации*

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

### **Дифференцируемые функции многих переменных**

Многие алгоритмы минимизации и критерии оптимальности в  $R^n$  используются только для функций, дифференцируемых необходимое число раз.

Напомним некоторые факты, известные из курса математического анализа.

1. Если функция дифференцируема в точке  $x^{(0)} \in R^n$ , то ее приращение  $\Delta f(x^{(0)}) = f(x^{(0)} + \Delta x) - f(x^{(0)})$  можно записать в виде

$$\Delta f(x^{(0)}) = df(x^{(0)}) + o(\|x^{(0)}\|),$$

где  $df(x^{(0)}) = \sum_{j=1}^n \frac{\partial f(x^{(0)})}{\partial x_j} \Delta x_j$  – первый дифференциал  $f(x)$  в точке  $x^{(0)}$ .

2. Вектор  $f'(x^{(0)}) = \left( \frac{\partial f(x^{(0)})}{\partial x_1}, \dots, \frac{\partial f(x^{(0)})}{\partial x_n} \right)$  – называется *градиентом* функции  $f(x)$  в точке  $x^{(0)}$ .

В малой окрестности точки  $x^{(0)}$  градиент указывает направление наискорейшего возрастания функции  $f(x)$ , а его норма характеризует скорость этого возрастания. Градиент в точке  $x^{(0)}$  перпендикулярен линии (поверхности) уровня  $f(x) = c$ , проходящей через эту точку. Очевидно,  $df(x^{(0)}) = \langle f'(x^{(0)}), \Delta x \rangle$ , поэтому

$$\Delta f(x^{(0)}) = \langle f'(x^{(0)}), \Delta x \rangle + o(\|\Delta x\|). \quad (5.2)$$

3. Если функция  $f(x)$  дважды дифференцируема в точке  $x^{(0)} \in R^n$ , то

$$\Delta f(x^{(0)}) = df(x^{(0)}) + \frac{1}{2} d^2f(x^{(0)}) + o(\|x^{(0)}\|^2),$$

где  $d^2f(x^{(0)}) = \sum_{j=1}^n \frac{\partial^2 f(x^{(0)})}{\partial x_i \partial x_j} \Delta x_i \Delta x_j$  – второй дифференциал  $f(x)$  в точке  $x^{(0)}$ .

Используя матрицу вторых производных (*матрицу Гессе, гессиан*)

$f''(x^{(0)}) = \left( \frac{\partial^2 f(x^{(0)})}{\partial x_i \partial x_j} \right)$ , второй дифференциал можно записать так:

$d^2f(x^{(0)}) = \langle f''(x^{(0)}) \Delta x, \Delta x \rangle$ , поэтому

$$\Delta f(x^{(0)}) = \langle f'(x^{(0)}), \Delta x \rangle + \langle f''(x^{(0)}) \Delta x, \Delta x \rangle / 2 + o(\|\Delta x\|^2). \quad (5.3)$$

4. Из формул (5.2) и (5.3) следует, что для малых  $\|\Delta x\|$

$$f(x) = f(x^{(0)}) + \langle f'(x^{(0)}), \Delta x \rangle \quad (5.4)$$

или

$$f(x) = f(x^{(0)}) + \langle f'(x^{(0)}), \Delta x \rangle + \langle f''(x^{(0)}) \Delta x, \Delta x \rangle / 2, \quad (5.5)$$

т.е. в малой окрестности точки  $x^{(0)}$  поведение дифференцируемой функции  $f(x)$  приближенно описывается формулой (5.4), а дважды

дифференцируемой – формулой (5.5), причем представление (5.5) является более точным.

Из курса математического анализа известны следующие условия минимума функции  $n$  переменных:

1. Если в точке  $x^{(0)} \in R^n$  функция  $f(x)$  дифференцируема и достигает локального минимума, то

$$\left. \frac{f(x)}{\partial x_j} \right|_{x=x^{(0)}} = 0, \quad j = \overline{1, n} \quad \text{или} \quad f'(x^{(0)}) = 0 \quad (5.6)$$

(необходимое условие минимума). Точки, в которых выполнено условие (5.6), называются *стационарными точками* дифференцируемой функции  $f(x)$ .

2. Если в стационарной точке  $x^{(0)} \in R^n$  функция  $f(x)$  дважды дифференцируема и матрица ее вторых производных  $f''(x^{(0)})$  положительно определена, то  $x^{(0)}$  есть точка локального минимума функции  $f(x)$  (*достаточное условие минимума*).

Условия 1 и 2 лежат в основе *классического метода минимизации функций*, дифференцируемых во всем пространстве  $R^n$ . Однако классический метод минимизации функций многих переменных имеет ограниченное практическое применение в основном из-за трудностей в аналитическом решении системы уравнений (5.6).

## 5.2. Выпуклые множества и выпуклые функции

Решение задач минимизации в  $R^n$ , как правило, сопряжено со значительными трудностями, особенно для многоэкстремальных функций. Многие из этих трудностей устраняются, если ограничиться рассмотрением только *выпуклых* целевых функций.

### Свойства выпуклых функций

**Определение 5.1.** Пусть  $x, y \in R^n$ , Множество  $\{z\} \subset R^n$  точек вида

$$z = \alpha x + (1 - \alpha)y, \quad \alpha \in [0; 1] \quad (5.7)$$

называется *отрезком*, соединяющим точки  $x$  и  $y$ .

В пространстве  $R^n$ ,  $n \leq 3$  соотношение (5.7) определяет обычный, отрезок, соединяющий точки  $x$  и  $y$ . В самом деле, вектор  $x - y$  является направляющим вектором прямой, проходящей через точки  $x$  и  $y$ , поэтому для любой точки  $z$  этой прямой справедливо представ-

ление  $z = y + \alpha(x - y)$ , которое отличается от (5.7) лишь формой записи. При  $\alpha = 0$  точка  $z = y$ , т.е. совпадает с одним из концов отрезка, а при  $\alpha = 1$  точка  $z = x$ , т.е. совпадает с другим концом. При изменении  $\alpha$  от 0 до 1 точка  $z$  пробегает отрезок от точки  $y$  до точки  $x$ .

**Определение 5.2.** Множество  $U \subset R^n$  называется *выпуклым*, если вместе с любыми точками  $x, y \in U$  оно содержит и весь отрезок (5.7).

Очевидно,  $R^n$  – выпуклое множество.

**Теорема 5.1.** Пересечение выпуклых множеств  $U_i, i = \overline{1, m}$ , есть выпуклое множество, если оно содержит более одной точки.

*Доказательство.* Пусть  $U = \bigcap_{i=1}^m U_i$ . Рассмотрим произвольные точки  $x, y \in U$ . Очевидно,  $x, y \in U_i$  при любом  $i = \overline{1, m}$  и, так как все  $U_i$  – выпуклы, отрезок  $[x, y] \subseteq U_i$ , для всех  $i$ . Следовательно, он целиком принадлежит и множеству  $U$ .

**Определение 5.3.** Функция  $f(x)$ , заданная на выпуклом множестве  $U \subset R^n$ , называется *выпуклой*, если для любых точек  $x, y \in U$  и любого  $\alpha \in [0; 1]$  выполняется неравенство

$$f[\alpha x + (1 - \alpha)y] \leq \alpha f(x) + (1 - \alpha)f(y). \quad (5.8)$$

Функция  $f(x)$  называется *строго выпуклой*, если для всех  $\alpha \in (0; 1)$  неравенство (5.8) выполняется как строгое.

**Теорема 5.2.** Линейная комбинация выпуклых на выпуклом множестве  $U$  функций  $f_i(x), i = \overline{1, m}$  с неотрицательными коэффициентами  $\lambda_i$ , т.е.  $f(x) = \sum_{i=1}^m \lambda_i f_i(x), \lambda_i \geq 0$ , есть выпуклая на множестве  $U$  функция.

*Доказательство.* При  $\lambda_i \geq 0$  функции  $\lambda_i f_i(x)$  выпуклы, поэтому для них выполняются неравенства (5.8). Складывая эти неравенства, получаем:

$$f[\alpha x + (1 - \alpha)y] \leq \sum_{i=1}^m \lambda_i f_i[\alpha x + (1 - \alpha)y] \leq$$

$$\begin{aligned} &\leq \alpha \sum_{i=1}^m \lambda_i f_i(x) + (1-\alpha) \sum_{i=1}^m \lambda_i f_i(y) = \\ &= \alpha f(x) + (1-\alpha) f(y); \quad x, y \in U, \quad \alpha \in [0; 1]. \end{aligned}$$

**Теорема 5.3.** Пусть  $g(x)$  – выпуклая функция, заданная в пространстве  $R^n$ . Тогда множество  $U$  точек  $x$ , удовлетворяющих неравенству  $g(x) \leq b$ , выпукло.

**Доказательство.** Пусть  $x, y \in U$  и  $z = \alpha x + (1-\alpha)y$ ,  $\alpha \in [0; 1]$ . Из выпуклости функции  $g(x)$  следует, что  $g(z) \leq \alpha g(x) + (1-\alpha)g(y)$  и, следовательно,  $g(z) \leq b$ , т.е. точка  $z \in U$ , множество  $U$  – выпукло.

**Следствие.** Пусть  $g_i(x)$ ,  $i = \overline{1, m}$  – выпуклые функции в  $R^n$ . Тогда множество точек  $x$ , удовлетворяющих системе неравенств  $g_i(x) \leq b_i$ ,  $i = \overline{1, m}$  выпукло.

Это следует из теорем 5.1 и 5.3.

Приведем свойства выпуклых функций, играющие важную роль в вопросах минимизации.

**Теорема 5.4.** Пусть  $f(x)$  – выпуклая на выпуклом множестве  $U$  функция. Тогда любой ее локальный минимум на множестве  $U$  является одновременно и глобальным.

**Доказательство.** Предположим противное, т.е. пусть  $x^{(0)}$  – точка локального, а  $x^*$  – точка глобального минимума  $f(x)$  на множестве  $U$ ,  $x^* \neq x^{(0)}$  и  $f(x^*) < f(x^{(0)})$ . Отсюда с учетом выпуклости функции имеем:

$$f[\alpha x^* + (1-\alpha)x^{(0)}] \leq \alpha f(x^*) + (1-\alpha) f(x^{(0)}) < f(x^{(0)}).$$

При  $\alpha \rightarrow +0$  точка  $x = \alpha x^* + (1-\alpha)x^{(0)}$  попадет в сколь угодно малую окрестность точки  $x^{(0)}$ . Поэтому полученное неравенство  $f(x) < f(x^{(0)})$  противоречит предположению о том, что  $x^{(0)}$  – точка локального минимума.

**Теорема 5.5.** Глобальный минимум строго выпуклой функции  $f(x)$  на выпуклом множестве  $U$  может достигаться лишь в единственной точке.

**Доказательство.** Предположим, что  $x^{(1)}$  и  $x^{(2)}$  – две различные точки глобального минимума. Из строгой выпуклости  $f(x)$  следует, что для всех  $\alpha \in (0; 1)$  выполняется строгое неравенство  $f[\alpha x^{(1)} + (1 - \alpha)x^{(2)}] < \alpha f(x^{(1)}) + (1 - \alpha)f(x^{(2)}) = f^* = \min_U f(x)$ ,

что противоречит предположению о том, что  $x^{(1)}$  и  $x^{(2)}$  – точки глобального минимума.

Применение многих методов минимизации обосновано только для функций, не имеющих точек локального минимума, отличных от глобального. Согласно теореме 5.4, таким свойством обладают выпуклые функции. Этим объясняется особая роль свойства выпуклости функции во многих вопросах оптимизации.

**З а м е ч а н и е.** Не всякая выпуклая функция  $f(x)$ ,  $x \in R^n$  достигает минимального значения, даже если она ограничена снизу. Например, функция  $f(x) = \exp(x)$  является выпуклой в пространстве  $R^1$ , но не достигает минимума ( $\inf f(x) = 0$ ).

Введем класс функций, для которых минимум в  $R^n$  обязательно существует.

**Определение 5.4.** Функция  $f(x)$ , заданная в  $R^n$ , называется *сильно выпуклой*, если существует такое число  $l > 0$  (*константа сильной*

*выпуклости*), что для всех  $x, y \in R^n$  и любого  $\alpha \in [0; 1]$  выполняется неравенство:

$$f[\alpha x + (1 - \alpha)y] \leq \alpha f(x) + (1 - \alpha)f(y) - \alpha(1 - \alpha)l \|x - y\|^2.$$

Очевидно, сильно выпуклая функция является выпуклой и строго выпуклой.

Можно показать, что у сильно выпуклой функции точка глобального минимума существует и единственна.

Следующие утверждения позволяют получить наглядное представление об особенностях графиков функций, обладающих свойством обычной, строгой и сильной выпуклости.

Для дифференцируемой функции  $f(x)$  при любых  $x, x^{(0)} \in R^n$

– *выпуклость* эквивалентна выполнению неравенства

$$f(x) \geq f(x^{(0)}) + \langle f'(x^{(0)}), x - x^{(0)} \rangle, \quad (5.9)$$

геометрически это означает, что график функции  $f(x)$  расположен не ниже касательной плоскости  $z = f(x^{(0)}) + \langle f'(x^{(0)}), x - x^{(0)} \rangle$ , проходящей через произвольную точку поверхности  $(x^{(0)}, f(x^{(0)}))$ ;

– *строгая выпуклость* эквивалентна строгому неравенству:

$$f(x) > f(x^{(0)}) + \langle f'(x^{(0)}), x - x^{(0)} \rangle, \quad (5.10)$$

график функции  $f(x)$  при этом имеет единственную точку с касательной плоскостью;

– *сильная выпуклость* определяется неравенством:

$$f(x) \geq f(x^{(0)}) + \langle f'(x^{(0)}), x - x^{(0)} \rangle + (l/2) \|x - x^{(0)}\|^2, \quad (5.11)$$

что геометрически означает расположение графика сильно выпуклой функции внутри некоторого параболоида вращения. Действительно, предположим, что  $f(x)$  сильно выпуклая функция и  $x^*$  – точка ее глобального минимума. Тогда  $f'(x^*) = 0$  и неравенство (5.11) принимает вид:  $f(x) \geq f(x^*) + (l/2) \|x - x^*\|^2$ . При этом поверхность  $z = f(x^*) + (l/2) \|x - x^*\|^2$  представляет собой параболоид вращения с вершиной в точке  $(x^*, f(x^*))$ .

*Замечание.* Из неравенства (5.9) для выпуклой дифференцируемой функции  $f(x)$  следует, что условие  $f'(x^*) = 0$  является не только необходимым, но и достаточным для того, чтобы  $x^*$  была точкой глобального минимума функции  $f(x)$ .

Приведем критерии строгой и сильной выпуклости для дважды дифференцируемых в  $R^n$  функций. Достаточным условием строгой выпуклости функции  $f(x)$  является положительная определенность при всех  $x \in R^n$  ее матрицы Гессе  $f''(x)$ , а сильной выпуклости – положительная определенность матрицы  $f''(x) - lE$ , где  $E$  – единичная матрица, а  $l > 0$ . Эти критерии в комбинации с критерием Сильвестра (5.1) составляют во многих случаях удобный аппарат для проверки выпуклости функций небольшого числа переменных.

### **Выпуклые квадратичные функции**

Важную роль в ряде вопросов минимизации играют квадратичные функции, которые в  $n$ -мерном случае являются обобщением

квадратного трехчлена одной переменной  $f(x) = \frac{1}{2}ax^2 + bx + c$ .

**Определение 5.5.** Функция вида

$$f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n \alpha_{ij} x_i x_j + \sum_{j=1}^n b_j x_j + c, \quad (5.12)$$

называется *квадратичной функцией  $n$  переменных*.

Положив  $a_{ij} = \alpha_{ij} + \alpha_{ji}$ , получим симметрическую матрицу  $A = (a_{ij})$ , с помощью которой выражение (5.12) можно записать в другой форме:

$$f(\mathbf{x}) = \frac{1}{2} \langle A \mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{b}, \mathbf{x} \rangle + c, \quad (5.13)$$

где  $\mathbf{b} = (b_1, \dots, b_n) \in R^n$  вектор коэффициентов  $b_j$ .

**Пример 5.1.** Функция

$$f(\mathbf{x}) = 2x_1^2 + x_2^2 + 4x_3^2 - 2x_1x_2 + 3x_1x_3 - 2x_2x_3 + x_1 - x_2 + 3x_3 + 5$$

является квадратичной. Запишем ее матрицу  $A$ , вектор  $\mathbf{b}$  и коэффициент  $c$  в соответствии с (5.13):

$$A = \begin{pmatrix} 4 & -2 & 3 \\ -2 & 2 & -2 \\ 3 & -2 & 8 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}, \quad c = 5.$$

Перечислим *основные свойства квадратичных функций*:

1. Для градиента квадратичной функции (5.13) справедлива формула:

$$\mathbf{f}'(\mathbf{x}) = A \mathbf{x} + \mathbf{b}.$$

Запишем, например,  $k$ -ю координату вектора  $\mathbf{f}'(\mathbf{x})$ :

$$\begin{aligned} \frac{\partial f}{\partial x_k} &= \frac{\partial}{\partial x_k} \left( \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \sum_{j=1}^n b_j x_j \right) = \\ &= \frac{1}{2} \sum_{i=1}^n (a_{ik} + a_{ki}) x_i + b_k = \sum_{i=1}^n a_{ki} x_i + b_k. \end{aligned}$$

2. Гессиан квадратичной функции (5.13) совпадает с матрицей  $A$ :

$$\mathbf{f}''(\mathbf{x}) = A.$$

Вычислим элемент матрицы Гессе:

$$\frac{\partial^2 f(\mathbf{x})}{\partial x_l \partial x_k} = \frac{\partial}{\partial x_l} \left( \frac{\partial f}{\partial x_k} \right) = \frac{\partial}{\partial x_l} \left( \sum_{i=1}^n a_{ki} x_i + b_k \right) = a_{kl}.$$

3. Квадратичная функция (5.13) с положительно определенной матрицей  $A$  сильно выпукла.

Так как матрица  $f''(x) = A$  симметрична и положительно определена, то все ее собственные значения  $\lambda_i$ , положительны и существует ортонормированный базис из собственных векторов этой матрицы (см. разд. 5.1). В этом базисе

$$A = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix}, \quad A - lE = \begin{pmatrix} \lambda_1 - l & 0 & \dots & 0 \\ 0 & \lambda_2 - l & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_n - l \end{pmatrix}.$$

Поэтому угловые миноры матрицы  $A - lE$  равны

$\Delta_k = \prod_{i=1}^k (\lambda_i - l)$  и положительны при  $0 < l < \min_i \lambda_i$ . Таким образом, существует число  $l > 0$  при котором матрица  $A - lE$  положительно определена. А это означает, что  $f(x)$  сильно выпукла.

Для квадратичной функции  $f(x)$  из (5.13) с положительно определенной матрицей  $A$  точка глобального минимума существует и единственна, так как  $f(x)$  сильно выпукла.

**Пример 5.2.** Квадратичная функция  $f(x)$  из примера 5.1 сильно выпукла.

Матрица  $f''(x) = A$  положительно определена, так как

$$\Delta_1 = 4 > 0; \quad \Delta_2 = \begin{vmatrix} 4 & -2 \\ -2 & 2 \end{vmatrix} = 4 > 0; \quad \Delta_3 = \begin{vmatrix} 4 & -2 & 3 \\ -2 & 2 & -2 \\ 3 & -2 & 8 \end{vmatrix} = 22 > 0.$$

Следовательно,  $f(x)$  сильно выпукла по свойству 3 квадратичных функций.

Выпуклые квадратичные функции играют важную роль в теории  $n$ -мерной оптимизации. Так, некоторые алгоритмы, разработанные с учетом свойств таких функций, позволяют найти их точку минимума за конечное число итераций. Во многих случаях эти алгоритмы оказываются эффективными и для неквадратичных выпуклых функций, так как в достаточно малой окрестности точки минимума  $x^*$  дважды дифференцируемая функция  $f(x)$  с положительно определенной

матрицей Гессе  $f''(x)$  хорошо аппроксимируется сильно выпуклой квадратичной функцией  $\bar{f}(x) = f(x^*) + \frac{1}{2} \langle f''(x^*) \cdot (x - x^*), x - x^* \rangle$ , где  $f'(x^*) = 0$ .

### 5.3. Общие принципы многомерной оптимизации

Для численного решения задач безусловной минимизации:  $f(x) \rightarrow \min, x \in R^n$  разработано много алгоритмов, использующих итерационные процедуры вида

$$x^{(k+1)} = \Phi(x^{(0)}, x^{(1)}, \dots, x^{(k-1)}, x^{(k)}), \quad x^{(0)} \in R^n \quad (5.14)$$

позволяющие при определенных условиях построить последовательность  $\{x^{(k)}\}$  такую, что

$$\lim_{k \rightarrow \infty} f(x^{(k)}) = \begin{cases} f^* = \min_{R^n} f(x), & \text{если } U^* \neq \emptyset, \\ f_* = \inf_{R^n} f(x), & \text{если } U^* = \emptyset, \end{cases} \quad (5.15)$$

где  $U^*$  – множество точек глобального минимума функции  $f(x)$ . Последовательность  $\{x^{(k)}\}$ , удовлетворяющая требованию (5.15), называется *минимизирующей* для функции  $f(x)$ . Если, кроме того, для случая  $U^* \neq \emptyset$  дополнительно выполняется условие

$$\lim_{k \rightarrow \infty} \rho(x^{(k)}, U^*) = \lim_{k \rightarrow \infty} \inf_{y \in U^*} \rho(x^{(k)}, y) = 0,$$

то говорят, что минимизирующая последовательность *сходится к множеству*  $U^*$ . Если множество  $U^*$  состоит из единственной точки  $x^*$ , то для сходящейся к  $U^*$  минимизирующей последовательности  $\lim_{k \rightarrow \infty} x^{(k)} = x^*$ .

Минимизирующая последовательность может и не сходиться к точке минимума. Например, для  $f(x) = x^2/(1+x^4)$ ,  $x \in R^1$  последовательность  $x^k = k$  является минимизирующей, но не сходится к единственной точке минимума  $x^* = 0$ .

Вопрос о существовании точки минимума обычно решается с помощью теоремы Вейерштрасса, которая гласит: если функция  $f(x)$  непрерывна в  $R^n$  и множество  $U_\alpha = \{x \mid f(x) \leq \alpha\}$  для некоторого

$\alpha$  непусто и ограничено, то  $f(x)$  достигает глобального минимума в  $R^n$ .

Отметим, что если множество  $U_\alpha$  – выпукло, а функция  $f(x)$  строго выпукла, то точка ее минимума единственна (см. теорему 5.5).

Среди итерационных процедур (5.14) можно условно выделить такие, которые гарантируют отыскание решения задачи за конечное число итераций (шагов). Однако их удается построить лишь для некоторых специальных типов задач минимизации. Как правило, приходится иметь дело с бесконечными последовательностями  $\{x^{(k)}\}$ , поэтому говорить о достижении решения можно лишь в пределе.

Важной характеристикой сходящихся минимизирующих последовательностей является *скорость сходимости*.

Последовательность  $\{x^{(k)}\}$  сходится к точке  $x^*$  *линейно* (со скоростью геометрической прогрессии), если существует такое число  $q \in (0; 1)$ , что выполняется неравенство  $\rho(x^{(k)}, x^*) \leq q\rho(x^{(k-1)}, x^*)$ , т.е.  $\rho(x^{(k)}, x^*) \leq q^k \rho(x^{(0)}, x^*)$ .

Сходимость называют *сверхлинейной* (т.е. более быстрой, чем определяемая любой геометрической прогрессией), если  $\rho(x^{(k)}, x^*) \leq q_k \rho(x^{(k-1)}, x^*)$ ,  $q_k \rightarrow +0$  при  $k \rightarrow \infty$ .

Наконец, термин *квадратичная* сходимость используется, если справедлива оценка  $\rho(x^{(k)}, x^*) \leq [c\rho(x^{(k-1)}, x^*)]^2$ ,  $c > 0$  или  $\rho(x^{(k)}, x^*) \leq q^{2^k}$ , где  $q = c\rho(x^{(0)}, x^*)$ .

Для многих алгоритмов скорость сходимости последовательности  $\{x^{(k)}\}$  из (5.14) характеризуется и другими неравенствами, например,  $\rho(x^{(k)}, x^*) \leq c/k^\alpha$ ,  $c > 0$ ,  $\alpha > 0$ .

Установление факта сходимости последовательности  $\{x^{(k)}\}$  из (5.14) и оценка скорости сходимости дают существенную информацию об итерационном процессе (5.14).

Конкретный вычислительный алгоритм на основе (5.14), в котором может получаться, вообще говоря, бесконечная последовательность  $\{x^{(k)}\}$ , необходимо дополнять *условием остановки* вычислений (критерием окончания счета). На практике часто пользуются следующими условиями:

$$\rho(x^{(k+1)}, x^{(k)}) < \varepsilon_1,$$

$$|f(x^{(k+1)}) - f(x^{(k)})| < \varepsilon_2,$$

$$\|f'(x^{(k)})\| < \varepsilon_3,$$

где  $\varepsilon_i$  – заранее заданные параметры точности.

Ниже будут рассмотрены вычислительные алгоритмы простейших процедур (5.14), как правило, основанные на рекуррентных формулах вида

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad k = 0, 1, \dots, \quad (5.16)$$

где  $p^{(k)}$  – направление поиска точки  $x^{(k+1)}$  из точки  $x^{(k)}$ , а число  $\alpha_k$  – величина шага, которая выбирается так, чтобы выполнялось условие  $f(x^{(k+1)}) < f(x^{(k)})$ .

Эти алгоритмы различаются способом построения вектора  $p^{(k)}$  и выбора шага  $\alpha_k$ .

Будем говорить, что в итерационном процессе (5.16) производится *исчерпывающий спуск*, если величина шага  $\alpha_k$  находится из решения одномерной задачи минимизации

$$\Phi_k(\alpha) \rightarrow \min, \quad \Phi_k(\alpha) = f(x^{(k)} + \alpha_k p^{(k)}). \quad (5.17)$$

Таким образом, при исчерпывающем спуске на каждом шаге полностью реализуется возможность уменьшить значение целевой функции  $f(x)$  при перемещении из точки  $x^{(k)}$  в направлении, коллинеарном вектору  $p^{(k)}$ . Величина шага  $\alpha_k$  может быть найдена, например, с помощью некоторого метода одномерного поиска (гл. 4).

В дальнейшем будет часто использоваться следующее свойство исчерпывающего спуска.

**Теорема 5.6.** Для дифференцируемой в  $R^n$  функции  $f(x)$  в итерационном процессе (5.16) с выбором шага  $\alpha_k$  в соответствии с (5.17) для всех  $k \geq 1$  выполняется условие

$$\langle f'(x^{(k+1)}), p^{(k)} \rangle = 0. \quad (5.18)$$

**Доказательство.** Запишем необходимое условие минимума функции одной переменной  $\Phi_k(\alpha)$  из (5.17), используя правило дифференцирования сложной функции:

$$\frac{d\Phi_k(\alpha)}{d\alpha} = \sum_{j=1}^n \frac{\partial f(x^{(k+1)})}{\partial x_j^{(k+1)}} \frac{dx_j^{(k+1)}}{d\alpha} = 0.$$

Учитывая, что  $x_j^{(k+1)} = x_j^{(k)} + \alpha p_j^{(k)}$  получаем условие (5.18).

Дадим геометрическую иллюстрацию соотношения (5.18) в пространстве  $R^n$ . При перемещении из точки  $x^{(k)}$  вдоль прямой, задаваемой вектором  $p^{(k)}$ , в направлении убывания функции, происходит пересечение линий уровня функции  $f(x)$  до тех пор, пока либо не будет достигнута стационарная точка ( $f'(x^{(k+1)}) = 0$ ), либо прямая не коснется в точке  $x^{(k+1)}$  некоторой линии уровня функции  $f(x)$ . Равенство (5.18) и есть условие касания (рис. 5.1).

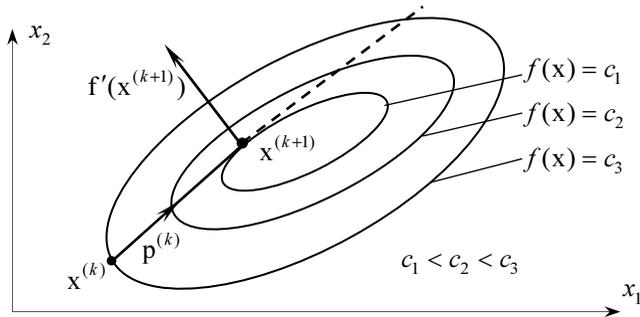


Рис. 5.1

Свойство (5.18) позволяет в явном виде найти величину  $\alpha_k$  для квадратичной функции.

**Теорема 5.7.** Для квадратичной функции  $f(x) = \frac{1}{2}\langle Ax, x \rangle + \langle b, x \rangle + c$  величина  $\alpha_k$  исчерпывающего спуска в итерационном процессе (5.16) будет

$$\alpha_k = -\frac{\langle f'(x^{(k)}), p^{(k)} \rangle}{\langle Ap^{(k)}, p^{(k)} \rangle} = -\frac{\langle Ax^{(k)} + b, p^{(k)} \rangle}{\langle Ap^{(k)}, p^{(k)} \rangle}. \quad (5.19)$$

**Доказательство.** Умножив равенство (5.16) слева на матрицу  $A$  квадратичной функции  $f(x)$  и прибавив к обеим частям вектор  $b$ , получим:  $Ax^{(k+1)} + b = Ax^{(k)} + b + \alpha_k Ap^{(k)}$ . Учитывая, что градиент квадратичной функции равен  $f'(x) = Ax + b$ , имеем:  $f'(x^{(k+1)}) = f'(x^{(k)}) + \alpha_k Ap^{(k)}$ . Подставляя выражение для  $f'(x^{(k+1)})$  в равенство (5.18), получаем формулу (5.19).

**Определение 5.6.** Направление вектора  $\mathbf{p}^{(k)}$  называется *направлением убывания* функции  $f(\mathbf{x})$  в точке  $\mathbf{x}^{(k)}$ , если при всех достаточно малых положительных  $\alpha$  выполняется неравенство

$$f(\mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)}) < f(\mathbf{x}^{(k)}).$$

В итерационном процессе (5.16) используются, как правило, направления убывания. Сформулируем признак направления убывания.

**Теорема 5.8.** Пусть функция  $f(\mathbf{x})$  дифференцируема в точке  $\mathbf{x}^{(k)}$ . Если вектор  $\mathbf{p}^{(k)}$  удовлетворяет условию

$$\langle \mathbf{f}'(\mathbf{x}^{(k)}), \mathbf{p}^{(k)} \rangle < 0, \quad (5.20)$$

то направление вектора  $\mathbf{p}^{(k)}$  является направлением убывания.

**Доказательство.** Из свойства дифференцируемой функции (5.2) и условия (5.20) следует, что  $f(\mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)}) - f(\mathbf{x}^{(k)}) = \langle \mathbf{f}'(\mathbf{x}^{(k)}), \alpha \mathbf{p}^{(k)} \rangle + o(\|\Delta \mathbf{x}\|) = \alpha \left( \langle \mathbf{f}'(\mathbf{x}^{(k)}), \mathbf{p}^{(k)} \rangle + \frac{o(\alpha)}{\alpha} \right) < 0$  при всех достаточно малых  $\alpha > 0$ , т.е. вектор  $\mathbf{p}^{(k)}$  задает направление убывания функции  $f(\mathbf{x})$  в точке  $\mathbf{x}^{(k)}$ .

Геометрически условие (5.20) означает, что вектор  $\mathbf{p}^{(k)}$  составляет тупой угол с градиентом  $\mathbf{f}'(\mathbf{x}^{(k)})$ .

## 5.4. Прямые методы безусловной оптимизации

Рассмотрим конкретные алгоритмы решения задачи безусловной минимизации  $f(\mathbf{x}) \rightarrow \min, \mathbf{x} \in R^n$ , которые опираются только на вычисление значений функции  $f(\mathbf{x})$ , т.е. *прямые методы минимизации*. Важно отметить, что для их применения не требуется не только дифференцируемости целевой функции, но даже аналитического ее задания. Нужно лишь иметь возможность вычислять или измерять значения  $f(\mathbf{x})$  в произвольных точках. Такие ситуации часто встречаются в практически важных задачах оптимизации.

Остановимся сначала на вычислительных процедурах вида (5.14), в которых выбор нового приближения к точке минимума определяется сравнением значений функции в нескольких точках пространства  $R^n$ .

## Метод минимизации по правильному симплексу

**Определение 5.7.** *Правильным симплексом* в пространстве  $R^n$  называется множество из  $n+1$  равноудаленных друг от друга точек (*вершин симплекса*). Отрезок, соединяющий две вершины, называется *ребром симплекса*.

В пространстве  $R^2$  правильным симплексом является совокупность вершин равностороннего треугольника, в  $R^3$  – правильного тетраэдра.

Если  $x^{(0)}$  – одна из вершин правильного симплекса в  $R^n$ , то координаты остальных  $n$  вершин  $x^{(1)}, \dots, x^{(n)}$  можно найти, например, по формулам:

$$x_i^{(j)} = \begin{cases} x_j^{(0)} + d_1, & i \neq j, \\ x_j^{(0)} + d_2, & i = j, \end{cases} \quad (5.21)$$

где  $d_1 = a(\sqrt{n+1}-1)/(n\sqrt{2})$ ,  $d_2 = a(\sqrt{n+1}+n-1)/(n\sqrt{2})$ ,  $a$  – длина ребра. Вершина  $x^{(0)}$  симплекса, построенного по формулам (5.21), называется *базовой*.

В алгоритме этого метода используется следующее важное свойство правильного симплекса. По известному симплексу можно построить новый симплекс путем *отражения* какой-либо вершины, например,  $x^{(k)}$  симметрично относительно центра тяжести  $x^c$  остальных вершин симплекса. Новая и старая вершины  $\hat{x}^{(k)}$  и  $x^{(k)}$  связаны соотношением:  $\frac{\hat{x}^{(k)} + x^{(k)}}{2} = x^c$ , где  $x^c = \frac{1}{n} \sum_{i=1, i \neq k}^n x^{(i)}$ . В результате

получается новый правильный симплекс с тем же ребром и вершинами  $\hat{x}^{(k)} = 2x^c - x^{(k)}$ ,  $x^{(i)}$ ,  $i = 0, \dots, n$ ,  $i \neq k$ . Таким образом, происходит перемещение симплекса в пространстве  $R^n$ . На рис. 5.2 представлена иллюстрация этого свойства симплекса в пространстве  $R^2$ . На рис. 5.2,а изображен начальный симплекс  $x^{(0)}, x^{(1)}, x^{(2)}$ . На рис. 5.2,б – новый симплекс  $x^{(0)}, x^{(1)}, \hat{x}^{(2)}$  с центром отражения в точке  $x^c = (x^{(0)} + x^{(1)})/2$ .

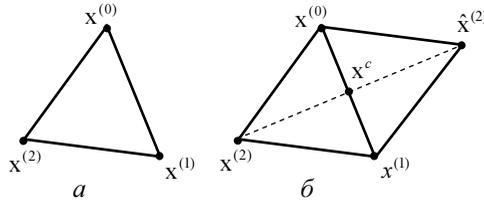


Рис. 5.2

Поиск точки минимума функции  $f(x)$  с помощью правильных симплексов производится следующим образом. На каждой итерации сравниваются значения  $f(x)$  в вершинах симплекса. Затем проводится описанная выше процедура отражения для той вершины, в которой  $f(x)$  принимает наибольшее значение. Если в отраженной вершине получается меньшее значение функции, то переходят к новому симплексу. В противном случае выполняют еще одну попытку отражения для вершины со следующим по величине значением  $f(x)$ . Если и она не приводит к уменьшению функции, то сокращают длину ребра симплекса, например, вдвое и строят новый симплекс с этим ребром. В качестве базовой вершины выбирают ту вершину  $x^{(0)}$  старого симплекса, в которой функция принимает минимальное значение. Поиск точки минимума  $f(x)$  заканчивают, когда либо ребро симплекса, либо разность между значениями функции в вершинах симплекса становятся достаточно малыми.

**Алгоритм метода минимизации по правильному симплексу**

1. Выбрать параметр точности  $\varepsilon$ , базовую точку  $x^{(0)}$ , ребро  $a$  и построить начальный симплекс по формулам (5.21). Вычислить  $f(x^{(0)})$ .
2. Вычислить значения  $f(x)$  в вершинах симплекса  $x^{(1)}, \dots, x^{(n)}$ .
3. Упорядочить вершины симплекса  $x^{(0)}, x^{(1)}, \dots, x^{(n)}$ , так чтобы  $f(x^{(0)}) \leq f(x^{(1)}) \leq \dots \leq f(x^{(n-1)}) \leq f(x^{(n)})$ .
4. Проверить условие (это одно из возможных условий остановки вычислительного процесса)

$$\frac{1}{n} \sum_{i=1}^n [f(x^{(i)}) - f(x^{(0)})]^2 < \varepsilon^2.$$

Если оно выполнено, то вычисления прекратить, полагая  $x^* = x^{(0)}$ ,  $f^* = f(x^{(0)})$ . В противном случае перейти к шагу 5.

5. Найти  $x^c = \frac{1}{n} \sum_{i=1, i \neq k}^n x^{(i)}$  и выполнить отражение вершины

$x^{(n)}$ :  $\hat{x}^{(n)} = 2x^c - x^{(n)}$ . Если  $f(\hat{x}^{(n)}) < f(x^{(n)})$ , то положить  $x^{(n)} = \hat{x}^{(n)}$  и перейти к шагу 3. Иначе – перейти к шагу 6.

6. Найти  $x^c = \frac{1}{n} \sum_{i=1, i \neq n-1}^n x^{(i)}$  и выполнить отражение вершины

$x^{(n-1)}$ :  $\hat{x}^{(n-1)} = 2x^c - x^{(n-1)}$ . Если  $f(\hat{x}^{(n-1)}) < f(x^{(n-1)})$ , то положить  $x^{(n-1)} = \hat{x}^{(n-1)}$  и перейти к шагу 3. Иначе – перейти к шагу 7.

7. Перейти к новому правильному симплексу с вдвое меньшим ребром, считая базовой вершиной  $x^{(0)}$ . Остальные  $n$  вершин симплекса найти по формуле  $x^{(i)} = (x^{(i)} + x^{(0)})/2$ ,  $i = \overline{1, n}$ . Перейти к шагу 2.

#### З а м е ч а н и я

1. Следует иметь в виду, что если функция  $f(x)$  не унимодальна, то описанным методом может быть найдена точка локального, а не глобального минимума  $f(x)$ .

2. Если ограниченность снизу целевой функции не очевидна, то в алгоритм метода следует включить дополнительную процедуру останковки вычислений.

#### Метод минимизации по деформируемому симплексу

Алгоритм минимизации по правильному симплексу можно модифицировать, добавив к процедуре отражения при построении нового симплекса процедуры *сжатия* и *растяжения*. А именно, положение новой вершины  $\hat{x}^{(n)}$  вместо вершины  $x^{(n)}$ , соответствующей наибольшему значению функции, находится сравнением и выбором наименьшего среди значений целевой функции в точках:

$$\begin{aligned} z^{(1)} &= x^c - \alpha(x^c - x^{(n)}), \quad 0 < \alpha < 1; \\ z^{(2)} &= x^c + \alpha(x^c - x^{(n)}), \quad 0 < \alpha < 1; \\ z^{(3)} &= x^c + \beta(x^c - x^{(n)}), \quad \beta \approx 1; \\ z^{(4)} &= x^c + \gamma(x^c - x^{(n)}), \quad \gamma > 1. \end{aligned} \tag{5.22}$$

Геометрическая иллюстрация этих процедур для пространства  $R^2$  приведена на рис. 5.3.

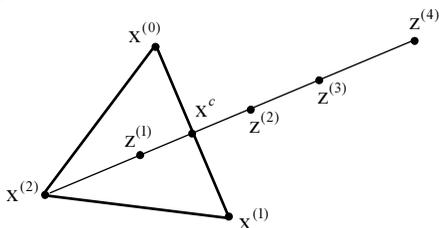


Рис. 5.3

Так как величина  $\alpha \in (0; 1)$ , то выбор точек  $z^{(1)}$  и  $z^{(2)}$  соответствует сжатию симплекса;  $\beta \approx 1$ , поэтому выбор точки  $z^{(3)}$  соответствует отражению, а  $\gamma > 1$  и выбор точки  $z^{(4)}$  приводит к растяжению симплекса. Численные эксперименты показывают, что этот алгоритм хорошо работает в пространстве  $R^n$  для  $n \leq 6$ . Отметим, что при деформациях утрачивается свойство правильности исходного симплекса. Поэтому, не стремясь к правильности начального симплекса, его строят из произвольной базовой точки  $x^{(0)} \in R^n$  по формуле

$$x^{(i)} = x^{(0)} + a e^{(i)}, \quad (5.23)$$

где  $e^{(i)}$  – единичный  $i$ -й базисный вектор;  $a$  – параметр симплекса.

На практике хорошо зарекомендовал себя следующий набор параметров  $\alpha, \beta, \gamma$  для выбора пробных точек  $z^{(i)}$  в формулах (5.22):  $\alpha = 0.5, \beta = 1, \gamma = 2$ .

*Алгоритм метода минимизации по деформируемому симплексу*

1. Выбрать параметр точности  $\varepsilon$ , параметры  $\alpha, \beta$  и  $\gamma$  базовую точку  $x^{(0)} \in R^n$ , параметр  $a$  и построить начальный симплекс по формулам (5.21) или (5.23). Вычислить  $f(x^{(0)})$ .

2. Вычислить значения функции в вершинах симплекса  $x^{(1)}, \dots, x^{(n)}$ .

3. Упорядочить вершины  $x^{(0)}, x^{(1)}, \dots, x^{(n)}$  так, чтобы  $f(x^{(0)}) \leq f(x^{(1)}) \leq \dots \leq f(x^{(n)})$ .

4. Проверить достижение заданной точности

$$\frac{1}{n} \sum_{i=1}^n [f(x^{(i)}) - f(x^{(0)})]^2 < \varepsilon^2.$$

Если оно выполняется, то вычисления завершить, полагая  $x^* = x^{(0)}$ ,  $f^* = f(x^{(0)})$ . Иначе – перейти к шагу 5.

5. Найти  $x^c = \frac{1}{n} \sum_{i=0}^{n-1} x^{(i)}$  и пробные точки  $z^{(k)}$ ,  $k=1, \dots, 4$  по формулам (5.22). Найти  $f(z^*) = \min_k f(x^{(k)})$ . Если  $f(z^*) < f(x^{(n)})$ , то положить  $x^{(n)} = z^*$  и перейти к шагу 3. Иначе – перейти к шагу 6.

6. Уменьшить симплекс, полагая  $x^{(i)} = (x^{(i)} + x^{(0)})/2$ ,  $i=1, \dots, n$  и перейти к шагу 2.

*З а м е ч а н и е.* Для того чтобы избежать сильной деформации симплекса, алгоритм иногда дополняют процедурой обновления. Например, после  $N$  шагов алгоритма из точки  $x^{(0)}$  снова строят симплекс по формулам (5.21) или (5.23), полагая  $a = \|x^{(0)} - x^{(n)}\|$

С теоретической точки зрения описанные методы минимизации слабо исследованы, однако практика подтверждает их работоспособность.

### Метод циклического покоординатного поиска

Этот метод заключается в последовательной минимизации целевой функции  $f(x)$  сначала по направлению первого базисного вектора  $e^{(1)}$ , затем второго –  $e^{(2)}$  и т.д. После окончания минимизации по направлению последнего базисного вектора  $e^{(n)}$  цикл повторяется.

#### Алгоритм метода циклического покоординатного поиска

1. Выбрать  $x^{(0)} \in R^n$ ,  $\varepsilon$  и критерий достижения точности:

$$\|x - \hat{x}\| < \varepsilon \quad \text{или} \quad |f(x) - f(\hat{x})| < \varepsilon.$$

Найти  $f(x)$ , положить  $j=1$ .

2. Решить задачу  $\Phi(\alpha) = f(x + \alpha e^{(j)}) \rightarrow \min$ ,  $\alpha \in R$  одним из методов одномерной минимизации, т.е. найти  $\alpha^*$ . Положить  $\hat{x} = x + \alpha^* e^{(j)}$ , вычислить  $f(\hat{x})$ .

3. Если  $j < n$ , то положить  $x = \hat{x}$ ,  $j = j + 1$  и перейти к шагу 2, иначе – перейти к шагу 4.

4. Проверить условие достижения точности  $\|x - \hat{x}\| < \varepsilon$  или  $|f(x) - f(\hat{x})| < \varepsilon$ . Если оно выполняется, то положить  $x^* = \hat{x}$ ,  $f^* = f(\hat{x})$  и закончить поиск. Иначе – положить  $x = \hat{x}$ ,  $f(x) = f(\hat{x})$  и перейти к шагу 2.

Эффективность метода циклического покоординатного спуска существенно зависит от свойств целевой функции.

Скорость сходимости к решению задачи методом циклического покоординатного поиска можно повысить, если дополнить его алгоритм периодически повторяющимся поиском точки минимума в направлениях  $p^{(i)} = x^{(i+n)} - x^{(i)}$  из точек  $x^{(i)} \in R^n$ . Такой подход, состоящий в последовательном нахождении направлений убывания функции и минимизация ее по этим направлениям, лежит в основе ряда алгоритмов [14].

Проиллюстрируем процесс поиска точки минимума на следующих примерах.

### Пример 5.3

Решить задачу  $f(x) = (x_1 - 2)^2 + (x_2 - 3)^2 \rightarrow \min$ ,  $x \in R^2$  методом циклического покоординатного спуска.

**Решение.** Линии уровня этой целевой функции – окружности с центром в точке  $x = (2, 3)$  (рис. 5.4). Выберем произвольную начальную точку  $x$ , например  $x = (1, 1)$ . Очевидно, два шага исчерпывающего спуска сначала по направлению  $e^{(1)}$ , затем –  $e^{(2)}$  приведут в точку минимума  $x^* = (2, 3)$ .

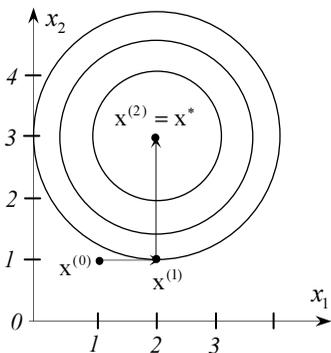


Рис. 5.4

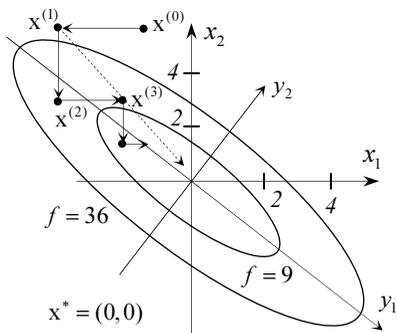


Рис. 5.5

В этом примере точку минимума функции удалось найти точно за конечное число шагов. Это скорее исключение, чем правило.

#### Пример 5.4

Решить задачу  $f(x) = 5x_1^2 + 5x_2^2 + 8x_1x_2 \rightarrow \min$ ,  $x \in R^2$  методом циклического покоординатного спуска.

Решение. Сначала заметим, что замена переменных  $x_1 = (y_1 + y_2)/\sqrt{2}$  и  $x_2 = (-y_1 + y_2)/\sqrt{2}$  приводит к повороту системы координат на угол  $-45^\circ$  и функцию – к виду

$$f(y) = y_1^2 + 9y_2^2 \rightarrow \min.$$

Очевидно, линии уровня целевой функции – эллипсы (рис. 5.5).

Если принять начальную точку  $x^{(0)} = (-2, 5)$ ,  $f(x^{(0)}) = 65$ , то в результате расчетов по приведенному выше алгоритму, например, на 10-й итерации получим:  $x^{(10)} = (-0.67, 0.54)$ ,  $f(x^{(10)}) = 0.81$ . Вычисления показывают, что минимизирующая последовательность  $\{x^{(k)}\}$  сходится к точке минимума  $x^* = (0, 0)$ . Однако, в отличие от решения задачи из примера 5.3, достижение точки минимума за конечное число шагов не гарантируется. Траектория поиска точки минимума в данной задаче имеет ярко выраженный зигзагообразный характер.

Из примера 5.4 видно, что эффективность решения задачи методом циклического покоординатного спуска можно повысить. Для этого достаточно дополнить его алгоритм периодически повторяющимся поиском точки минимума в направлениях  $p^{(i)} = x^{(i)} - x^{(i-2)}$  из точек  $x^{(i)}$ . Так, например, если из точки  $x^{(3)}$  провести исчерпывающий поиск в направлении  $p^{(3)} = x^{(3)} - x^{(1)}$  (на рис. 5.4 направление вектора  $p^{(3)}$  показано пунктиром), то получим точку  $(-2.2 \cdot 10^{-5}, 5.6 \cdot 10^{-3})$ , расположенную значительно ближе к точке минимума, чем точка после сотни итераций.

Такой подход, состоящий в последовательном нахождении направлений убывания функции и минимизация ее по этим направлениям, лежит в основе ряда алгоритмов [14].

Описанные выше прямые методы безусловной минимизации функций многих переменных содержат детерминированные процедуры поиска точек с меньшим значением функции.

Однако разработано довольно много методов минимизации, в которых в процедуру поиска точек минимума намеренно вводится элемент случайности. Эти методы объединены под общим названием –

методы случайного поиска [3, 14]. Основой для этих методов служит итерационный процесс

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k (\xi / \|\xi\|), \quad k = 0, 1, \dots,$$

где  $\alpha_k$  – величина шага,  $\xi = (\xi_1, \dots, \xi_n)$  – некоторая реализация  $n$ -мерного случайного вектора  $\xi$ .

В этой книге методы случайного поиска не рассматриваются.

### Метод сопряженных направлений

Описанные прямые методы минимизации требуют, вообще говоря, бесконечного числа итераций для точного определения точки минимума целевой функции. Это относится и к сильно выпуклым квадратичным функциям, вопросы минимизации которых хорошо изучены.

Однако существуют прямые итерационные методы, приводящие к точке минимума сильно выпуклой квадратичной функции за конечное число шагов. Эти методы обладают высокой эффективностью и в случае выпуклой неквадратичной функции. Они основаны на вычислительной процедуре вида (5.16):  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$ . В зависимости от способа выбора направления  $\mathbf{p}^{(k)}$  и шага  $\alpha_k$  получаются различные алгоритмы минимизации.

Опишем один из них – *метод сопряженных направлений*.

Рассмотрим сначала проблему поиска точки минимума сильно выпуклой квадратичной функции двух переменных. Ее линиями уровня являются эллипсы (рис. 5.6).

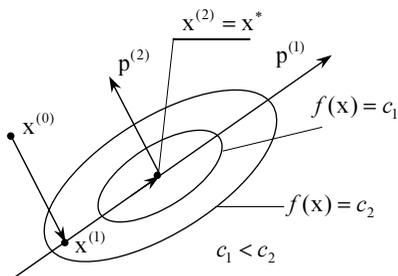


Рис. 5.6

Пусть  $\mathbf{p}^{(1)}$  и  $\mathbf{p}^{(2)}$  – направления главных осей этих эллипсов (они могут быть найдены как ортонормированный базис из собственных векторов матрицы  $A$  квадратичной функции). Если из произвольной точки  $\mathbf{x}^{(0)} \in R^2$  выполнить итерационную процедуру  $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \alpha_k \mathbf{p}^{(k)}$ ,  $k = 1, 2, \dots$ , где величина шага  $\alpha_k$  находится из

условия исчерпывающего спуска, то, очевидно, потребуется не более двух шагов для отыскания точки  $x^*$ .

Такого же результата можно достичь и другим способом. Выберем некоторое направление  $p^{(1)}$  и две точки  $x^{(0)}$  и  $y^{(0)}$  такие, чтобы векторы  $x^{(0)} - y^{(0)}$  и  $p^{(1)}$  были неколлинеарны (рис. 5.7).

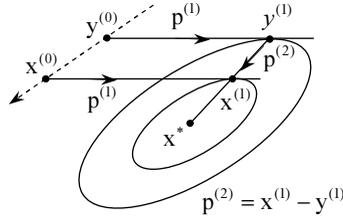


Рис. 5.7

Выполнив исчерпывающий спуск из точек  $x^{(0)}$  и  $y^{(0)}$  в направлении  $p^{(1)}$ , получим точки  $x^{(1)}$  и  $y^{(1)}$ . По свойству исчерпывающего спуска в точках  $x^{(1)}$  и  $y^{(1)}$  имеет место касание соответствующих прямых (направлений убывания) и эллипсов (линий уровня целевой функции). Так как эллипсы различаются гомотетией с центром в точке  $x^*$ , то точки  $x^*$ ,  $x^{(1)}$  и  $y^{(1)}$  расположены на одной прямой. Поэтому, полагая  $p^{(2)} = x^{(1)} - y^{(1)}$  и решая задачу  $f(x^{(1)} + \alpha p^{(2)}) \rightarrow \min$ , мы находим точку  $x^*$ . Таким образом, и в этом случае решение задачи минимизации квадратичной сильно выпуклой функции будет получено за конечное число шагов.

Рассмотренному способу минимизации квадратичных функций *двух переменных* соответствует, например, такой **алгоритм**:

1. Выбрать начальную точку  $x^{(0)} \in R^2$ .
2. Положить  $p^{(1)} = e^{(1)}$ . Найти  $x^{(1)}$  с помощью исчерпывающего спуска из точки  $x^{(0)}$  по направлению  $p^{(1)}$ :  $f(x^{(1)}) = \min_{\alpha \in R} f(x^{(0)} + \alpha p^{(1)})$ .
3. Основная итерация
  - 3.1. Положить  $y^{(0)} = x^{(1)} + e^{(2)}$ .
  - 3.2. Найти точку  $y^{(1)}$  из условия исчерпывающего спуска из точки  $y^{(0)}$  по направлению  $p^{(1)}$ :  $f(y^{(1)}) = \min_{\alpha \in R} f(y^{(0)} + \alpha p^{(1)})$ .

**3.3.** Положить  $p^{(2)} = x^{(1)} - y^{(1)}$ , найти точку  $x^{(2)}$  из условия  $f(x^{(2)}) = \min_{\alpha \in R} f(x^{(1)} + \alpha p^{(1)})$ . Вычисления закончить, положив  $x^* = x^{(2)}$ .

Графическая иллюстрация работы алгоритма представлена на рис. 5.8. Поиск точки минимума проводится по так называемым сопряженным направлениям.

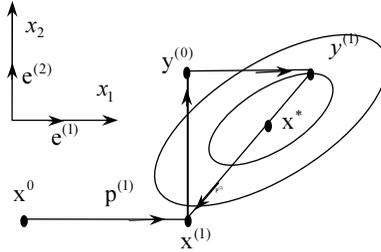


Рис. 5.8

**Определение 5.8.** Ненулевые векторы  $p^{(1)}, \dots, p^{(k)}$  называются сопряженными относительно матрицы  $A$  размера  $(n \times n)$  ( $A$  – ортогональными), если

$$\langle Ap^{(i)}, p^{(j)} \rangle = 0, \quad i \neq j \quad i, j = 1, \dots, k. \quad (5.24)$$

**Пример 5.5**

Показать, что направления  $p^{(1)}$  и  $p^{(2)}$ , использованные в описанном выше алгоритме минимизации квадратичной функции двух переменных, являются  $A$ - ортогональными.

**Решение.** Рассмотрим скалярное произведение

$$\begin{aligned} \langle Ap^{(2)}, p^{(1)} \rangle &= \langle A(x^{(1)} - y^{(1)}), p^{(1)} \rangle = \langle f'(x^{(1)}) - f'(y^{(1)}), p^{(1)} \rangle = \\ &= \langle f'(x^{(1)}), p^{(1)} \rangle - \langle f'(y^{(1)}), p^{(1)} \rangle. \end{aligned}$$

Так как точки  $x^{(1)}$  и  $y^{(1)}$  получены в результате исчерпывающего поиска по направлению  $p^{(1)}$ , то скалярные произведения  $\langle f'(x^{(1)}), p^{(1)} \rangle$  и  $\langle f'(y^{(1)}), p^{(1)} \rangle$  равны нулю (см. (5.18)), поэтому  $\langle Ap^{(2)}, p^{(1)} \rangle = 0$ .

**Лемма 5.1.** Система из  $n$  векторов  $p^{(1)}, \dots, p^{(n)}$ , сопряженных относительно положительно определенной матрицы  $A$ , линейно независима.

**Доказательство.** Предположим противное, т.е. что существует линейная комбинация, равная нулю:

$$\sum_{i=1}^n \gamma_i \mathbf{p}^{(i)} = \mathbf{0}, \quad (5.25)$$

где не все  $\gamma_i = 0$ , например  $\gamma_k \neq 0$ . Умножим обе части равенства (5.25) скалярно на вектор  $A\mathbf{p}^{(k)}$ . Тогда, с учетом свойства (5.24), получим  $\gamma_k \langle A\mathbf{p}^{(k)}, \mathbf{p}^{(k)} \rangle = 0$ . В силу положительной определенности матрицы  $A$  для ненулевого вектора  $\mathbf{p}^{(k)}$  квадратичная форма  $\langle A\mathbf{p}^{(k)}, \mathbf{p}^{(k)} \rangle$  принимает положительное значение и, следовательно,  $\gamma_k = 0$ . Полученное противоречие доказывает лемму.

Таким образом,  $n$  ненулевых  $A$ -ортогональных векторов образуют базис в  $R^n$ .

Рассмотрим минимизацию в  $R^n$  квадратичной функции  $f(\mathbf{x}) = (1/2)\langle A\mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{b}, \mathbf{x} \rangle + c$  с положительно определенной матрицей  $A$  с помощью итерационного процесса

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \alpha_k \mathbf{p}^{(k)}, \quad k = 1, 2, \dots, \quad (5.26)$$

где векторы  $\mathbf{p}^{(k)}$   $A$ -ортогональны.

**Лемма 5.2.** *Если в итерационном процессе (5.26) на каждом шаге используется исчерпывающий спуск, то величина шага  $\alpha_k$ :*

$$\alpha_k = -\frac{\langle \mathbf{f}'(\mathbf{x}^{(0)}), \mathbf{p}^{(k)} \rangle}{\langle A\mathbf{p}^{(k)}, \mathbf{p}^{(k)} \rangle}, \quad k = 1, 2, \dots \quad (5.27)$$

**Доказательство.** Раскрывая рекуррентную формулу (5.26), получаем

$$\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \sum_{i=1}^k \alpha_i \mathbf{p}^{(i)}. \quad (5.28)$$

Из формулы (5.28), учитывая выражение для градиента квадратичной функции  $\mathbf{f}'(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ , находим

$$\mathbf{f}'(\mathbf{x}^{(k)}) = \mathbf{f}'(\mathbf{x}^{(0)}) + \sum_{i=1}^k \alpha_i A\mathbf{p}^{(i)}.$$

Умножая обе части этого равенства скалярно на вектор  $\mathbf{p}^{(k)}$  и учитывая условие исчерпывающего спуска по направлению  $\mathbf{p}^{(k)}$ , т.е.  $\langle \mathbf{f}'(\mathbf{x}^{(k)}), \mathbf{p}^{(k)} \rangle = 0$ , и  $A$ -ортогональность векторов (5.24), получаем

$$\langle f'(x^{(0)}), p^{(k)} \rangle + \alpha_k \langle Ap^{(k)}, p^{(k)} \rangle = 0$$

Так как матрица  $A$  положительно определена, квадратичная форма  $\langle Ap^{(k)}, p^{(k)} \rangle > 0$  и для величины шага  $\alpha_k$  получаем выражение (5.27).

**Теорема 5.9.** *Последовательный исчерпывающий спуск по  $A$ -ортогональным направлениям (5.26) приводит к точке минимума квадратичной функции не более чем за  $n$  шагов.*

**Доказательство.** Согласно лемме 5.1 векторы  $p^{(1)}, \dots, p^{(n)}$  образуют базис в  $R^n$ , поэтому будем искать точку минимума  $x^*$  в виде

$$x^* = x^{(0)} + \sum_{i=1}^k u_i p^{(i)}, \quad (5.29)$$

где  $x^{(0)}$  – произвольная точка  $R^n$ . Подставим выражение (5.29) в необходимое и достаточное условие минимума сильно выпуклой квадратичной функции  $f'(x^*) = Ax^* + b = 0$ :

$$Ax^{(0)} + b + \sum_{i=1}^k u_i Ap^{(i)} = 0.$$

Умножая это равенство скалярно на вектор  $p^{(k)}$ , находим

$$\langle f'(x^{(0)}), p^{(k)} \rangle + u_k \langle Ap^{(k)}, p^{(k)} \rangle = 0$$

или

$$u_k = -\frac{\langle f'(x^{(0)}), p^{(k)} \rangle}{\langle Ap^{(k)}, p^{(k)} \rangle}.$$

Коэффициенты разложения  $u_k$  точки  $x^* - x^{(0)}$  по базису  $p^{(1)}, \dots, p^{(n)}$  совпадают с длинами шагов  $\alpha_k$  исчерпывающего спуска (5.27) в итерационном процессе (5.26). Поэтому определение точки  $x^*$  из (5.29) можно рассматривать как результат  $n$  шагов итерационного процесса (5.26), где  $\alpha_k = u_k$ .

Таким образом, точка минимума квадратичной функции будет найдена не более чем за  $n$  шагов.

Вопрос о нахождении базиса из  $A$ -ортогональных векторов в пространстве  $R^n$  решается неоднозначно. В качестве такого базиса можно, например, взять ортогональный базис из собственных векторов матрицы  $A$ . Однако их поиск особенно при  $n > 2$  представляет собой самостоятельную довольно сложную задачу.

Итерационный процесс (5.26) последовательной одномерной минимизации по сопряженным направлениям  $p^{(k)}$  можно организовать и без предварительного построения векторов  $p^{(1)}, \dots, p^{(n)}$ , последовательно находя их в процессе минимизации, как это было сделано выше для функции двух переменных.

**Алгоритм метода сопряженных направлений**

**1. Итерация 0.**

**1.1.** Выбрать начальную точку  $x^{(0)} \in R^n$ .

**1.2.** Положить  $p^{(1)} = e^{(1)}$ . Найти точку  $x^{(1)}$  из условия исчерпывающего спуска из точки  $x^{(0)}$  по направлению вектора поиска  $p^{(1)}$ :

$$f(x^{(1)}) = \min_{\alpha \in R} f(x^{(0)} + \alpha p^{(1)}).$$

**2. Итерация 1.**

**2.1.** Положить  $y^{(0)} = x^{(1)} + e^{(2)}$ .

**2.2.** Найти точку  $y^{(1)}$  из условия исчерпывающего спуска из точки  $y^{(0)}$  по направлению  $p^{(1)}$ :  $f(y^{(1)}) = \min_{\alpha \in R} f(y^{(0)} + \alpha p^{(1)})$ .

**2.3.** Положить  $p^{(2)} = x^{(1)} - y^{(1)}$ , найти точку  $x^{(2)}$  из условия  $f(x^{(2)}) = \min_{\alpha \in R} f(x^{(1)} + \alpha p^{(1)})$ .

**3. Итерация 2**

**3.1.** Положить  $y^{(1)} = x^{(2)} + e^{(3)}$ .

**3.2.** Найти точку  $y^{(2)}$ , минимизируя  $f(x)$  последовательно по направлениям  $p^{(1)}$  и  $p^{(2)}$ , начиная из точки  $y^{(1)}$ .

**3.3.** Положить  $p^{(3)} = x^{(2)} - y^{(2)}$ , найти точку  $x^{(3)}$  из условия  $f(x^{(3)}) = \min_{\alpha \in R} f(x^{(2)} + \alpha p^{(3)})$ .

**4. Итерация 3**

**4.1.** Положить  $y^{(2)} = x^{(3)} + e^{(4)}$ .

**4.2.** Найти точку  $y^{(3)}$ , минимизируя  $f(x)$  последовательно по направлениям  $p^{(1)}$ ,  $p^{(2)}$ ,  $p^{(3)}$ , начиная из точки  $y^{(2)}$ .

**4.3.** Положить  $p^{(4)} = x^{(3)} - y^{(3)}$ , найти точку  $x^{(4)}$  из условия  $f(x^{(4)}) = \min_{\alpha \in R} f(x^{(3)} + \alpha p^{(4)})$ .

и т. д. . . . .

На  $(n-1)$ -ой итерации находится точка  $x^{(n)}$  в направлении  $p^{(n)}$  из условия  $f(x^{(n)}) = \min_{\alpha \in R} f(x^{(n-1)} + \alpha p^{(n)})$ .

Как и в двумерном случае, можно показать, что направления  $p^{(1)}, \dots, p^{(n)}$ , построенные при выполнении этого алгоритма, являются  $A$ - ортогональными. Поэтому, если  $f(x)$  является квадратичной функцией с положительно определенной матрицей  $A$  и все задачи одномерной минимизации решаются точно, то  $x^* = x^{(n)}$  и вычисления на этом завершаются.

Если же  $f(x)$  не является квадратичной функцией или вспомогательные задачи одномерной минимизации решаются приближенно, то на *последней итерации* выполняется проверка условия окончания вычислительного процесса: если  $\|x^{(0)} - x^{(n)}\| \leq \varepsilon$ , где  $\varepsilon$  – параметр точности, то поиск завершить, полагая  $x^* = x^{(n)}$ , иначе – положить  $x^{(0)} = x^{(n)}$  и перейти к шагу 1.2 алгоритма сопряженных направлений.

Метод сопряженных направлений, описанный выше, относится к числу наиболее эффективных прямых методов. Недостатком его является необходимость решать довольно большое количество задач одномерной минимизации.

### 5.5. Методы оптимизации, использующие производные функций

Пусть функция  $f(x)$  дифференцируема в  $R^n$ . В этом разделе рассмотрены методы, основанные на итерационной процедуре минимизации вида:

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad k = 0, 1, 2, \dots; \quad x^{(0)} \in R^n, \quad (5.30)$$

где направление убывания  $p^{(k)}$  определяется тем или иным способом с учетом информации о частных производных функции  $f(x)$ , а величина шага  $\alpha_k > 0$  такова, что:

$$f(x^{(k+1)}) < f(x^{(k)}), \quad k = 0, 1, 2, \dots . \quad (5.31)$$

Так как функция предполагается дифференцируемой, то в качестве критерия останова в случае бесконечной итерационной последова-

тельности  $\{x^{(k)}\}$ , как правило, выбирается условие  $\|f'(x^{(k)})\| < \varepsilon$ , хотя, разумеется, могут быть использованы и другие критерии.

### Метод наискорейшего спуска

При решении задачи минимизации  $f(x) \rightarrow \min, x \in R^n$  метод градиентного поиска называют *методом наискорейшего спуска*.

Положим в (5.30) на каждом шаге  $p^{(k)} = -f'(x^{(k)})$ . Если  $f'(x^{(k)}) \neq 0$ , то условие (5.20), очевидно, выполнено, т.е. направление вектора  $p^{(k)}$  является направлением убывания функции  $f(x)$ , причем в малой окрестности точки  $x^{(k)}$  направление  $p^{(k)}$  обеспечивает наискорейшее убывание этой функции. Поэтому можно найти такое  $\alpha_k > 0$ , что будет обеспечено выполнение условия (5.31).

В этом варианте метода полагают  $p^{(k)} = -f'(x^{(k)})$  и величина шага  $\alpha_k$  в (5.30) находится в результате решения задачи одномерной минимизации

$$\Phi_k(\alpha) \rightarrow \min, \text{ где } \Phi_k(\alpha) = f(x^{(k)} - \alpha f'(x^{(k)})), \alpha > 0, \quad (5.32)$$

т.е. на каждой итерации в направлении антиградиента  $-f'(x^{(k)})$  совершается исчерпывающий спуск.

#### Алгоритм метода наискорейшего спуска

1. Задать параметр точности  $\varepsilon > 0$ , выбрать  $x \in R^n$ .
2. Вычислить  $f'(x)$  и проверить условие достижения точности:  $\|f'(x)\| < \varepsilon$ . Если оно выполнено, то положить  $x^* = x$ ,  $f^* = f(x)$  и поиск завершить, иначе – перейти к шагу 3.
3. Решить задачу одномерной минимизации (5.32) для  $x^{(k0)} = x$ , т.е. найти  $\alpha^*$ . Положить  $x = x - \alpha^* f'(x)$  и перейти к шагу 2.

### Пример 5.6

Решить следующую задачу градиентным методом:

$$f(x) = x_1^4 + x_2^2 + 4x_1x_2 + 8x_1 + 6x_2 \rightarrow \min.$$

Решение. Используем рассмотренный выше вариант градиентного метода, в котором на каждой итерации в направлении антиградиента совершается исчерпывающий поиск минимума. Для определения  $\alpha^*$  на каждой итерации применяем один из методов одномерного поиска (например, поразрядного поиска).

Определяем компоненты вектора–градиента:

$$\frac{\partial f(x)}{\partial x_1} = 4x_1^3 + 4x_2 + 8; \quad \frac{\partial f(x)}{\partial x_2} = 4x_1 + 2x_2 + 6. \quad (5.33)$$

Задаем, с учетом особенностей задачи, параметр точности  $\varepsilon = 0.01$  и начальные условия  $x^{(0)} = (0; 0)$ , при этом  $f(x^{(0)}) = 0$ .

*Итерация 1.* Полагаем  $x^{(k)} = x^{(0)}$ .

1. Направление поиска  $p^{(k)} = -f'(x^{(k)}) = (-8, -6)$ . Так как норма вектора  $\|f'(x^{(k)})\| = 10 > \varepsilon$ , то решение продолжается.

2. Новое решение задачи:  $x^{(k+1)} = x^{(k)} - \alpha f'(x^{(k)})$ . Одним из методов одномерного поиска (например, поразрядного) определяем  $\alpha^* = 0.135$ , при этом  $x^{(k+1)} = (-1.08; -0.81)$  и  $f(x^{(k+1)}) = -7.9842$ .

Вектор градиента в этой точке имеет следующие компоненты:  $f'(x^{(k+1)}) = (-0.2789; 0.06)$ . Норма вектора  $\|f'(x^{(k+1)})\| = 0,28 > \varepsilon$ . Следовательно, решение продолжается.

3. Полагаем  $x^{(k)} = (-1.08; -0.81)$  и  $f(x^{(k)}) = -7.9842$ .

*Итерация 2.* Начальные условия  $x^{(k)} = (-1.08; -0.81)$ .

1. Направление поиска  $p^{(k)} = -f'(x^{(k)}) = (0.2789; -0.06)$ .

2. Новое решение задачи:  $x^{(k+1)} = x^{(k)} - \alpha f'(x^{(k)})$ . Методом одномерного поиска определяем  $\alpha^* = 0.135$ . При этом  $x^{(k+1)} = (-1.08; -0.81)$  и значение функции  $f(x^{(k+1)}) = -7.9842$ . Вектор градиента в этой точке:  $f'(x^{(k+1)}) = (0.0763; 0.1592)$ . Норма вектора  $\|f'(x^{(k+1)})\| = 0.1769 > \varepsilon$ . Следовательно, решение продолжается.

Аналогично выполняются все последующие итерации до получения решения с заданной точностью (эти итерации здесь не приводятся). На 27-ой итерации было получено решение  $x^{(k+1)} = (-1.0017; -0.9966)$  и  $f(x^{(k+1)}) = -7.9998$ . Вектор-градиент:  $f'(x^{(k+1)}) = (-0.0069; 0)$ . Норма вектора  $\|f'(x^{(k+1)})\| = 0.0069 < \varepsilon$ . Следовательно, получено оптимальное решение с заданной точностью:  $x^* = (-1.0017; -0.9966)$  и  $f_{\min} = f(x^*) = -7.9998$ .

Если функция  $f(x)$  квадратичная, то для величины шага исчерпывающего поиска можно воспользоваться формулой (5.19) при  $p^{(k)} = -f'(x^{(k)})$ .

### Метод сопряженных градиентов

До сих пор в итерационной процедуре (5.30) в качестве направления убывания функции  $f(x)$  использовалось направление антиградиента:  $p^{(k)} = -f'(x^{(k)})$ . Однако такой выбор направления убывания не всегда бывает удачным. В частности, для *плохо обусловленных задач* минимизации направление антиградиента в точке  $x^{(k)}$  может значительно отличаться от направления к точке минимума  $x^*$ . В результате траектория приближения к точке минимума имеет зигзагообразный характер.

В разд. 5.4 был описан метод сопряженных направлений, позволяющий найти точку минимума квадратичной функции за конечное число шагов. Опишем метод, позволяющий получить сопряженные направления для квадратичной функции  $f(x)$  с использованием ее производных.

В этом методе используется итерационный процесс:

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad k = 0, 1, 2, \dots; \quad x^{(0)} \in R^n, \quad p^{(0)} = -f'(x^{(0)}), \quad (5.34)$$

в котором величина шага  $\alpha_k$  находится из условия исчерпывающего спуска по направлению  $p^{(k)}$ . После вычисления очередной точки  $x^{(k+1)}$ ,  $k = 0, 1, 2, \dots$  новое направление поиска  $p^{(k+1)}$  находится по формуле:

$$p^{(k+1)} = -f'(x^{(k+1)}) + \beta_k p^{(k)}, \quad k = 0, 1, 2, \dots, \quad (5.35)$$

где коэффициенты  $\beta_k$  выбираются так, чтобы при минимизации квадратичной функции  $f(x)$  с положительно определенной матрицей  $A$  получалась последовательность  $A$ -ортогональных векторов  $p^{(0)}, p^{(1)}, \dots$ .

Из условия  $\langle Ap^{(k+1)}, p^{(k)} \rangle = 0$  имеем:

$$\beta_k = -\frac{\langle Af'(x^{(k+1)}), p^{(k)} \rangle}{\langle Ap^{(k)}, p^{(k)} \rangle}. \quad (5.36)$$

Напомним, что для квадратичной функции шаг исчерпывающего спуска по направлению  $p^{(k)}$  равен:

$$\alpha_k = -\frac{\langle f'(x^{(k)}), p^{(k)} \rangle}{\langle Ap^{(k)}, p^{(k)} \rangle}. \quad (5.37)$$

Можно показать, что процесс (5.34) – (5.37) минимизации квадратичной функции с положительно определенной симметрической матрицей  $A$  дает точки  $x^{(0)}, \dots, x^{(k)}$  и векторы  $p^{(0)}, \dots, p^{(k)}$  такие, что если  $f'(x^{(i)}) \neq 0$  при  $0 \leq i < k \leq n-1$ , то векторы  $p^{(0)}, \dots, p^{(k)}$   $A$ -ортогональны, а градиенты  $f'(x^{(0)}), \dots, f'(x^{(k)})$  взаимно ортогональны.

Обращение градиента в нуль в очередной точке  $x^{(k)}$  итерационного процесса свидетельствует о достижении точки глобального минимума.

Так как направления  $p^{(k)}$  в (5.34) являются  $A$ -ортогональными, рассматриваемый метод гарантирует нахождение точки минимума сильно выпуклой квадратичной функции не более чем за  $n$  шагов (согласно теореме 5.9).

С учетом взаимной ортогональности градиентов  $f'(x^{(i)})$  и условий исчерпывающего спуска по направлениям  $p^{(k)}$  можно упростить выражения (5.36) и (5.37). Представим числитель дроби (5.37) в следующем виде:

$$\begin{aligned} \langle f'(x^{(k)}), p^{(k)} \rangle &= \langle f'(x^{(k)}), -f'(x^{(k)}) + \beta_{k-1} p^{(k-1)} \rangle = \\ &= -\|f'(x^{(k)})\|^2 + \beta_{k-1} \langle f'(x^{(k)}), p^{(k-1)} \rangle = -\|f'(x^{(k)})\|^2. \end{aligned}$$

Умножив обе части равенства (5.34) слева на матрицу  $A$  и прибавив к ним по вектору  $b$ , получим:

$$f'(x^{(k+1)}) = f'(x^{(k)}) + \alpha_k Ap^{(k)}. \quad (5.38)$$

С учетом формулы (5.38) упростим числитель в выражении (5.36) для  $\beta_k$  следующим образом:

$$\begin{aligned} \langle Af'(x^{(k+1)}), p^{(k)} \rangle &= \langle f'(x^{(k+1)}), Ap^{(k)} \rangle = \\ &= \left\langle f'(x^{(k+1)}), \frac{f'(x^{(k+1)}) - f'(x^{(k)})}{\alpha_k} \right\rangle = \frac{\|f'(x^{(k+1)})\|^2}{\alpha_k}. \end{aligned}$$

В результате выражения для  $\alpha_k$  и  $\beta_k$  примут вид:

$$\alpha_k = -\frac{\|f'(x^{(k)})\|^2}{\langle Ap^{(k)}, p^{(k)} \rangle},$$

$$\beta_k = -\frac{\|f'(x^{(k+1)})\|^2}{\|f'(x^{(k)})\|^2}. \quad (5.39)$$

Выражение (5.39) для коэффициента  $\beta_k$  не содержит в явном виде матрицу  $A$  квадратичной функции. Поэтому метод сопряженных градиентов может применяться и для минимизации неквадратичных функций. В этом случае итерационный процесс метода описывается соотношениями:

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad x^{(0)} \in R^n, \quad p^{(0)} = -f'(x^{(0)}), \quad k = 0, 1, 2, \dots; \quad (5.40)$$

$$f(x^{(k)} + \alpha_k p^{(k)}) = \min_{\alpha > 0} f(x^{(k)} + \alpha_k p^{(k)}), \quad k = 0, 1, 2, \dots; \quad (5.41)$$

$$p^{(k+1)} = -f'(x^{(k+1)}) + \beta_k p^{(k)}, \quad k = 0, 1, 2, \dots; \quad (5.42)$$

$$\beta_k = \frac{\|f'(x^{(k+1)})\|^2}{\|f'(x^{(k)})\|^2}, \quad k = 0, 1, 2, \dots \quad (5.43)$$

Разумеется, что процесс (5.40) – (5.43) может не приводить к точке минимума неквадратичной функции  $f(x)$  за конечное число итераций. К тому же точное определение  $\alpha_k$  из условия (5.41) возможно лишь в редких случаях. Поэтому реализация каждой итерации метода будет сопровождаться неизбежными погрешностями. Эти погрешности, накапливаясь, могут привести к тому, что векторы  $p^{(k)}$  перестанут указывать направление убывания функции и сходимость метода может нарушиться. Поэтому на практике в методе сопряженных градиентов через  $N$  шагов производят обновление метода, полагая  $\beta_{mN} = 0$ ,  $m = 1, 2, \dots$ . Номера  $mN$  называются моментами обновления метода (рестарта). Часто полагают  $N = n$  – размерности пространства  $R^n$ . Если  $N = 1$ , то получается частный случай метода сопряженных градиентов – метод наискорейшего спуска.

#### *Алгоритм метода сопряженных градиентов*

Задать параметр точности  $\varepsilon > 0$ , выбрать  $x^{(0)} \in R^n$ , найти  $f'(x^{(0)})$ .

1. Положить  $k = 0$ ,  $p^{(0)} = -f'(x^{(0)})$ .
2. Решить задачу одномерной минимизации

$$f(x^{(k)} + \alpha p^{(k)}) \rightarrow \min, \quad \alpha > 0, \quad \text{т.е. найти } \alpha = \alpha_k.$$

3. Положить  $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$  и вычислить  $f'(x^{(k+1)})$ . Проверить условие достижения точности  $\|f'(x^{(k+1)})\| < \varepsilon$ . Если оно выполняется, то положить  $x^* = x^{(k+1)}$ ,  $f_{\min} = f(x^*)$  и закончить поиск, иначе – перейти к шагу 4.

4. Проверить условие  $k + 1 = n$ . Если оно выполняется, то положить  $x^{(0)} = x^{(k+1)}$ ,  $f'(x^{(0)}) = f'(x^{(k+1)})$  и перейти к шагу 1 (рестарт), иначе – перейти к шагу 5.

5. Вычислить коэффициент  $\beta_k = \|f'(x^{(k+1)})\|^2 / \|f'(x^{(k)})\|^2$  и найти новое направление поиска  $p^{(k+1)} = -f'(x^{(k+1)}) + \beta_k p^{(k)}$ . Положить  $k = k + 1$  и перейти к шагу 2.

Вблизи точки минимума дважды дифференцируемая функция с положительно определенной матрицей Гессе  $f''(x^*)$ , как правило, достаточно хорошо аппроксимируется квадратичной функцией. Поэтому можно надеяться на хороший результат применения этого метода для таких функций.

**Пример 5.7.** Методом сопряженных градиентов найти точку минимума функции  $f(x) = 4x_1^2 + 3x_2^2 - 4x_1x_2 + x_1$  из начальной точки  $x^{(0)} = (0, 0)$ .

**Решение**

*Итерация 1*

Зададим  $\varepsilon = 0.01$ ,  $x^{(0)} = (0, 0)$ , найдем  $f'(x^{(0)}) = (1, 0)$ .

1. Положим  $k = 0$ ,  $p^{(0)} = f'(x^{(0)}) = (-1, 0)$ .

2. Решим задачу одномерной минимизации  $f(x^{(0)} + \alpha p^{(0)}) \rightarrow \min$ . Получим  $\alpha_0 = 0.125$  (для нахождения  $\alpha_0$  можно было воспользоваться формулой (5.19)).

3. Найдем  $x^{(1)} = x^{(0)} + \alpha_0 p^{(0)} = (-0.125, 0)$  и  $f'(x^{(1)}) = (0, 0.5)$ . Точность не достигнута, перейдем к шагу 4.

4. Условие  $k + 1 = n$  не выполняется, перейдем к шагу 5.

5. Найдем коэффициент  $\beta_0 = 0.25$  и новое направление спуска  $p^{(1)} = -f'(x^{(1)}) + \beta_0 p^{(0)} = (-0.25, -0.5)$

*Итерация 2*

2. Решим задачу одномерной минимизации  $f(x^{(1)} + \alpha p^{(1)}) \rightarrow \min$ . Получим  $\alpha_1 = 0.25$ .

3. Найдем  $x^{(2)} = x^{(1)} + \alpha_1 p^{(1)} = (-0.1875, -0.125)$  и

$f'(x^{(2)}) = (0, 0)$  – задача решена точно.

Таким образом,  $x^* = x^{(2)}$ . Решение получено в результате двух итераций метода сопряженных градиентов, поскольку целевая функция квадратичная в  $R^n$  и одномерные задачи минимизации на шаге 2 алгоритма решены точно.

### Метод Ньютона

Пусть функция  $f(x)$  дважды дифференцируема в  $R^n$ . Тогда для нее разложение по формуле Тейлора в окрестности точки  $x^{(k)}$ :

$$f(x) = f(x^{(k)}) + \langle f'(x^{(k)}), x - x^{(k)} \rangle + \frac{1}{2} \langle f''(x^{(k)})(x - x^{(k)}), x - x^{(k)} \rangle + o(\|\Delta x^{(k)}\|^2).$$

Отсюда видно, что поведение функции  $f(x)$  с точностью до величины порядка  $o(\|\Delta x^{(k)}\|^2) = o(\|x - x^{(k)}\|^2)$  может быть описано квадратичной функцией

$$\Phi_k(x) = \frac{1}{2} \langle f''(x^{(k)})(x - x^{(k)}), x - x^{(k)} \rangle + \langle f'(x^{(k)}), x - x^{(k)} \rangle + f(x^{(k)}). \quad (5.44)$$

Минимизируем функцию  $\Phi_k(x)$  вместо  $f(x)$ . Найдем ее точку минимума  $x^{(k+1)}$  из условия  $\Phi'_k(x) = 0$ :

$$\Phi'_k(x) = f''(x^{(k)})(x - x^{(k)}) + f'(x^{(k)}) = 0. \quad (5.45)$$

Пусть матрица Гессе  $f''(x^{(k)})$  положительно определена при всех  $x \in R^n$  и, следовательно, невырождена (т.е.  $\det f''(x^{(k)}) > 0$ ). Тогда существует обратная матрица  $[f''(x^{(k)})]^{-1}$ . Отметим, что квадратичная функция (5.44) с положительно определенной матрицей  $f''(x^{(k)})$  сильно выпукла и уравнение (5.45) определяет единственную точку глобального минимума функции  $\Phi_k(x)$ . Умножим слева обе части равенства (5.45) на матрицу  $[f''(x^{(k)})]^{-1}$  и найдем точку минимума квадратичной функции (5.44), аппроксимирующей  $f(x)$  в окрестности точки  $x = x^{(k)}$ :

$$x^{(k+1)} = x^{(k)} - [f''(x^{(k)})]^{-1} f'(x^{(k)}), \quad k = 0, 1, \dots \quad (5.46)$$

Итерационный процесс (5.46), начатый из произвольной точки  $x^{(0)} \in R^n$ , называется *методом Ньютона минимизации функции мно-*

гих переменных и является обобщением метода Ньютона в одномерном случае (см. разд. 4.3).

Очевидно, для квадратичной функции с положительно определенной матрицей  $A$  применение метода Ньютона обеспечивает получение точки глобального минимума ровно за один шаг из любой точки  $x^{(0)} \in R^n$ .

Для выпуклой неквадратичной функции применение этого метода обеспечивает, как правило, быструю сходимость. Дело в том, что на каждом шаге итерационного процесса (5.46) используется информация

о поведении функции  $f(x)$  в окрестности точки  $x^{(k)}$ , содержащаяся не только в значениях первых, но и вторых ее частных производных. Поэтому при прочих равных условиях следует ожидать более быструю сходимость метода Ньютона по сравнению с градиентными методами.

При выборе достаточно хорошего начального приближения  $x^{(0)} \in R^n$  минимизирующая последовательность  $\{x^{(k)}\}$  для сильно выпуклой дважды дифференцируемой функции  $f(x)$  сходится к точке минимума с квадратичной скоростью  $\rho(x^{(k)}, x^*) \leq cq^{2^k}$ ,  $q \in (0; 1)$ . Если же точка  $x^{(0)}$  выбрана недостаточно близкой к точке  $x^*$ , то последовательность (5.46) может расходиться (см. разд. 5.3).

Отметим, что даже сходящаяся последовательность  $\{x^{(k)}\}$  метода Ньютона не всегда обеспечивает монотонное убывание  $f(x)$ , т.е. неравенство  $f(x^{(k+1)}) < f(x^{(k)})$  для некоторых  $k = 0, 1, \dots$  может нарушаться. Этот недостаток устранен в *обобщенном методе Ньютона*:

$$x^{(k+1)} = x^{(k)} - \alpha_k [f''(x^{(k)})]^{-1} \cdot f'(x^{(k)}), \quad k = 0, 1, \dots,$$

где величина  $\alpha_k > 0$  находится на каждом шаге из условия исчерпывающего спуска по направлению  $p^{(k)} = -[f''(x^{(k)})]^{-1} \cdot f'(x^{(k)})$ .

Недостатком метода Ньютона является необходимость вычисления и обращения матрицы Гессе на каждой итерации.

### **Квазиньютоновские методы**

Эти методы близки по скорости сходимости к обобщенному методу Ньютона, но не используют вторые производные целевой функции  $f(x)$  и процедуры обращения матрицы  $f''(x^{(k)})$ . В методах этого типа используется итерационный процесс

$$x^{(k+1)} = x^{(k)} - \alpha_k H^{(k)} f'(x^{(k)}), \quad (5.47)$$

где  $H^{(k)}$  – некоторая квадратная матрица размера  $(n \times n)$ , а величина шага  $\alpha_k > 0$  выбирается из условия исчерпывающего поиска вдоль направления  $\mathbf{p}^{(k)} = -H^{(k)} \mathbf{f}'(\mathbf{x}^{(k)})$ .

В квазиньютоновских методах матрица  $H^{(k)}$  строится с помощью рекуррентных формул на каждом шаге процесса (5.47). Вид этих формул определяет метод. Рассмотрим в качестве примера *метод Давидона*

– *Флетчера – Пауэла* (ДФП – метод). Рекуррентная формула для  $H^{(k)}$  имеет вид:

$$H^{(k+1)} = H^{(k)} + A^{(k)} + B^{(k)}, \quad H^{(0)} = E, \quad A^{(0)} = B^{(0)} = 0. \quad (5.48)$$

Матрицы  $A^{(k)}$  и  $B^{(k)}$  выражаются на каждом шаге через матрицу  $H^{(k)}$  и векторы-столбцы  $\mathbf{v}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ ,  $\mathbf{u}^{(k)} = \mathbf{f}'(\mathbf{x}^{(k+1)}) - \mathbf{f}'(\mathbf{x}^{(k)})$  следующим образом:

$$A^{(k)} = \frac{\mathbf{v}^{(k)} \cdot [\mathbf{v}^{(k)}]^T}{\langle \mathbf{v}^{(k)}, \mathbf{u}^{(k)} \rangle}, \quad B^{(k)} = -\frac{H^{(k)} \mathbf{u}^{(k)} [\mathbf{u}^{(k)}]^T H^{(k)}}{\langle H^{(k)} \mathbf{u}^{(k)}, \mathbf{u}^{(k)} \rangle} \quad (5.49)$$

В итерационном процессе (5.47) – (5.49) при минимизации выпуклой дифференцируемой функции  $f(\mathbf{x})$  в каждой очередной точке  $\mathbf{x}^{(k)}$  матрица  $H^{(k)}$  обновляется на основании информации о значениях градиентов  $\mathbf{f}'(\mathbf{x}^{(k)})$  в соответствующих точках. В пределе матрица  $H^{(k)}$  становится равной матрице  $\mathbf{f}''(\mathbf{x}^{(k)})$ .

#### **Алгоритм метода ДФП**

1. Задать параметр точности  $\varepsilon > 0$ , выбрать начальное значение  $\mathbf{x} \in R^n$ . Вычислить  $\mathbf{f}'(\mathbf{x})$ , положить  $H = E$ .
2. Найти направление  $\mathbf{p} = -H \cdot \mathbf{f}'(\mathbf{x})$ .
3. Решить задачу одномерной минимизации  $f(\mathbf{x} + \alpha \mathbf{p}) \rightarrow \min$ ,  $\alpha > 0$ , т.е. найти  $\alpha^*$ .
4. Положить  $\hat{\mathbf{x}} = \mathbf{x} + \alpha^* \mathbf{p}$  и найти  $\mathbf{f}'(\hat{\mathbf{x}})$ . Проверить условие достижения точности  $\|\mathbf{f}'(\hat{\mathbf{x}})\| < \varepsilon$ . Если оно выполнено, то положить  $\mathbf{x}^* = \hat{\mathbf{x}}$ ,  $f^* = f(\hat{\mathbf{x}})$  и прекратить поиск, иначе перейти к следующему шагу 5.
5. Найти векторы  $\mathbf{v} = \hat{\mathbf{x}} - \mathbf{x}$ ,  $\mathbf{u} = \mathbf{f}'(\hat{\mathbf{x}}) - \mathbf{f}'(\mathbf{x})$  и матрицы:

$$A = \frac{\mathbf{v} \cdot \mathbf{v}^T}{\langle \mathbf{v}, \mathbf{u} \rangle}, \quad B = -\frac{H \mathbf{u} \cdot \mathbf{u}^T H}{\langle H \mathbf{u}, \mathbf{u} \rangle}.$$

Положить  $H = H + A + B$ ,  $\mathbf{x} = \hat{\mathbf{x}}$ ,  $f'(\mathbf{x}) = f'(\hat{\mathbf{x}})$  и перейти к шагу 2.

Существуют и другие варианты квазиньютоновских методов. Достоинством квазиньютоновских методов является их быстрая сходимость и отсутствие операции обращения матрицы на каждой итерации. Недостатком этих методов является необходимость хранения матриц в памяти компьютера, что при решении задач высокой размерности может создать определенные трудности.

### Пример 5.8

Решить следующую задачу квазиньютоновским методом, используя алгоритм ДФП-метода:

$$f(\mathbf{x}) = x_1^4 + x_2^2 + 4x_1x_2 + 8x_1 + 6x_2 \rightarrow \min.$$

**Решение.** Поиск минимума основан на итерационных формулах (5.47) – (5.49). На каждой итерации в направлении  $\mathbf{p}^{(k)} = -H^{(k)} f'(\mathbf{x}^{(k)})$  выполняется исчерпывающий поиск минимума одним из методов одномерной минимизации для определения  $\alpha^*$ .

Компоненты вектора градиента:

$$\frac{\partial f(\mathbf{x})}{\partial x_1} = 4x_1^3 + 4x_2 + 8; \quad \frac{\partial f(\mathbf{x})}{\partial x_2} = 4x_1 + 2x_2 + 6.$$

Задаем, с учетом характера задачи, параметр точности  $\varepsilon = 0.01$  и начальные условия  $\mathbf{x}^{(0)} = (0; 0)$ , при этом  $f(\mathbf{x}^{(0)}) = 0$ .

*Итерация 1*

1. Находим направление поиска минимума функции  $f(\mathbf{x})$  в начальной точке:

$$\mathbf{p} = -H f'(\mathbf{x}) = -\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 8 \\ 6 \end{pmatrix} = \begin{pmatrix} -8 \\ -6 \end{pmatrix}.$$

2. Решим задачу  $f(\mathbf{x} + \alpha \mathbf{p}) \rightarrow \min$  методом поразрядного поиска. Получаем  $\alpha^* = 0.135$ .

3. Находим  $\hat{\mathbf{x}} = \mathbf{x} + \alpha^* \mathbf{p} = (-1.08; -0.81)$ ,  $f'(\hat{\mathbf{x}}) = (-0.2789; 0.06)$ . Так как  $\|f'(\hat{\mathbf{x}})\| = 0.285 > \varepsilon$ , то переходим к шагу 4.

4. Находим векторы  $\mathbf{v} = \hat{\mathbf{x}} - \mathbf{x} = (-1.08; -0.81)$ ,  $\mathbf{u} = f'(\hat{\mathbf{x}}) - f'(\mathbf{x}) = (-0.2789; 0.06) - (8; 6) = (-8.2789; -5.94)$  и матрицы

$$A = \frac{\mathbf{v} \mathbf{v}^T}{\langle \mathbf{v}, \mathbf{u} \rangle} = \frac{\begin{pmatrix} -1.08 \\ -0.81 \end{pmatrix} \begin{pmatrix} -1.08 & -0.81 \end{pmatrix}}{\begin{pmatrix} -1.08 & -0.81 \end{pmatrix} \begin{pmatrix} -8.2789 \\ -5.94 \end{pmatrix}} = \begin{pmatrix} 0.0849 & 0.0636 \\ 0.0636 & 0.0477 \end{pmatrix},$$

$$B = \frac{\mathbf{H} \mathbf{u} \cdot \mathbf{u}^T \mathbf{H}}{\langle \mathbf{H} \mathbf{u}, \mathbf{u} \rangle} = - \frac{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -8.2789 \\ -5.94 \end{pmatrix} \begin{pmatrix} -8.2789 & -5.94 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}{\begin{pmatrix} -8.2789 & -5.94 \end{pmatrix} \begin{pmatrix} -8.2789 \\ -5.94 \end{pmatrix}} = - \begin{pmatrix} 0.0284 & 0.0204 \\ 0.0204 & 0.0146 \end{pmatrix},$$

$$\mathbf{H} = \mathbf{H} + \mathbf{A} + \mathbf{B} = \begin{pmatrix} 1.0565 & 0.0432 \\ 0.0432 & 1.0331 \end{pmatrix}.$$

### Итерация 2

1. Находим направление поиска минимума функции  $f(\mathbf{x})$  в следующей точке:

$$\mathbf{p} = -\mathbf{H} \mathbf{f}'(\mathbf{x}) = - \begin{pmatrix} 1.0565 & 0.0432 \\ 0.0432 & 1.0331 \end{pmatrix} \cdot \begin{pmatrix} -0.2789 \\ 0.067 \end{pmatrix} = \begin{pmatrix} 0.2921 \\ -0.0499 \end{pmatrix}.$$

2. Решим задачу  $f(\mathbf{x} + \alpha \mathbf{p}) \rightarrow \min$ . Получаем  $\alpha^* = 0.08$ .

3. Находим  $\hat{\mathbf{x}} = \mathbf{x} + \alpha^* \mathbf{p} = (-1.0566; -0.814)$ ,  $f(\hat{\mathbf{x}}) = -7.9875$ .  
Градиент  $\mathbf{f}'(\hat{\mathbf{x}}) = (0.0256; 0,1456)$ . Так как  $\|\mathbf{f}'(\hat{\mathbf{x}})\| = 0.148 > \varepsilon$ , то переходим к следующему шагу.

4. Находим соответственно векторы

$$\mathbf{v} = \hat{\mathbf{x}} - \mathbf{x} = (0.0234; -0.004), \quad \mathbf{u} = \mathbf{f}'(\hat{\mathbf{x}}) - \mathbf{f}'(\mathbf{x}) = (0.3045; 0.0856)$$

и матрицы

$$\mathbf{A} = \begin{pmatrix} 0.0085 & -0.0015 \\ -0.0015 & 0.0003 \end{pmatrix}; \quad \mathbf{B} = \begin{pmatrix} -0.9824 & -0.3071 \\ -0.3071 & -0.0965 \end{pmatrix}; \quad \mathbf{H} = \begin{pmatrix} 0.0826 & -0.2654 \\ -0.2654 & 0.9369 \end{pmatrix}.$$

### Итерация 3

1. Находим новое направление поиска минимума функции  $f(\mathbf{x})$  в полученной точке  $\mathbf{x} = (-1,0566; -0,814)$ :

$$\mathbf{p} = -\mathbf{H} \mathbf{f}'(\mathbf{x}) = - \begin{pmatrix} 0.0826 & -0.2654 \\ -0.2654 & 0.9369 \end{pmatrix} \cdot \begin{pmatrix} -0.0256 \\ 0.1456 \end{pmatrix} = \begin{pmatrix} 0.0365 \\ -0.1297 \end{pmatrix}.$$

2. Решим задачу  $f(\mathbf{x} + \alpha \mathbf{p}) \rightarrow \min$ . Получаем  $\alpha^* = 1.2$ .

3. Находим  $\hat{\mathbf{x}} = \mathbf{x} + \alpha^* \mathbf{p} = (-1.0128; -0.9697)$ ,  $f(\hat{\mathbf{x}}) = -8.0014$ .

Градиент  $\mathbf{f}'(\hat{\mathbf{x}}) = (0.0035; 0.0084)$ . Так как  $\|\mathbf{f}'(\hat{\mathbf{x}})\| = 0.0091 < \varepsilon$ , то получено оптимальное решение:

$$\mathbf{x}^* = (-1.0128; -0.9697); \quad f_{\min} = -8.0014.$$

## Глава 6

### ЭЛЕМЕНТЫ ТЕОРИИ ВЫПУКЛОГО ПРОГРАММИРОВАНИЯ

Эта глава посвящена исследованию задач оптимизации, сводящихся к поиску точек минимума многомерной функции в области допустимы решений  $U$ . В большинстве случаев допустимое множество  $U$  определяется ограничениями-равенствами и (или) неравенствами. Такие задачи называются *задачами математического программирования*.

Рассмотрены метод множителей Лагранжа и теоремы Куна–Таккера, определяющие существование условного минимума.

#### 6.1. Постановка задачи математического программирования

Многие практические задачи оптимизации сводятся к математическим моделям вида

$$f(x) \rightarrow \min, \quad x \in U \subset R^n,$$

где допустимое множество  $U$  не совпадает со всем пространством  $R^n$ . В этих случаях говорят об *условной минимизации* функции  $n$  переменных.

Рассмотрим задачу математического программирования в следующем виде:

$$f(x) \rightarrow \min, \quad x \in R^n; \quad (6.1)$$

$$\begin{cases} g_i(x) \leq 0, & i \in I_1, \end{cases} \quad (6.2)$$

$$\begin{cases} g_i(x) = 0, & i \in I_2, \end{cases} \quad (6.3)$$

где  $I_1, I_2 \subset \{1, 2, \dots, m\}$  – заданные множества индексов, удовлетворяющие условиям  $I_1 \cup I_2 = \{1, 2, \dots, m\}$ ,  $I_1 \cap I_2 = \emptyset$ . Соотношения (6.1) – (6.3) представляют собой запись условий *задачи математического программирования*. Отметим несколько важных частных видов задач математического программирования.

1. *Задача линейного программирования* – в (6.1) – (6.3) все функции  $f(x)$ ,  $g_i(x)$  линейны. Методы решения задач линейного программирования уже рассмотрены в гл. 2.

2. *Задача нелинейного программирования* – хотя бы одна из функций  $f(x)$ ,  $g_i(x)$  в (6.1) – (6.3) не является линейной.

3. *Задача на условный экстремум* – отсутствуют ограничения-неравенства (6.2), т.е.  $I_1 = \emptyset$ .

4. *Задача выпуклого программирования* – все функции  $f(x)$ ,  $g_i(x)$  в (6.1) и (6.3) выпуклы, а ограничения-равенства отсутствуют, т.е.  $I_2 = \emptyset$ . Ее допустимое множество  $U$  – выпукло (см. теорему 5.3).

**Пример 6.1.** На двух предприятиях отрасли необходимо изготовить  $N$  изделий некоторой продукции. Затраты на производство  $x_1$  изделий (в ед. стоимости) на первом предприятии составляют  $ax_1^2 + c$ , а на производство  $x_2$  изделий на втором предприятии составляют  $bx_2$ . Составить математическую модель задачи оптимизации количества изделий на каждом из предприятий, чтобы общие затраты на их производство были минимальными.

### **Решение**

Построим математическую модель задачи. *Объектом оптимизации* является план выпуска продукции.

*Управляемые переменные* – это планируемые количества изделий для первого и второго предприятий  $x_1$  и  $x_2$ .

Очевидно, они должны быть неотрицательными:

$$x_1 \geq 0, \quad x_2 \geq 0 \quad (6.4)$$

и в сумме составлять плановое задание

$$x_1 + x_2 = N. \quad (6.5)$$

*Целевой функцией* являются совокупные затраты на производство изделий  $f(x) = ax_1^2 + c + bx_2$ .

Математическая постановка задачи состоит в определении минимального значения целевой функции

$$f(x) = ax_1^2 + bx_2 + c \rightarrow \min$$

при ограничениях (6.4) и (6.5). Это задача нелинейного программирования.

Решение задач нелинейного математического программирования, как правило, связано с более существенными трудностями, чем решение мостью с величиной  $b_1$ . Выразим частную производную  $\partial f^*/\partial b_1$  помощью правила дифференцирования сложной функции:

задач линейного программирования или задач безусловной минимизации. Например, область допустимых решений  $U$  может состоять из нескольких непересекающихся множеств. Глобальный экстре-

мум может находиться как в любой точке границы области  $U$ , так и внутри области  $U$ . И, конечно, основная проблема – поиск глобального экстремума задачи нелинейного программирования в общем случае.

**Пример 6.2.**

Решить следующую задачу нелинейного программирования графическим методом:

$$f(x) = -(x_1 - 12)^2 - x_2^2 \rightarrow \min ,$$

$$\begin{cases} x_1 + x_2 \geq 7, & (1) \\ (x_1 - 7)^2 + (x_2 - 7)^2 \geq 25, & (2) \\ x_1 \leq 10, & (3) \quad x_2 \leq 10, & (4) \end{cases}$$

**Решение.** Для решения задачи нелинейного программирования графическим методом на основании системы ограничений (1) – (4) строится область допустимых решений (ОДР). В нашей задаче ОДР состоит из двух не связанных множеств, которые на рис. 6.1 выделены внутренней штриховкой.

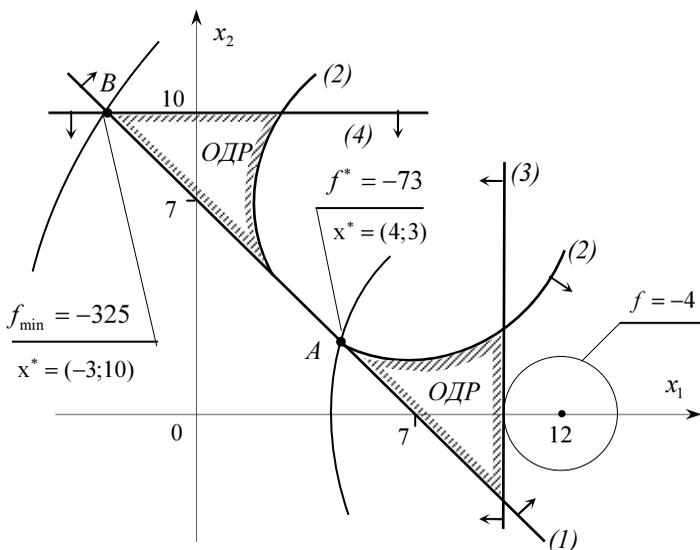


Рис. 6.1.

Целевая функция задачи геометрически изображается множеством линий уровня в виде концентрических окружностей с центром (12, 0). Каждой окружности соответствует определенное значение

целевой функции  $f(x)$ . С увеличением радиуса окружности значение целевой функции уменьшается, т.к.  $f(x) = -r^2$ , где  $r$  – радиус.

При решении задачи выбираются такие окружности (линии уровня), которые имеют общие точки с *крайними* точками ОДР. Крайняя точка  $A$  с координатами  $x_1^* = 4$ ,  $x_2^* = 3$  и значением целевой функции  $f^* = -73$  соответствует *локальному* минимуму (рис. 6.1) и определяется как одно из решений системы уравнений:

$$\begin{cases} x_1 + x_2 = 7, & (1) \\ (x_1 - 7)^2 + (x_2 - 7)^2 = 25. & (2) \end{cases}$$

Точкой *глобального* минимума является крайняя точка  $B$  с координатами  $x_1^* = -3$ ,  $x_2^* = 10$  и значением  $f_{\min}(x^*) = -325$ . Это решение является оптимальным решением задачи.

## 6.2. Необходимые условия минимума в терминах направлений

В задачах минимизации с ограничениями значительно труднее сформулировать условия оптимальности и они, как правило, сложнее, чем в случае безусловной оптимизации (см. разд. 5.1). Ниже формулируем такие условия для некоторых видов задач математического программирования.

Рассмотрим задачу математического программирования

$$f(x) \rightarrow \min, \quad x \in U \subseteq R^n, \quad (6.6)$$

где допустимое множество  $U$  выпукло и задается тем или иным способом (см. разд. 6.1). Для точек множества  $U$  введем следующие понятия направлений, учитывающих локальные свойства множества  $U$  и целевой функции  $f(x)$ .

**Определение 6.1.** Вектор  $p \in R^n$  задает в точке  $x^0 \in R^n$  *возможное направление*, если  $x^0 + \alpha p \in U$  при всех достаточно малых  $\alpha > 0$ .

Если  $x^{(0)}$  – внутренняя точка множества  $U$  то, очевидно, возможным является любое направление  $p \in R^n$ . Для граничной точки  $x^{(0)}$  множества  $U$  возможные направления определяются видом границы  $U$  в окрестности этой точки.

**Пример 6.3.** Точка  $x^{(0)}$  лежит на границе множества  $U$ , определяемого неравенством  $\langle a, x \rangle \leq b$ ,  $x \in R^n$ , т.е.  $x^{(0)}$  принадлежит гиперплоскости  $\langle a, x \rangle = b$ , нормальный вектор  $a$  которой направлен из полупространства  $U$ . Тогда возможные направления  $p$  удовлетворяют условию  $\langle a, x \rangle \leq 0$ .

**Пример 6.4.** Множество  $U \subset R^n$ , на границе которого лежит точка  $x^{(0)}$ , задано неравенством  $g(x) \leq 0$ , т.е.  $g(x^{(0)}) = 0$ . Здесь  $g(x)$  – нелинейная выпуклая дифференцируемая функция, причем

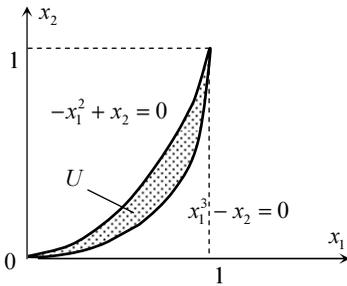


Рис. 6.2

предполагается, что  $g'(x^{(0)}) \neq 0$ .

Градиент  $g'(x^{(0)})$  является нормальным вектором гиперплоскости, касательной к гиперповерхности  $g(x) = 0$  в точке  $x^{(0)}$  и направлен из множества  $U$ . Поэтому возможные направления  $p$  в точке  $x^{(0)}$  должны удовлетворять неравенству  $\langle g'(x^{(0)}), p \rangle < 0$ . Для линейной функции  $g(x)$  это нера-

венство следует записывать как нестрогое.

В граничных точках невыпуклого множества  $U$  возможные направления могут и не существовать. Например, пусть множество задано системой неравенств:

$$\begin{cases} -x_1^2 + x_2 \leq 0, \\ x_1^3 - x_2 \leq 0, \\ 0 \leq x_1 \leq 1. \end{cases}$$

Нетрудно убедиться, что в граничной точке  $x^{(0)} = (0, 0)$  нет ни одного возможного направления (рис. 6.2).

Для любой точки  $x^{(0)}$  выпуклого множества  $U$  всегда можно указать хотя бы одно возможное направление. Пусть, например,  $x \in U$ ,  $x \neq x^{(0)}$ . Тогда  $p = x - x^{(0)}$  – возможное направление. В самом деле,  $x^{(0)} + \alpha p = x^{(0)} + \alpha(x - x^{(0)}) = \alpha x + (1 - \alpha)x^{(0)} \in U$  при всех  $\alpha \in [0, 1]$ , так как множество  $U$  – выпукло.

**Определение 6.2.** Вектор  $p \in R^n$  задает *направление убывания* функции  $f(x)$  в точке  $x^{(0)} \in U$ , если  $f(x^{(0)} + \alpha p) < f(x^{(0)})$  для всех достаточно малых  $\alpha > 0$ .

Рис. 6.3 поясняет определения 6.1 и 6.2. В точке  $x^{(1)}$  показаны возможные направления убывания, в точке  $x^{(2)}$  – возможные направления, не являющиеся направлениями убывания, в точке  $x^{(3)}$  – направления убывания, не являющиеся возможными.

Если функция  $f(x)$  дифференцируема в точке  $x^{(0)}$  и вектор  $p$  удовлетворяет условию

$$\langle f'(x^{(0)}), p \rangle < 0, \quad (6.7)$$

то он задает направление убывания функции  $f(x)$  в точке  $x^{(0)}$ . В самом деле, запишем приращение  $f(x)$  в точке  $x^{(0)}$ :  $f(x^{(0)} + \alpha p) - f(x^{(0)}) = \langle f'(x^{(0)}), \alpha p \rangle + o(\alpha) = \alpha (\langle f'(x^{(0)}), p \rangle + o(\alpha)/\alpha) < 0$ , при всех достаточно малых  $\alpha > 0$ . Геометрически условие (6.7) означает, что вектор  $p$  составляет тупой угол с градиентом  $f'(x^{(0)})$ .

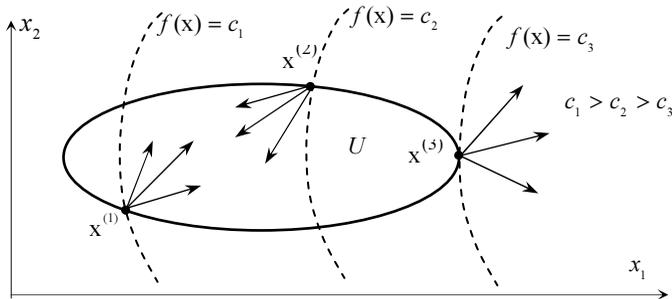


Рис. 6.3

Сформулируем с использованием введенных выше понятий необходимое условие локального минимума в задаче (6.6).

**Теорема 6.1.** Если  $x^*$  – точка локального минимума в задаче (6.6), то среди возможных направлений  $p \in R^n$  из этой точки ни одно не является направлением убывания целевой функции  $f(x)$ .

**Доказательство.** Предположим противное, тогда существует вектор  $p \in R^n$  такой, что при всех достаточно малых  $\alpha > 0$  точка

$x = x^* + \alpha r \in U$  и  $f(x) < f(x^*)$ , что противоречит определению локального минимума.

**Теорема 6.2.** Пусть в задаче (6.6) допустимое множество  $U$  является выпуклым, а целевая функция  $f(x)$  дифференцируема в точке  $x^* \in U$ . Тогда

1) если  $x^*$  – точка локального минимума в задаче (6.6), то

$$\langle f'(x^*), x - x^* \rangle \geq 0 \text{ при всех } x \in U \quad (6.8)$$

2) если функция  $f(x)$  выпукла на  $U$  и выполняется условие (6.8), то  $x^*$  является точкой глобального минимума функции  $f(x)$  на множестве  $U$ .

**Доказательство.** Докажем сначала первое утверждение. Предположим противное, т.е. что для некоторой точки  $x \in U$  выполняется неравенство  $\langle f'(x^*), x - x^* \rangle < 0$ . Так как множество  $U$  выпукло и точки  $x^*, x \in U$ , то направление  $p = x - x^*$  является возможным. С другой стороны, это направление является направлением убывания (6.7). Но это противоречит теореме 6.1. Указанное противоречие и доказывает утверждение 1.

Утверждение 2 следует непосредственно из свойства выпуклой функции  $f(x)$  на выпуклом множестве  $U: f(x) - f(x^*) \geq \langle f'(x^*), x - x^* \rangle$  для всех  $x \in U$ .

**Следствие.** Если  $x^*$  – внутренняя точка множества  $U$ , то неравенство (6.8) эквивалентно условию

$$f'(x^*) = 0. \quad (6.9)$$

**Доказательство.** Так как  $x^*$  – внутренняя точка множества  $U$ , то в ней любое направление  $p$  является возможным. Выберем  $p = -f'(x^*)$ . При достаточно малом  $\alpha > 0$  точка  $x = x^* - \alpha f'(x^*) \in U$ . Подставляя это выражение для  $x$  в (6.8), убеждаемся в справедливости равенства (6.9).

Обратно, при  $f'(x^*) = 0$ , очевидно, выполняется условие (6.8).

В некоторых случаях, приступая к решению задачи математического программирования, целесообразно найти сначала точку безусловного минимума целевой функции, и если она принадлежит допус-



$g_i(x^* + \Delta x) = 0, i = \overline{1, m}$ . Для таких приращений с точностью до бесконечно малых более высокого порядка, чем  $\|\Delta x\|$ , имеем:

$$\begin{aligned} \Delta g_i(x^*) &= g_i(x^* + \Delta x) - g_i(x^*) = dg_i(x^*) = \\ &= \sum_{j=1}^n \frac{\partial g_i(x^*)}{\partial x_j} \Delta x_j = 0, \quad i = \overline{1, m}. \end{aligned} \quad (6.12)$$

Из равенств (6.12) видно, что независимых приращений  $\Delta x_j$  существует только  $n - m$ , а остальные могут быть выражены через них. Для определенности будем считать приращения  $\Delta x_j$ , с индексами  $j = \overline{1, m}$  зависимыми, а с индексами  $j = \overline{m+1, n}$  – независимыми.

Для рассматриваемых приращений  $\Delta x$  разность  $f(x^* + \Delta x) - f(x^*) \geq 0$ , так как  $x^*$  – точка минимума  $f(x)$  на множестве  $U$ . Это возможно только при выполнении условия

$$\sum_{j=1}^n \frac{\partial f(x^*)}{\partial x_j} \Delta x_j = 0. \quad (6.13)$$

Из равенств (6.12) и (6.13) следует, что

$$\sum_{j=1}^n \left( \frac{\partial f(x^*)}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial g_i(x^*)}{\partial x_j} \right) \Delta x_j = 0,$$

где  $\lambda_i$  – множители Лагранжа.

Поскольку приращения  $\Delta x_j$  при  $j = \overline{m+1, n}$  произвольны и независимы, из последнего равенства следует, что выражения в скобках для таких  $j$  равны нулю. Выберем множители Лагранжа  $\lambda_i$  так, чтобы выражения в скобках обращались в нуль и при  $j = \overline{1, m}$ . Тогда для определения решения получим систему из  $n + m$  уравнений:

$$\frac{\partial f(x^*)}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial g_i(x^*)}{\partial x_j} = 0, \quad j = \overline{1, n}; \quad (6.14)$$

$$g_i(x^*) = 0, \quad i = \overline{1, m}, \quad (6.15)$$

содержащую  $n + m$  неизвестных  $x_1^*, \dots, x_n^*, \lambda_1, \dots, \lambda_m$ .

Соотношения (6.14) – (6.15) являются необходимыми условиями минимума функции  $f(x)$  из (6.10) на множестве  $U$ , заданном огра-

ничениями (6.11). С другой стороны, они определяют стационарные точки так называемой *функции Лагранжа* задачи (6.14) – (6.15):

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x),$$

потому что равенства (6.14) можно интерпретировать как  $\frac{\partial L}{\partial x_j} = 0$ , а

равенства (6.15) – как  $\frac{\partial L}{\partial \lambda_i} = 0$ .

Сформулируем теорему о необходимых условиях локального условного экстремума в задаче (6.10), (6.11), известную как *правило множителей Лагранжа*.

**Теорема 6.3.** Пусть в задаче (6.10) и (6.11) функции  $f(x)$  и  $g_i(x)$ ,  $i = \overline{1, m}$  дифференцируемы в  $R^n$ . Тогда если точка  $x^*$  есть решение этой задачи и градиенты  $g'_i(x^*)$  линейно независимы, существует вектор  $\lambda^* = (\lambda_1^*, \dots, \lambda_m^*)$  такой, что:

$$f'(x^*) + \sum_{i=1}^m \lambda_i^* g'_i(x^*) = 0. \quad (6.16)$$

Условие (6.16) эквивалентно равенствам (6.14) и имеет простой геометрический смысл: в точке  $x^*$  антиградиент целевой функции  $-f'(x^*)$  является линейной комбинацией градиентов  $g'_i(x^*)$  (нормалей к поверхностям  $g_i(x) = 0$ ). На рис. 6.4 приведена иллюстрация этого факта для задачи (6.10) – (6.11) при  $n = 2, m = 1$ . Пунктиром изображены линии уровня целевой функции  $f(x) = c$ . В точке  $x^*$  векторы  $f'(x^*)$  и  $g'(x^*)$  коллинеарны. В точках  $x$  из достаточно малой окрестности точки  $x^*$ , удовлетворяющих условию  $g(x) = 0$ , выполняется неравенство  $f(x) \geq f(x^*)$ . Очевидно,  $x^{(1)}$  является точкой локального условного минимума, а  $x^{(2)}$  – локального максимума  $f(x)$  при ограничении  $g(x) = 0$ . Стационарная точка функции Лагранжа  $x^{(3)}$  не является точкой локального условного экстремума, так как из нее можно сместиться, оставаясь на линии  $\{x \mid g(x) = 0\}$  так, что значение функции  $f(x)$  изменится в ту или иную сторону по сравнению с  $f(x^{(3)})$ .

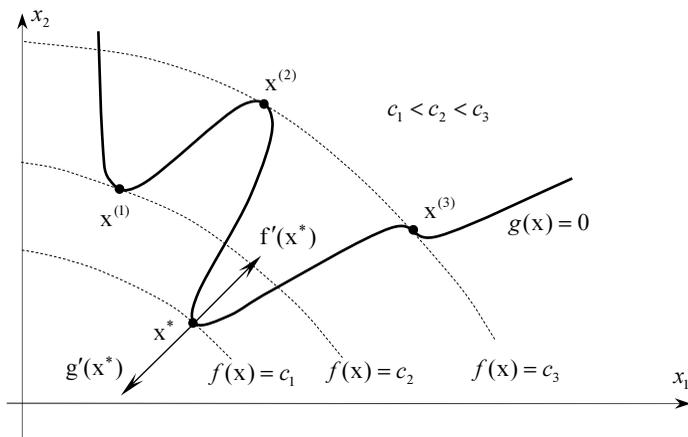


Рис.6.4

Отметим, что на допустимом множестве  $U$  задачи (6.10), (6.11) выполняются условия  $g_i(x) = 0$ ,  $i = \overline{1, m}$ , поэтому  $L(x, \lambda) = f(x)$  для

всех  $x \in U$ . Пусть  $x^*$  – решение задачи (6.10) – (6.11), а  $\lambda_i^*$  – множители Лагранжа, удовлетворяющие совместно с  $x_j^*$  системе уравнений (6.14), (6.15). Тогда

$$L(x^*, \lambda^*) = f(x^*) = \min_{x \in U} L(x, \lambda^*).$$

С учетом равенства (6.13) знак приращения  $\Delta L = L(x, \lambda^*) - L(x^*, \lambda^*)$  функции Лагранжа в точке  $x^*$  при фиксированном  $\lambda = \lambda^*$  и малых приращениях  $\Delta x_j$  определяется ее вторым дифференциалом

$$d^2 L(x^*, \lambda^*) = \sum_{j=1}^n \sum_{k=1}^n \frac{\partial^2 L(x^*, \lambda^*)}{\partial x_j \partial x_k} dx_j dx_k + \sum_{j=1}^n \frac{\partial^2 L(x^*, \lambda^*)}{\partial x_j^2} d^2 x_j.$$

Здесь, вообще говоря,  $d^2 x_j \neq 0$ , так как дифференциалы  $dx_j$  зависимы (см. (6.12)), однако с учетом (6.14) последняя сумма обращается в нуль, поэтому второй дифференциал функции Лагранжа в точке  $x^*$  при  $\lambda = \lambda^*$  является квадратичной формой дифференциалов  $dx_j$  и не зависит от вторых дифференциалов  $d^2 x_j$ .

Сформулируем достаточные условия оптимальности в задаче (6.10) – (6.11) с использованием матрицы вторых производных по переменным  $x_j$  функции Лагранжа:

$$L''_{xx}(x, \lambda^*) = f''(x) + \sum_{i=1}^m \lambda_i^* g_i''(x).$$

**Теорема 6.4.** Пусть функции  $f(x)$  и  $g_i(x)$ ,  $i = \overline{1, m}$  задачи (6.10) – (6.11) дважды дифференцируемы в  $R^n$ , точка  $(x^*, \lambda^*)$  удовлетворяет системе уравнений (6.14), (6.15), градиенты  $g'_i(x^*)$ ,  $i = \overline{1, m}$  линейно независимы и квадратичная форма  $d^2L(x^*, \lambda^*) = \langle L''_{xx}(x^*, \lambda^*) dx, dx \rangle$  положительно определена при всех приращениях  $dx$ , удовлетворяющих системе (6.12). Тогда  $x^*$  является точкой локального минимума функции  $f(x)$  из (6.10) на множестве  $U$ , определенном ограничениями (6.11).

**Замечание.** Если матрица  $L''_{xx}(x^*, \lambda^*)$  положительно определена, то квадратичная форма  $\langle L''_{xx}(x^*, \lambda^*) dx, dx \rangle > 0$  при всех ненулевых векторах  $dx$  и проверка достаточного условия локального условного минимума значительно упрощается. В противном случае для такой проверки приходится из системы уравнений (6.12) выражать зависимые приращения  $dx_j$  и подставлять их в квадратичную форму для исследования ее положительной определенности.

Пусть в задаче (6.10), (6.11) функция  $f(x)$  ограничена снизу на множестве  $U$ .

**Алгоритм метода множителей Лагранжа** решения задачи (6.10), (6.11):

1. Составить функцию Лагранжа  $L(x, \lambda)$ .
2. Найти частные производные от  $L(x, \lambda)$  по переменным  $x_j$ ,  $\lambda_i$  и приравнять их к нулю.
3. Решить полученную систему уравнений, т.е. найти стационарные точки функции Лагранжа.
4. С помощью достаточного условия (см. теорему 6.4) отобрать среди найденных стационарных точек точки локального условного минимума.

5. Сравнить значения функции в точках локального условного минимума и найти точку глобального минимума.

Замечание. Система уравнений (6.14), (6.15) представляет собой необходимые условия минимума функции Лагранжа  $L(x, \lambda)$  в пространстве  $R^{n+m}$ . Поэтому ее решения на шаге 3 можно искать численными методами, например, методом Ньютона (разд. 5.5), минимизируя  $L(x, \lambda)$ .

**Пример 6.5.** Решить задачу методом множителей Лагранжа:

$$f(x) = (x_1 - 2)^2 + (x_2 - 2)^2 \rightarrow \min ;$$

$$x_1^2 + x_2^2 = 2 .$$

**Решение.** Используя алгоритм метода множителей Лагранжа, последовательно получим:

1.  $L(x, \lambda) = (x_1 - 2)^2 + (x_2 - 2)^2 + \lambda(x_1^2 + x_2^2 - 2) \rightarrow \min .$

$$2. \begin{cases} \frac{\partial L}{\partial x_1} = 2(x_1 - 2) + 2\lambda x_1 = 0, \\ \frac{\partial L}{\partial x_2} = 2(x_2 - 2) + 2\lambda x_2 = 0, \\ \frac{\partial L}{\partial \lambda} = x_1^2 + x_2^2 - 2 = 0. \end{cases}$$

3. Система уравнений шага 2 имеет два решения:

$$(1) x_1^{(1)} = x_2^{(1)} = 1; \quad \lambda^{(1)} = 1, \quad (2) x_1^{(2)} = x_2^{(2)} = -1; \quad \lambda^{(2)} = -3.$$

4. Составим матрицу вторых производных функции Лагранжа:

$$L''_{xx}(x, \lambda) = \begin{pmatrix} 2(1 + \lambda) & 0 \\ 0 & 2(1 + \lambda) \end{pmatrix}.$$

Для первого решения матрица  $L''_{xx}$  положительно определена и, следовательно,  $x^{(1)} = (1, 1)$  – точка локального условного минимума.

Для второго решения требуется дополнительно учесть зависимость между приращениями  $2x_1 dx_1 + 2x_2 dx_2 = 0$  и исследовать квадратичную форму  $\langle L''_{xx}(x^*, \lambda^*) dx, dx \rangle = 2(1 + \lambda^{(2)}) dx_1^2 + 2(1 + \lambda^{(2)}) dx_2^2$ . Для  $x = x^{(2)} = (-1, -1)$ ,  $\lambda = \lambda^{(2)} = -3$  имеем:  $\langle L''_{xx}(x^*, \lambda^*) dx, dx \rangle = -8 dx_1^2 < 0$  при всех  $dx_1 \neq 0$ . Следовательно, вторая стационарная точка функции

Лагранжа определяет точку  $x^{(2)}$  локального условного максимума задачи.

5. Окончательно находим:  $x^* = (1, 1)$  и  $f_{\min} = 2$ .

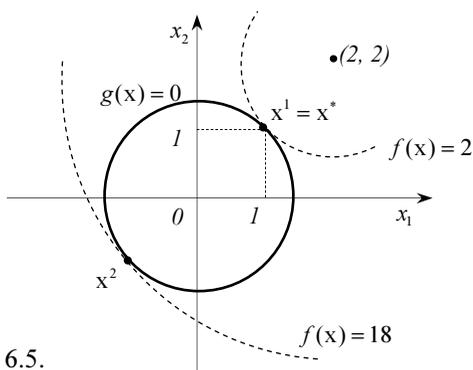


Рис. 6.5.

На рис. 6.5 представлено графическое решение этой задачи.

З а м е ч а н и е. Покажем, что множитель Лагранжа  $\lambda_i^*$  характеризует чувствительность оптимального значения  $f^*$  в задаче (6.10), (6.11) к изменениям правой части  $b_i$  в  $i$ -м ограничении (6.11). Рассмотрим задачу условной минимизации в пространстве  $R^2$  с одним ограничением:

$$\begin{aligned} f(x_1, x_2) &\rightarrow \min, \\ g_1(x_1, x_2) &= b_1. \end{aligned}$$

Пусть стационарная точка  $(x^*, \lambda^*)$  функции Лагранжа  $L(x, \lambda) = f(x) + \lambda(g_1(x) - b_1)$  является решением этой задачи. Так как  $g(x^*) = b_1$  и  $f^* = L(x^*, \lambda^*) = f(x^*)$  то, очевидно, оптимальные значения  $x^*$ ,  $\lambda^*$  и  $f^*$  связаны функциональной зависимостью с величиной  $b_1$ . Выразим частную производную  $\partial f^* / \partial b_1$  помощью правила дифференцирования сложной функции:

$$\frac{\partial f^*}{\partial b_1} = \frac{\partial f^*}{\partial x_1} \frac{\partial x_1}{\partial b_1} + \frac{\partial f^*}{\partial x_2} \frac{\partial x_2}{\partial b_1}. \quad (6.17)$$

С другой стороны, из условия  $g_1(x_1, x_2) = b_1$  получаем:

$$\frac{\partial g_1}{\partial x_1} \frac{\partial x_1}{\partial b_1} + \frac{\partial g_1}{\partial x_2} \frac{\partial x_2}{\partial b_1} - 1 = 0. \quad (6.18)$$

Умножим обе части равенства (6.18) на  $\lambda^*$  и прибавим к (6.17):

$$\frac{\partial f^*}{\partial b_1} = -\lambda^* + \sum_{j=1}^2 \left( \frac{\partial f^*}{\partial x_j} + \lambda^* \frac{\partial g_1}{\partial x_j} \right) \frac{\partial x_j}{\partial b_1}.$$

Так как  $(x^*, \lambda^*)$  – стационарная точка функций Лагранжа, то с учетом (6.14) получаем:  $\partial f^*/\partial b_1 = -\lambda^*$ . Таким образом, скорость изменения оптимального значения  $f^*$ , вызываемого изменением правой части в ограничении  $g_1(x_1, x_2) = b_1$ , определяется значением множителя Лагранжа. Для задачи на условный экстремум с  $m$  ограничениями  $g_i(x) = b_i$ ,  $i = \overline{1, m}$  аналогично можно получить:  $\frac{\partial f^*}{\partial b_i} = -\lambda_i^*$ .

#### 6.4. Условия оптимальности в выпуклом программировании

Рассмотрим гладкую задачу выпуклого программирования, т.е.

$$f(x) \rightarrow \min, \quad x \in R^n; \quad (6.19)$$

$$g_i(x) \leq 0, \quad i = \overline{1, m}, \quad (6.20)$$

где  $f(x)$  и  $g_i(x)$  – выпуклые дифференцируемые функции в  $R^n$ .

Пусть допустимое множество  $U$  удовлетворяет так называемому условию регулярности (условию Слейтера): существует точка  $x^{(0)} \in R^n$ , такая, что  $g_i(x^{(0)}) < 0$  при всех  $i = \overline{1, m}$ . Это означает, что у множества  $U$  есть хотя бы одна внутренняя точка.

Если в некоторой точке  $x^{(0)} \in U$  выполняется неравенство  $g_i(x^{(0)}) < 0$ , то ограничение (6.20) с номером  $i$  называется *пассивным* в точке  $x^{(0)}$ . Очевидно, для внутренней точки допустимого множества  $U$  все ограничения (6.20) оказываются пассивными. А в граничной точке  $x^{(0)} \in U$  хотя бы одно из ограничений (6.20) обращается в равенство:  $g_i(x^{(0)}) = 0$ . Такое ограничение называется *активным* в точке  $x^{(0)}$ . Пусть  $I(x^{(0)})$  множество индексов активных ограничений в точке  $x^{(0)}$ :

$$I(x^{(0)}) = \{ i \mid g_i(x^{(0)}) = 0, \quad i = \overline{1, m} \}.$$

З а м е ч а н и е . Условие регулярности допустимого множества  $U$  можно сформулировать и в иной форме – как линейную независимость в каждой граничной точке  $x^{(0)}$  множества  $U$  градиентов  $g'_i(x^{(0)})$  (внешних нормалей к поверхностям  $g_i(x) = 0$ ) для всех  $i \in I(x^{(0)})$  (предполагается, что в этих точках отсутствуют избыточные активные ограничения (6.20)).

Введем дополнительные переменные  $z_i, i = \overline{1, m}$  и перейдем от ограничений-неравенств (6.20) к ограничениям-равенствам:

$$g_i(x) + z_i^2 = 0, \quad i = \overline{1, m}. \quad (6.21)$$

Запишем функцию Лагранжа задачи (6.19), (6.21):

$$\overline{L}(x, \lambda, z) = f(x) + \sum_{i=1}^m \lambda_i (g_i(x) + z_i^2). \quad (6.22)$$

Система уравнений для ее стационарных точек имеет вид:

$$\frac{\partial \overline{L}}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial g_i}{\partial x_j} = 0, \quad j = \overline{1, n}; \quad (6.23)$$

$$\frac{\partial \overline{L}}{\partial z_i} = 2 \lambda_i z_i = 0, \quad i = \overline{1, m}; \quad (6.24)$$

$$\frac{\partial \overline{L}}{\partial \lambda_i} = g_i(x) + z_i^2 = 0, \quad i = \overline{1, m}. \quad (6.25)$$

Условия (6.23) – (6.25) являются необходимыми условиями минимума в задаче (6.19), (6.21) (теорем 6.3).

Исключим из этой системы вспомогательные переменные  $z_i$ . Очевидно, равенства (6.25) с величинами  $z_i^2 \geq 0$  эквивалентны неравенствам (6.20). Далее, умножив каждое равенство из (6.24) на  $z_i/2$ , получим:  $\lambda_i z_i^2 = 0$  или (см. (6.25))

$$\lambda_i g_i(x) = 0. \quad (6.26)$$

Составим функцию Лагранжа для задачи (6.19), (6.20):

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x).$$

С учетом соотношений (6.23), (6.20) и (6.26) необходимые условия минимума в задаче (6.19), (6.20) принимают вид:

$$\frac{\partial L(x, \lambda)}{\partial x_j} = \frac{\partial f(x)}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial g_i(x)}{\partial x_j} = 0, \quad j = \overline{1, n}; \quad (6.27)$$

$$\frac{\partial L(x, \lambda)}{\partial \lambda_i} = g_i(x) \leq 0, \quad i = \overline{1, m}; \quad (6.28)$$

$$\lambda_i g_i(x) = 0, \quad i = \overline{1, m}. \quad (6.29)$$

Условие (6.29) означает, что в искомой точке  $x^*$  хотя бы один из сомножителей обращается в нуль. Если  $\lambda_i \neq 0$ , то  $g_i(x^*) = 0$  (ограничение с номером  $i$  является активным). Если в точке  $x^*$  имеет место  $g_i(x^*) < 0$  (пассивное ограничение), то  $\lambda_i = 0$ . Условия (6.27) можно записать в векторной форме:

$$-f'(x) = \sum_{i=1}^m \lambda_i g'_i(x), \quad (6.30)$$

откуда следует, что антиградиент  $-f'(x^*)$  в точке минимума  $x^*$  является линейной комбинацией внешних нормалей к активным для точки  $x^*$  ограничениям.

**Теорема 6.5.** Пусть  $(x^*, \lambda^*)$  – решение системы (6.27) - (6.29). Тогда если точка  $x^*$  является решением задачи (6.19), (6.20), то  $\lambda_i^* \geq 0$  для всех  $i = \overline{1, m}$ .

*Доказательство.* Пусть  $x^*$  – внутренняя точка множества  $U$ , т.е.  $g_i(x^*) < 0$  для всех  $i = \overline{1, m}$ . Тогда из (6.29) следует, что все  $\lambda_i^* = 0$ .

Пусть теперь  $x^*$  – граничная точка множества  $U$ , т.е. часть ограничений (6.20) обращается в этой точке в равенства. Введем множество индексов  $I(x^*)$ , соответствующих активным в точке  $x^*$  ограничениям:  $I(x^*) = \{i = \overline{1, m} \mid g_i(x^*) = 0\}$ . Будем рассматривать активные ограничения  $g_i(x^*) = 0$ ,  $i \in I(x^*)$  как частный случай ограничений более общего вида  $g_i(x^*) \leq c_i$ , соответствующий  $c_i = 0$ . Очевидно, при увеличении  $c_i$ , допустимое множество  $U$  расширяется, и значение  $f^* = \min_U f(x)$  может только уменьшиться. Поэтому

$$\frac{\partial f^*}{\partial c_i} \leq 0, \quad i \in I(x^*). \quad (6.31)$$

Рассмотрим функцию Лагранжа в точке  $(x^*, \lambda^*)$ . Так как  $g_i(x^*) = c_i$  для  $i \in I(x^*)$  и  $\lambda_i^* = 0$  для  $i \notin I(x^*)$ , можно записать:

$$\begin{aligned} L(x^*, \lambda^*) &= f(x^*) + \sum_{i \in I(x^*)} \lambda_i^* g_i(x^*) + \sum_{i \notin I(x^*)} \lambda_i^* g_i(x^*) = \\ &= f(x^*) + \sum_{i \in I(x^*)} \lambda_i^* c_i. \end{aligned}$$

Поэтому

$$\frac{\partial L}{\partial c_i} = \frac{\partial f^*}{\partial c_i} + \lambda_i^*. \quad (6.32)$$

С другой стороны, по правилу дифференцирования сложной функции

$$\begin{aligned} \frac{\partial L}{\partial c_i} &= \sum_{j=1}^n \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial c_i} + \sum_{i=1}^m \lambda_i^* \sum_{j=1}^n \frac{\partial g_i}{\partial x_j} \frac{\partial x_j}{\partial c_i} = \\ &= \sum_{j=1}^n \left( \frac{\partial f}{\partial x_j} + \sum_{i=1}^m \lambda_i^* \frac{\partial g_i}{\partial x_j} \right) \frac{\partial x_j}{\partial c_i} = 0, \end{aligned}$$

так как в силу условий (6.27) выражения в круглых скобках в последнем равенстве обращаются в нуль. Отсюда с учетом соотношения (6.32) имеем:

$$\frac{\partial f^*}{\partial c_i} + \lambda_i^* = 0, \quad i \in I(x^*),$$

и так как  $\frac{\partial f^*}{\partial c_i} \leq 0$  (см. (6.31)), получаем, что  $\lambda_i^* \geq 0$ . Для пассивных

ограничений из равенства (6.29) следует, что  $\lambda_i^* = 0$ ,  $i \in I(x^*)$ . Таким образом  $\lambda_i^* \geq 0$  при всех  $i = \overline{1, m}$ .

Условие неотрицательности множителей Лагранжа  $\lambda_i^*$  имеет следующий геометрический смысл. Если  $f'(x^*) \neq 0$ , то  $x^*$  – граничная точка множества  $U$ . Пусть для нее активными являются ограничения  $g_i(x^*) \leq 0$  с номерами  $i \in I(x^*)$ . Тогда антиградиент  $-f'(x^*)$  с учетом (6.30) представляется в виде линейной комбинации градиентов

(внешних нормалей)  $g'_i(x^*)$ ,  $i \in I(x^*)$  с неотрицательными коэффициентами  $\lambda_i^*$  (рис. 6.6).

Итак, с учетом теоремы 6.5 и формул (6.27) – (6.29) можно сформулировать следующие необходимые условия минимума в задаче (6.19), (6.20) с допустимым множеством, удовлетворяющим условию регуляр-

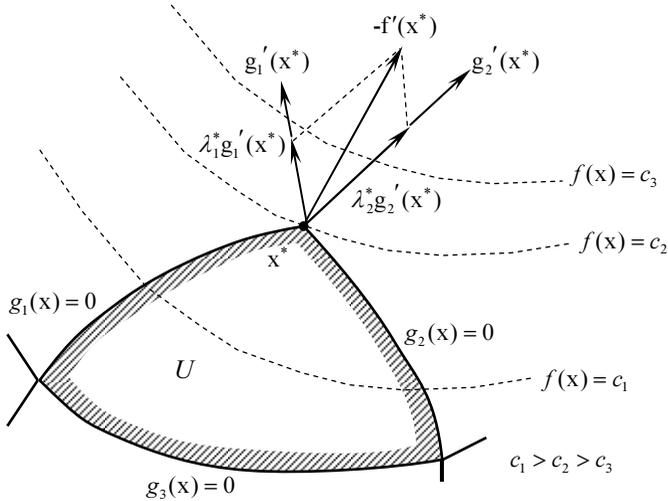


Рис. 6.6

ности: если  $x^*$  является решением задачи (6.19), (6.20), то для некоторых чисел  $\lambda_i^*$ ,  $i = \overline{1, m}$  выполняются соотношения

$$\lambda_i^* \geq 0, \quad i = \overline{1, m}; \quad (6.33)$$

$$\frac{\partial f(x^*)}{\partial x_j} + \sum_{i=1}^m \lambda_i^* \frac{\partial g_i(x^*)}{\partial x_j} = 0, \quad j = \overline{1, n}; \quad (6.34)$$

$$\lambda_i^* g_i(x^*) = 0, \quad i = \overline{1, m}; \quad (6.35)$$

$$g_i(x^*) \leq 0, \quad i = \overline{1, m}, \quad (6.36)$$

которые называются условиями Куна – Таккера.

Эти условия являются также и достаточными условиями минимума в задаче (6.19), (6.20).

**Теорема 6.6 (Куна – Таккера).** Для того чтобы точка  $x^* \in R^n$  была решением задачи выпуклого программирования (6.19) – (6.20) с

дифференцируемыми функциями  $f(x)$  и  $g_i(x)$ ,  $i = \overline{1, m}$ , достаточно (а если допустимое множество  $U$  удовлетворяет условию регулярности, то и необходимо), чтобы существовал вектор  $\lambda^* \in R^m$ , для которого выполняются условия (6.33) – (6.36).

**Доказательство.** Докажем достаточность условий теоремы. Так как функции  $f(x)$ ,  $g_i(x)$  выпуклы и дифференцируемы и  $\lambda_i^* \geq 0$ , то

функция Лагранжа  $L(x, \lambda^*) = f(x) + \sum_{i=1}^m \lambda_i^* g_i(x)$  также является выпуклой и дифференцируемой. Поэтому условия (6.34) необходимы и достаточны для того, чтобы  $L(x^*, \lambda^*) = \min_{x \in R^n} L(x, \lambda^*)$ , т.е.

$$\begin{aligned} L(x^*, \lambda^*) &= f(x^*) + \sum_{i=1}^m \lambda_i^* g_i(x^*) \leq \\ &\leq f(x) + \sum_{i=1}^m \lambda_i^* g_i(x) = L(x, \lambda^*) \end{aligned}$$

для всех  $x \in R^n$ . С учетом (6.35) получаем, что

$$f(x^*) \leq f(x) + \sum_{i=1}^m \lambda_i^* g_i(x).$$

А так как  $\lambda_i^* \geq 0$  и  $g_i(x^*) \leq 0$  для всех  $x \in U$ , то  $f(x^*) \leq f(x)$ ,  $x \in U$ , т.е. точка  $x^*$  – это решение задачи (6.19), (6.20).

**Пример 6.6.** Проверить, что точка  $x = (3, 1)$  является решением задачи

$$\begin{aligned} f(x) &= 3x_1^2 + 4x_1x_2 + 5x_2^2 \rightarrow \min ; \\ -x_1 - x_2 + 4 &\leq 0, \quad -x_1 \leq 0, \quad -x_2 \leq 0. \end{aligned}$$

**Решение.** Матрица вторых производных целевой функции

$$f''(x) = \begin{pmatrix} 6 & 4 \\ 4 & 10 \end{pmatrix}$$

положительно определена. Следовательно,  $f(x)$  выпукла. Очевидно, допустимое множество  $U$  выпукло.

Функция Лагранжа задачи имеет вид:

$$L(x, \lambda) = 3x_1^2 + 4x_1x_2 + 5x_2^2 + \lambda_1(-x_1 - x_2 + 4) + \lambda_2(-x_1) + \lambda_3(-x_2).$$

Выпишем условия Куна – Таккера (6.33) – (6.36):

$$\begin{aligned} \lambda_i &\geq 0, \quad i = 1, 2, 3; \\ 6x_1 + 4x_2 - \lambda_1 - \lambda_2 &= 0, \\ 4x_1 + 10x_2 - \lambda_1 - \lambda_3 &= 0, \\ \lambda_1(-x_1 - x_2 + 4) &= 0, \\ \lambda_2 x_1 = 0, \quad \lambda_3 x_2 &= 0, \\ -x_1 - x_2 + 4 &\leq 0, \\ -x_1 \leq 0, \quad -x_2 &\leq 0. \end{aligned}$$

При  $x = (3, 1)$  находим:  $\lambda_2 = \lambda_3 = 0$ ,  $\lambda_1 = 22$ . Так как условия Куна – Таккера выполнены, то  $x^* = (3, 1)$  – это решение поставленной задачи выпуклого программирования, т.е. точка минимума исходной функции.

Приведем условия оптимальности Куна - Таккера в задаче выпуклого программирования (6.19), (6.20) для недифференцируемых функций.

**Определение 6.3.** Точка  $(x^*, \lambda^*)$  называется *седловой точкой* функции Лагранжа задачи (6.19), (6.20), если

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*)$$

для всех  $x \in U$  и всех  $\lambda_i \geq 0$ ,  $i = \overline{1, m}$ .

*Замечание.* Можно показать, что необходимыми и достаточными для седловой точки функции Лагранжа задачи (6.19), (6.20) являются следующие условия:

$$L(x^*, \lambda^*) = \min_{x \in R^n} L(x, \lambda^*);$$

$$\lambda_i^* g_i(x^*) = 0, \quad i = \overline{1, m}.$$

**Теорема 6.7** (Куна-Таккера – о седловой точке функции Лагранжа). Для того чтобы точка  $x^* \in R^n$  была решением задачи выпуклого программирования (6.19), (6.20), допустимое множество которой удовлетворяет условию регулярности, необходимо и достаточно, чтобы существовал вектор  $\lambda^* \in R^m$ ,  $\lambda_i^* \geq 0$ ,  $i = \overline{1, m}$ , такой, что  $(x^*, \lambda^*)$  – седловая точка функции Лагранжа этой задачи.

## 6.5. Задача математического программирования со смешанными ограничениями

Рассмотрим гладкую задачу математического программирования

$$f(x) \rightarrow \min, \quad x \in R^n; \quad (6.37)$$

$$g_i(x) \leq 0, \quad i = 1, 2, \dots, p; \quad (6.38)$$

$$h_i(x) = 0, \quad i = p+1, \dots, m, \quad (6.39)$$

где все функции  $f(x)$ ,  $g_i(x)$ ,  $h_i(x)$  – дифференцируемы в  $R^n$ . Эта задача отличается от задачи (6.19), (6.20) наличием ограничений-равенств (6.39).

Потребуем, чтобы допустимое множество  $U$  удовлетворяло условию регулярности, т.е. чтобы в оптимальной точке  $x^*$  векторы градиентов  $g'_i(x^*)$  активных ограничений и  $h'_i(x^*)$  были линейно независимыми. Проверка этого условия затруднительна, так как обычно оптимальное решение задачи заранее неизвестно. Однако условие регулярности заведомо выполняется для задач (6.37) – (6.39) следующего вида:

- 1) все ограничения (6.38), (6.39) содержат линейные функции;
- 2) все ограничения-неравенства (6.38) содержат выпуклые функции и существует хотя бы одна точка  $x^{(0)} \in U$  такая, что  $g_i(x^{(0)}) < 0$ ,  $i = \overline{1, p}$ , а все ограничения-равенства (6.39) линейны.

Для функции Лагранжа этой задачи

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^p \lambda_i g_i(x) + \sum_{i=p+1}^m \mu_i h_i(x)$$

можно по аналогии со случаем, рассмотренным в разд. 6.4, получить условия Куна – Таккера

$$f'(x^*) + \sum_{i=1}^m \lambda_i^* g'_i(x^*) + \sum_{i=1}^m \mu_i^* h'_i(x^*) = 0; \quad (6.40)$$

$$g_i(x^*) \leq 0, \quad i = \overline{1, p}; \quad (6.41)$$

$$h_i(x^*) = 0, \quad i = \overline{p+1, m}; \quad (6.42)$$

$$\lambda_i^* g_i(x^*) = 0, \quad i = \overline{1, p}; \quad (6.43)$$

$$\lambda_i^* \geq 0, \quad i = \overline{1, p}. \quad (6.44)$$

Приведем формулировку теоремы о необходимых и достаточных условиях минимума для задачи (6.37) – (6.39).

**Теорема 6.8.** (Куна – Таккера). Пусть в задаче (6.37) – (6.39) функции  $f(x)$  и  $g_i(x)$  выпуклы и дифференцируемы, функции  $h_i(x)$  – линейны и допустимое множество  $U$  удовлетворяет условию регулярности. Тогда, для того чтобы точка  $x^*$  была решением этой задачи, необходимо и достаточно, чтобы существовали векторы  $\lambda^* = (\lambda_1^*, \dots, \lambda_p^*)$  и  $\mu^* = (\mu_{p+1}^*, \dots, \mu_m^*)$ , для которых выполняются условия (6.40) – (6.44).

Доказательство теоремы можно найти в [3].

Для решения задач условной оптимизации условия Куна–Таккера (6.40) – (6.44) применять, как правило, достаточно сложно. Поэтому рассмотрим пример на идентификацию точек с помощью этих условий.

**Пример 6.6.** Сделать заключение об оптимальности следующих точек  $x^{(1)} = (0, 0)$ ,  $x^{(2)} = (1, 1/2)$  и  $x^{(3)} = (1/3, -1/6)$  для следующей задачи

$$f(x) = x_1^2 + 2x_2^2 \rightarrow \min ;$$

$$\begin{cases} x_1^2 + x_2^2 \leq 5, \\ 2x_1 - 2x_2 = 1. \end{cases}$$

**Решение.** Функция Лагранжа задачи имеет вид:

$$L(x, \lambda, \mu) = x_1^2 + 2x_2^2 + \lambda(x_1^2 + x_2^2 - 5) + \mu(2x_1 - 2x_2 - 1).$$

Выпишем условия Куна – Таккера (6.40) – (6.44):

$$(1) \quad \frac{\partial L}{\partial x_1} = 2x_1 + 2\lambda x_1 + 2\mu = 0,$$

$$(2) \quad \frac{\partial L}{\partial x_2} = 4x_2 + 2\lambda x_2 - 2\mu = 0,$$

$$(3) \quad \lambda(x_1^2 + x_2^2 - 5) = 0,$$

$$(4) \quad x_1^2 + x_2^2 - 5 \leq 0,$$

$$(5) \quad 2x_1 - 2x_2 - 1 = 0,$$

$$(6) \quad \lambda \geq 0.$$

1. Проверим точку  $x^{(1)} = (0, 0)$ . Условие (5) не удовлетворяется, т.е. точка не принадлежит области допустимых решений (ОДР). Этого достаточно, чтобы сделать вывод о том, что точка  $x^{(1)}$  не является точкой экстремума.

2. Точка  $x^{(2)} = (1, 1/2)$  принадлежит ОДР, т.к. удовлетворяются условия (4) и (5). Из условия (3) следует:  $\lambda = 0$ , т.к.  $x_1^2 + x_2^2 - 5 \neq 0$ . Из условий (1) и (2) находим  $\mu = -1$  и  $\mu = 1$  соответственно. Следовательно, условия теоремы 6.8 не удовлетворяются и точка  $x^{(2)}$  не является точкой оптимального решения.

3. Точка  $x^{(3)} = (1/3, -1/6)$  принадлежит ОДР, т.к. удовлетворяются условия (4) и (5). Из условия (3) следует, что  $\lambda = 0$ . Тогда из условий (1) и (2) следует:  $\mu = -1/3$ . Следовательно, условия теоремы 6.8 удовлетворяются и точка  $x^{(3)}$  является *точкой минимума* целевой функции, значение которой  $f_{\min}(x^{(3)}) = 1/6$ .

## Глава 7

### МЕТОДЫ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Эффективные алгоритмы минимизации разработаны лишь для некоторых классов задач (например, квадратичного программирования).

Для задач нелинейного программирования общего вида используются численные методы. Обзору таких задач и методов их решения посвящена настоящая глава.

Решение некоторых классов задач нелинейного программирования сводится к решению задач линейного программирования. Ряд других классов задач – к решению задач безусловной минимизации.

#### 7.1. Задача квадратичного программирования

Требуется минимизировать выпуклую квадратичную функцию на допустимом множестве  $U$ , заданном линейными ограничениями-неравенствами, т.е.

$$f(x) = \frac{1}{2} \langle Ax, x \rangle + \langle b, x \rangle + c \rightarrow \min, \quad x \in R^n, \quad (7.1)$$

$$\sum_{j=1}^n \alpha_{ij} x_j \leq \beta_i, \quad i = \overline{1, m}, \quad (7.2)$$

$$x_j \geq 0, \quad j = \overline{1, n}. \quad (7.3)$$

Здесь  $A = \{a_{ij}\}$  – симметрическая положительно определенная матрица размера  $(n \times n)$ ;  $b \in R^n$  – вектор;  $c$  – число.

Так как целевая функция (7.1) выпукла, а ограничения (7.2), (7.3) линейны, то задача (7.1) – (7.3) является задачей выпуклого программирования (см. гл. 6). Ограничения (7.2) можно записать в виде  $g_i(x) =$

$$= \sum_{j=1}^n \alpha_{ij} x_j - \beta_i \leq 0, \text{ а ограничения (7.3) – в виде } h_j(x) = -x_j \leq 0.$$

Составим функцию Лагранжа этой задачи, введя множители Лагранжа  $\lambda_i$ ,  $i = \overline{1, m}$ , для ограничений (7.2) и  $\mu_j$ ,  $j = \overline{1, n}$  для ограничений (7.3):

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i \left( \sum_{j=1}^n \alpha_j^{(i)} x_j - \beta_i \right) + \sum_{j=1}^n \mu_j (-x_j).$$

Запишем условия Куна – Таккера (6.33) – (6.36) для задачи (7.1) – (7.3):

$$\frac{\partial L}{\partial x_j} = \sum_{i=1}^n a_{ij} x_j + b_j + \sum_{i=1}^n \lambda_i \alpha_j^{(i)} - \mu_j = 0, \quad j = \overline{1, n}; \quad (7.4)$$

$$\lambda_i g_i = \lambda_i \left( \sum_{j=1}^n \alpha_j^{(i)} x_j - \beta_i \right) = 0, \quad i = \overline{1, m}; \quad (7.5)$$

$$\mu_j h_j = -\mu_j x_j = 0, \quad j = \overline{1, n}; \quad (7.6)$$

$$\lambda_i \geq 0, \quad i = \overline{1, m}; \quad \mu_j \geq 0, \quad j = \overline{1, n}, \quad (7.7)$$

включающие в себя и ограничения (7.2) и (7.3) задачи

$$g_i(x) = \sum_{j=1}^n \alpha_j^{(i)} x_j - \beta_i \leq 0, \quad i = \overline{1, m}, \quad (7.8)$$

$$h_j(x) = -x_j \leq 0, \quad j = \overline{1, n}. \quad (7.9)$$

Введя дополнительные переменные  $x_{n+i} \geq 0$ , перейдем от неравенств (7.8) к равенствам  $\sum_{j=1}^n \alpha_j^{(i)} x_j - \beta_i + x_{n+i} = 0$ . Тогда соотношения (7.5) переписываются в виде  $\lambda_i x_{n+i} = 0$ .

Итак, условия Куна – Таккера (7.4) – (7.9) для задачи квадратичного программирования можно записать следующим образом:

$$\sum_{i=1}^n a_{ij} x_j + \sum_{i=1}^n \lambda_i \alpha_j^{(i)} - \mu_j = -b_j, \quad j = \overline{1, n}; \quad (7.10)$$

$$\sum_{j=1}^n \alpha_j^{(i)} x_j + x_{n+i} = \beta_i, \quad i = \overline{1, m}; \quad (7.11)$$

$$\lambda_i \geq 0, \quad i = \overline{1, m}; \quad \mu_j \geq 0, \quad j = \overline{1, n}; \quad (7.12)$$

$$x_k \geq 0, \quad k = \overline{1, n+m}; \quad (7.13)$$

$$\lambda_i x_{n+i} = 0, \quad i = \overline{1, m}; \quad \mu_j x_j = 0, \quad j = \overline{1, n}. \quad (7.14)$$

В соответствии с теоремой 6.6 (Куна – Таккера) решение  $x^* = (x_1^*, \dots, x_1^*)$  системы (7.10) – (7.14) является искомой точкой минимума функции  $f(x)$  на множестве  $U$ .

Отметим, что условия (7.10) – (7.14) содержат  $(n + m)$  линейных уравнений (7.10) и (7.11) с  $2(n + m)$  неотрицательными неизвестными  $x_k, \lambda_i, \mu_j$ , из которых, согласно условиям (7.14), по крайней мере  $(n + m)$  равны нулю.

Допустимое базисное решение этой системы может быть получено симплекс–методом с искусственным базисом или двойственным симплекс–методом. При этом следует учитывать условия (7.14), т.е. не включать в базисные переменные одновременно  $\lambda_i, x_{n+i}$  с одним и тем же номером  $i$  и переменные  $\mu_j, x_j$  с одинаковым номером  $j$ .

## 7.2. Методы возможных направлений

При построении вычислительных алгоритмов методов этой группы используются понятия возможного направления и направления убывания целевой функции для точек  $x$  допустимого множества  $U$  задачи выпуклого программирования:

$$f(x) \rightarrow \min, \quad x \in U \subset R^n,$$

где  $U$  – выпуклое множество.

Основу алгоритмов составляет итерационный процесс

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad k = 0, 1, \dots; \quad x^{(0)} \in U. \quad (7.15)$$

Вектор  $p^{(k)}$ , определяющий направление перемещения (7.15) из точки  $x^{(k)}$  в точку  $x^{(k+1)}$ , должен удовлетворять следующим двум требованиям:

1. Для достаточно малых  $\alpha_k > 0$  точка  $x^{(k+1)} \in U$ , т.е.  $p^{(k)}$  задает возможное *направление*.
2. Для достаточно малых  $\alpha_k > 0$  выполняется неравенство  $f(x^{(k+1)}) < f(x^{(k)})$ , т.е.  $p^{(k)}$  определяет *направление убывания* функции  $f(x)$ .

Первое условие означает, в частности, что для граничных точек  $x^{(k)}$  допустимого множества  $U$  вектор  $p^{(k)}$  не должен быть направлен за пределы допустимого множества.

Величина  $\alpha_k > 0$  в (7.15) выбирается, как правило, из условия исчерпывающего спуска в направлении  $p^{(k)}$  с учетом требования  $x^{(k+1)} \in U$ .

### Случай линейных ограничений

Рассмотрим задачу

$$f(x) \rightarrow \min, \quad x \in R^n; \quad (7.16)$$

$$\langle a^{(i)}, x \rangle = \sum_{j=1}^n a_j^{(i)} x_j \leq b_i, \quad i = \overline{1, m}; \quad (7.17)$$

$$x_j \geq 0, \quad j = \overline{1, n}, \quad (7.18)$$

где целевая функция  $f(x)$  выпукла и дифференцируема на выпуклом множестве  $U$ , заданном ограничениями (7.17) и (7.18).

Будем искать решение задачи (7.16) – (7.18) с помощью итерационной последовательности (7.15). Опишем сначала правило выбора вектора  $p^{(k)}$  для очередной точки  $x^{(k)}$ . Возможны два случая:

1. Точка  $x^{(k)}$  является внутренней точкой множества  $U$  (все неравенства (7.17) и (7.18) выполняются как строгие). Тогда полагают

$$p^{(k)} = -f'(x^{(k)}). \quad (7.19)$$

2. Пусть хотя бы одно из неравенств (7.17), (7.18) в точке  $x^{(k)}$  обращается в равенство, т.е.  $x^{(k)}$  является граничной точкой допустимого множества  $U$ . Тогда направление  $p^{(k)}$  из (7.19) не является, вообще говоря, возможным в точке  $x^{(k)}$  направлением, так как антиградиент  $-f'(x^{(k)})$  может быть направлен за пределы множества  $U$  (т.е. перемещение (7.15) с любым шагом  $\alpha_k > 0$  дает точку  $x^{(k+1)} \notin U$ ). Можно привести следующие соображения по выбору вектора  $p^{(k)}$  в этом случае: с одной стороны,  $p^{(k)}$  должен быть направлен внутрь множества  $U$  или вдоль его границы; с другой стороны,  $-p^{(k)}$  должен составлять с направлением наискорейшего убывания  $f(x)$ , т.е. с антиградиентом  $-f'(x^{(k)})$ , возможно меньший угол.

Поиск направления  $p^{(k)}$  с такими свойствами можно свести к решению задачи линейного программирования.

Обозначим через  $I_k$  и  $J_k$  множества индексов, соответствующих тем ограничениям (7.17) и (7.18), которые в точке  $x^{(k)}$  обращаются в равенства (активными ограничениями), т.е.

$$I_k = \left\{ i \mid \sum_{j=1}^n a_j^{(i)} x_j^{(k)} = b_i \right\}, \quad J_k = \left\{ j \mid x_j^{(k)} = 0 \right\}. \quad (7.20)$$

Соотношения (7.20) означают, что граничная точка  $x^{(k)}$  принадлежит гиперплоскостям

$$\langle a^{(i)}, x \rangle = b_i, \quad i \in I_k \quad \text{и} \quad \langle e^{(j)}, x \rangle = 0, \quad j \in J_k, \quad (7.21)$$

где  $e^{(j)}$  –  $j$ -й базисный вектор.

Покажем, что  $a^{(i)}$  и  $-e^{(j)}$  – нормальные векторы к указанным гиперплоскостям, направленные из допустимого множества  $U$ . В самом деле, гиперплоскость  $\langle a^{(i)}, x \rangle = b_i$  есть поверхность уровня функции  $\Phi_i(x) = \langle a^{(i)}, x \rangle$ , соответствующая значению  $b_i$  этой функции. Значит, градиент  $\Phi'_i(x^{(k)}) = a^{(i)}$  – это вектор, нормальный к данной гиперплоскости. Далее, вдоль направления  $\Phi'_i(x^{(k)})$  функция  $\Phi_i(x)$  возрастает, т.е. равенство (7.21) нарушается и переходит в неравенство противоположного, чем (7.17), знака. Следовательно, нормальный вектор  $a^{(i)} = \Phi'_i(x^{(k)})$  направлен из множества  $U$ . Аналогичные рассуждения можно провести и для вектора  $-e^{(j)}$ ,  $j \in J_k$ , записав ограничения (7.18) в виде  $-\langle e^{(j)}, x \rangle \leq 0$ .

Очевидно, направление вектора  $p^{(k)}$  из точки  $x^{(k)}$  по отношению к множеству  $U$  будет возможным, если он образует с соответствующими нормальными векторами  $a^{(i)}$ ,  $i \in I_k$  и  $-e^{(j)}$ ,  $j \in J_k$ , углы не меньшие чем  $\pi/2$ , т.е.

$$\langle a^{(i)}, p^{(k)} \rangle \leq 0, \quad i \in I_k, \quad (7.22)$$

$$-\langle e^{(j)}, p^{(k)} \rangle \leq 0 \quad \text{или} \quad p_j^{(k)} \geq 0, \quad j \in J_k. \quad (7.23)$$

Рис. 7.1 поясняет смысл ограничений (7.22) и (7.23) в двумерном случае. Рассмотрим теперь второе из упомянутых условий выбора  $p^{(k)}$  (минимальный угол между  $p^{(k)}$  и  $-f'(x^{(k)})$ ). Так как скалярное произведение двух векторов постоянной длины есть убывающая функция угла  $\varphi$  между ними при  $\varphi \in [0; \pi]$ , то минимальному углу между векторами  $p^{(k)}$  и  $-f'(x^{(k)})$  соответствует максимум скалярного про-

изведения  $\langle p^{(k)}, -f'(x^{(k)}) \rangle$  или минимум противоположной величины (вспомогательной целевой функции  $F(p_1, \dots, p_n)$ ):

$$F(p^{(k)}) = \langle f'(x^{(k)}), p^{(k)} \rangle \rightarrow \min. \quad (7.24)$$

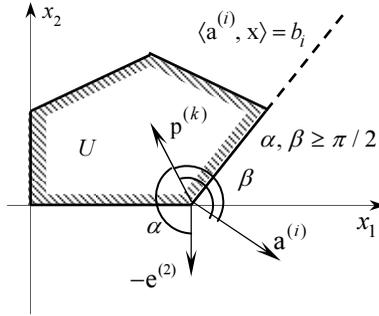


Рис. 7.1

Но минимум целевой функции (7.24) не достигается (она не ограничена снизу), если на норму вектора  $p^{(k)}$  не наложить ограничение, например,  $\|p^{(k)}\|^2 = \sum_{j=1}^n (p_j^k)^2 \leq 1$ . Однако задача минимизации (7.22) – (7.24) с этим ограничением становится задачей нелинейного программирования, поэтому введем вместо него другое, более удобное ограничение, препятствующее неограниченному росту нормы вектора  $p^{(k)}$ :

$$\sum_{j=1}^n |p_j^{(k)}| \leq 1. \quad (7.25)$$

Тогда поиск направления вектора  $p^{(k)}$  сведется к решению задачи (7.22), (7.23), (7.25). Хотя в этом случае вектор  $p^{(k)}$  не обязательно будет составлять с антиградиентом  $-f'(x^{(k)})$  минимально возможный угол, зато, как будет показано ниже, соотношения (7.22), (7.23), (7.25) можно представить в виде задачи линейного программирования.

Замечание. Если  $J_k \neq \emptyset$ , то среди переменных  $p_1, \dots, p_n$  задачи (7.22), (7.23), (7.25) есть заведомо неотрицательные, а именно  $p_j \geq 0$ ,  $j \in J_k$ ,  $k < n$  (см. условие (7.23)). К остальным переменным  $p_j$ , которые не являются неотрицательными, можно применить сле-

дующий искусственный прием. Представим переменную  $p_j$ ,  $j \notin J_k$  в виде  $p_j = p_j^+ - p_j^-$ ,

$$\text{где } p_j^+ = \begin{cases} p_j, & \text{если } p_j \geq 0, \\ 0, & \text{если } p_j < 0, \end{cases} \quad p_j^- = \begin{cases} 0, & \text{если } p_j \geq 0, \\ -p_j, & \text{если } p_j < 0. \end{cases}$$

Например, если  $p_j = 3$ , то  $p_j^+ = 3$ ,  $p_j^- = 0$ ; если  $p_j = -5$ , то  $p_j^+ = 0$ ,  $p_j^- = 5$ .

Таким образом, вновь введенные переменные подчинены условию неотрицательности  $p_j^+ \geq 0$ ,  $p_j^- \geq 0$  при произвольном значении  $p_j$ . Однако новые переменные должны удовлетворять дополнительному требованию  $p_j^+ \cdot p_j^- = 0$ , которое не входит в задачу линейного программирования. Для учета этого требования достаточно на любом шаге преобразования симплекс-методом переменные  $p_j^+$  и  $p_j^-$  с одинаковым индексом  $j$  не включать в базис одновременно.

$$p_j = p_j^+ - p_j^-, \quad j \notin J_k, \quad p_j^+ \geq 0, \quad p_j^- \geq 0 \quad \text{и} \quad p_j^+ \cdot p_j^- = 0. \quad (7.26)$$

Итак, направление спуска  $\mathbf{p}^{(k)}$  из точки  $\mathbf{x}^{(k)}$  в (7.15) ищется как решение следующей задачи линейного программирования:

$$F(p_1, \dots, p_n) = \langle \mathbf{f}'(\mathbf{x}^{(k)}), \mathbf{p} \rangle = \\ = \sum_{j \in J_k} \frac{\partial f(\mathbf{x}^{(k)})}{\partial x_j} p_j + \sum_{j \notin J_k} \frac{\partial f(\mathbf{x}^{(k)})}{\partial x_j} (p_j^+ - p_j^-) \rightarrow \min; \quad (7.27)$$

$$\langle \mathbf{a}^{(i)}, \mathbf{p} \rangle = \sum_{j \in J_k} a_j^{(i)} p_j + \sum_{j \notin J_k} a_j^{(i)} (p_j^+ - p_j^-) \leq 0, \quad i \in I_k; \quad (7.28)$$

$$\sum_{j=1}^n |p_j| = \sum_{j \in J_k} p_j + \sum_{j \notin J_k} (p_j^+ + p_j^-) \leq 1; \quad (7.29)$$

$$p_j \geq 0, \quad j \in J_k; \quad p_j^+, p_j^- \geq 0, \quad j \notin J_k \quad (7.30)$$

с дополнительным условием

$$p_j^+ \cdot p_j^- = 0, \quad j \notin J_k. \quad (7.31)$$

При решении задачи (7.27) – (7.30) симплекс-методом для учета условий (7.31) переменные  $p_j^+$  и  $p_j^-$  с одинаковым номером  $j \notin J_k$  не следует включать в число базисных одновременно.

Опишем теперь, как определить величину шага  $\alpha_k$  в формуле (7.15). Для найденного вектора  $\mathbf{p}^{(k)}$  она находится из условия

$$f(\mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}) = \min_{\alpha > 0, \mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)} \in U} f(\mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)}),$$

$$\text{т. е.} \quad \alpha_k = \min \{ \bar{\alpha}_1, \dots, \bar{\alpha}_m, \hat{\alpha}_1, \dots, \hat{\alpha}_n, \alpha^* \}, \quad (7.32)$$

где  $\bar{\alpha}_i$  и  $\hat{\alpha}_j$  – максимальные перемещения, при которых для точки  $\mathbf{x}^{(k+1)}$  из (7.15) выполняются соответственно  $i$ -е ограничение (7.17) и  $j$ -е ограничение (7.18), а именно:

$$\bar{\alpha}_i = \begin{cases} +\infty, & \text{если } \langle \mathbf{a}^{(i)}, \mathbf{p}^{(k)} \rangle \leq 0, \\ \frac{b_i - \langle \mathbf{a}^{(i)}, \mathbf{x}^{(k)} \rangle}{\langle \mathbf{a}^{(i)}, \mathbf{p}^{(k)} \rangle}, & \text{если } \langle \mathbf{a}^{(i)}, \mathbf{p}^{(k)} \rangle > 0, \end{cases} \quad (7.33)$$

$$\hat{\alpha}_j = \begin{cases} +\infty, & \text{если } p_j^{(k)} \geq 0, \\ -\frac{x_j^{(k)}}{p_j^{(k)}}, & \text{если } p_j^{(k)} < 0, \end{cases} \quad (7.34)$$

а величина  $\alpha^*$  находится из условия исчерпывающего спуска вдоль направления вектора  $\mathbf{p}^{(k)}$  без учета ограничений (7.17), (7.18), т.е.

$$\Phi_k(\alpha^*) = \min_{\alpha > 0} \Phi_k(\alpha), \quad \text{где } \Phi_k(\alpha) = f(\mathbf{x}^{(k)} + \alpha \mathbf{p}^{(k)}). \quad (7.35)$$

*Алгоритм метода возможных направлений* ( $k$ -я итерация) решения задачи (7.16) – (7.18) :

1. Найти  $\mathbf{f}'(\mathbf{x}^{(k)})$ . Если  $\|\mathbf{f}'(\mathbf{x}^{(k)})\| \leq \varepsilon$ , где  $\varepsilon$  заданный параметр точности, то вычисления прекратить, положив  $\mathbf{x}^* = \mathbf{x}^{(k)}$ . Иначе – перейти к шагу 2.

2. Подставить  $\mathbf{x}^{(k)}$  в неравенства (7.17) и (7.18) и определить множества индексов  $I_k, J_k$  по формулам (7.20).

3. Если  $I_k = J_k = \emptyset$ , то найти вектор  $\mathbf{p}^{(k)}$  из формулы (7.19), в противном случае определить  $\mathbf{p}^{(k)}$  из решения задачи линейного программирования (7.27) – (7.31) и формул (7.26).

4. Если  $\mathbf{p}^{(k)} = \mathbf{0}$ , то задача (7.16) – (7.18) решена точно,  $\mathbf{x}^* = \mathbf{x}^{(k)}$ . Иначе – для найденного вектора  $\mathbf{p}^{(k)}$  определить  $\alpha_k$  из формул (7.32) – (7.35).

5. Найти очередное приближение  $x^{(k+1)}$  по формуле (7.15).

**Пример 7.1.** Решить следующую задачу нелинейного программирования с линейными ограничениями:

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

$$\begin{cases} x_1 + x_2 \leq 3, \\ x_1 + 2x_2 \leq 4, \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

методом возможных направлений, полагая  $\varepsilon = 0.01$ . В качестве начального приближения взять точку  $x^{(0)} = (1/2, 17/12) \in U$ .

**Решение**

*Итерация 1*

1. Найдем градиент  $f'(x^{(0)}) = (-7, -7/6)$ . Так как  $\|f'(x^{(0)})\| > \varepsilon$ , то перейдем к шагу (2).

2. Ограничения (7.17) и (7.18) в точке  $x^{(0)}$  выполняются как строгие неравенства (легко проверить), т.е.  $x^{(0)}$  – внутренняя точка множества  $U$ . Поэтому  $I_0 = J_0 = \emptyset$ .

3. В соответствии с (7.19) полагаем  $p^{(0)} = -f'(x^{(0)}) = (7, 7/6)$ .

4. Из формул (7.33) и (7.34) находим:  $\bar{\alpha}_1 = 13/98$ ,  $\bar{\alpha}_2 = 1/14$ ,  $\hat{\alpha}_1 = \hat{\alpha}_2 = \infty$ .

Определим  $\alpha^*$  в соответствии с (7.35), используя условие минимума  $\Phi'_0(\alpha) = 0$ :

$$\Phi_0(\alpha) = f(x^{(0)} + \alpha p^{(0)}) = \left(\frac{1}{2} + 7\alpha - 4\right)^2 + \left(\frac{17}{12} + \frac{7}{6}\alpha - 2\right)^2,$$

поэтому

$$\Phi'_0(\alpha) = 14 \left(7\alpha - \frac{7}{2}\right) + \frac{7}{3} \left(\frac{7}{6}\alpha - \frac{7}{12}\right) = 0,$$

откуда  $\alpha^* = 0.5$ .

Из (7.32) окончательно находим  $\alpha_0 = 1/14$ .

5. Используя равенство (7.15), получаем  $x^{(1)} = (1/2, 17/12) + (1/14)(7, 7/6) = (1, 3/2)$ .

### Итерация 2

1. Найдем  $f'(x^{(1)}) = (-6, -1)$ . Так как  $\|f'(x^{(1)})\| > \varepsilon$ , то перейдем к шагу 2.

2. Точка  $x^{(1)}$  принадлежит границе множества  $U$ , причем второе ограничение задачи является для этой точки активным, поэтому  $I_1 = \{2\}$ ,  $J_1 = \emptyset$ .

3. Задача (7.27) – (7.31) для определения вектора  $p^{(1)}$  принимает вид

$$\begin{aligned} F(p) &= -6p_1^{(1)+} + 6p_1^{(1)-} - p_2^{(1)+} + p_2^{(1)-} \rightarrow \min, \\ \begin{cases} p_1^{(1)+} - p_1^{(1)-} + 2p_2^{(1)+} - 2p_2^{(1)-} \leq 0, \\ p_1^{(1)+} + p_1^{(1)-} + p_2^{(1)+} + p_2^{(1)-} \leq 1, \\ p_j^{(1)+}, p_j^{(1)-} \geq 0, \quad p_j^{(1)+} \cdot p_j^{(1)-} = 0, \quad j=1,2. \end{cases} \end{aligned}$$

Записав эту задачу в каноническом виде с помощью дополнительных переменных и выбрав эти переменные в качестве базисных, в результате двух итераций симплекс-метода получим:  $p_1^{(1)+} = 2/3$ ,  $p_1^{(1)-} = 0$ ,  $p_2^{(1)+} = 0$ ,  $p_2^{(1)-} = 1/3$ , т.е.  $p^{(1)} = (p_1^{(1)+} - p_1^{(1)-}, p_2^{(1)+} - p_2^{(1)-}) = (2/3, -1/3)$ .

4. По формулам (7.33) – (7.35) находим  $\bar{\alpha}_1 = 3/2$ ,  $\bar{\alpha}_2 = \infty$ ,  $\hat{\alpha}_1 = \infty$ ,  $\hat{\alpha}_2 = 9/2$ ,  $\alpha^* = 156/85$ , поэтому в соответствии с (7.32) имеем  $\alpha_1 = 3/2$ .

5. Находим очередное приближение  $x^{(2)}$  по формуле (7.15):  $x^{(2)} = (2, 1)$ .

### Итерация 3

1. Определяем  $f'(x^{(2)}) = (-4, -2)$ . Так как  $\|f'(x^{(2)})\| > \varepsilon$ , то переходим к шагу (2).

2. В точке  $x^{(2)}$  имеем:  $I_2 = \{1, 2\}$ ,  $J_2 = \emptyset$ .

3. Составим задачу линейного программирования для нахождения направления  $p^{(2)}$ :

$$F(p) = -4p_1^{(2)+} + 4p_1^{(2)-} - 2p_2^{(2)+} + 2p_2^{(2)-} \rightarrow \min;$$

$$\begin{cases} p_1^{(2)+} - p_1^{(2)-} + p_2^{(2)+} - p_2^{(2)-} \leq 0, \\ p_1^{(2)+} - p_1^{(2)-} + 2p_2^{(2)+} - 2p_2^{(2)-} \leq 0, \\ p_1^{(2)+} + p_1^{(2)-} + p_2^{(2)+} + p_2^{(2)-} \leq 1, \\ p_j^{(2)+}, p_j^{(2)-} \geq 0, \quad p_j^{(2)+} \cdot p_j^{(2)-} = 0, \quad j=1,2. \end{cases}$$

Решив эту задачу, получим:  $p^{(2)} = (1/2, -1/2)$ .

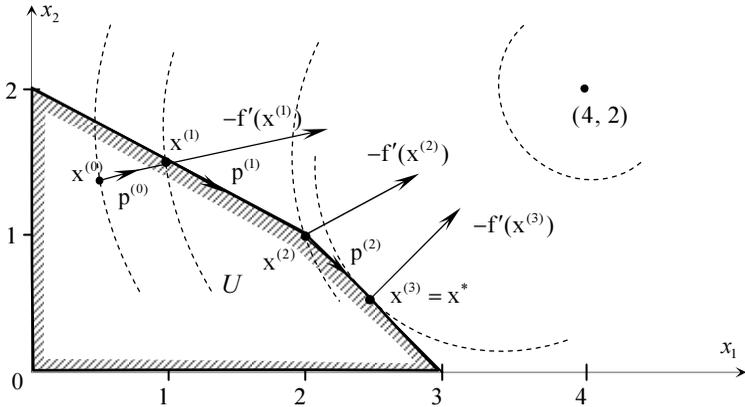


Рис. 7.2

4. По формулам (7.32) – (7.35) получим  $\bar{\alpha}_1 = \bar{\alpha}_2 = \infty$ ,  $\hat{\alpha}_1 = \infty$ ,  $\hat{\alpha}_2 = 1$ ,  $\alpha^* = 1$ , т.е.  $\alpha_2 = 1$ .

5. По формуле (7.15) находим  $x^{(3)} = (5/2, 1/2)$ .

#### Итерация 4

На третьем шаге этой итерации получим  $p^{(3)} = (0, 0)$ . Это означает, что точка минимума найдена точно:  $x^* = x^{(3)} = (5/2, 1/2)$ . Отсюда  $f_{\min} = f(x^*) = 9/2$ .

На рис. 7.2 приведена геометрическая иллюстрация хода решения задачи. На нем штриховыми линиями показаны линии уровня целевой функции (окружности с центром в точке  $(4, 2)$ ).

#### Случай нелинейных ограничений

Метод возможных направлений используется также для решения задачи нелинейного программирования более общего вида, чем (7.16) – (7.18), а именно:

$$f(x) \rightarrow \min, \quad x \in R^n; \quad (7.36)$$

$$g_i(x) \leq 0, \quad i = \overline{1, m}, \quad (7.37)$$

где  $f(x)$  и  $g_i(x)$  – выпуклые дифференцируемые в  $R^n$  функции. Допустимое множество  $U$  задачи (7.36), (7.37) выпукло, поэтому для любой точки  $x \in U$  существует возможное направление  $p$  [3].

Опишем один из вариантов построения итерационной последовательности (7.15) при решении задачи (7.36), (7.37), а также укажем критерий окончания вычислений.

Начнем с определения вектора  $p^{(k)}$ .

Если  $x^{(k)}$  – внутренняя точка множества  $U$ , то следует положить

$$p^{(k)} = -f'(x^{(k)}). \quad (7.38)$$

Пусть теперь  $x^{(k)}$  – граничная точка множества  $U$ , т.е. существует непустое множество индексов  $I_k = \{i \mid g_i(x^{(k)}) = 0\}$ . Среди функций  $g_i(x)$ ,  $i \in I_k$  могут встретиться линейные, тогда разобьем множество индексов  $I_k$  на два непересекающихся подмножества  $I_k^N$  и  $I_k^L$ , относящиеся соответственно к нелинейным и линейным функциям  $g_i(x)$ :

$$I_k^N = \{i \in I_k \mid g_i(x) \text{ – нелинейна}\}, \quad I_k^L = \{i \in I_k \mid g_i(x) \text{ – линейна}\}.$$

В пространстве  $R^{n+1}$  переменных  $(p, \sigma) = (p_1, \dots, p_n, \sigma)$  рассмотрим вспомогательную задачу линейного программирования, позволяющую найти направление убывания в точке  $x^{(k)}$  целевой функции  $f(x)$  из (7.36) на множестве  $U$ , определяемом ограничениями (7.37):

$$F(p_1, \dots, p_n, \sigma) = -\sigma \rightarrow \min; \quad (7.39)$$

$$\langle f'(x^{(k)}), p \rangle + \sigma \leq 0; \quad (7.40)$$

$$\langle g'_i(x^{(k)}), p \rangle + \sigma \leq 0, \quad i \in I_k^N; \quad (7.41)$$

$$\langle g'_i(x^{(k)}), p \rangle \leq 0, \quad i \in I_k^L; \quad (7.42)$$

$$\sigma \geq 0, \quad p_j = p_j^+ - p_j^-, \quad p_j^+ \cdot p_j^- = 0, \quad p_j^+ \geq 0, \quad p_j^- \geq 0, \quad j = \overline{1, n}; \quad (7.43)$$

$$\sum_{j=1}^n |p_j| = \sum_{j=1}^n (p_j^+ + p_j^-) \leq 1. \quad (7.44)$$

Соотношения (7.43), (7.44) имеют тот же смысл, что и во вспомогательной задаче (7.27) – (7.31). Очевидно, остается в силе высказан-

ное там требование к процедуре применения симплекс-метода: переменные  $p_j^+$  и  $p_j^-$  с одинаковыми индексами  $j$  нельзя вводить в число базисных одновременно.

**Теорема 7.1.** Пусть  $x^{(k)}$  – граничная точка допустимого множества  $U$  задачи (7.36), (7.37). Тогда решение  $(p^{(k)}, \sigma_k)$  задачи линейного программирования (7.39) – (7.44) существует. При этом, если  $\sigma_k > 0$ , то вектор  $p^{(k)}$  задает направление убывания функции  $f(x)$  на множестве  $U$  из точки  $x^{(k)}$ . Если же  $\sigma_k = 0$ , то  $x^{(k)}$  есть точка минимума  $f(x)$  на множестве  $U$ .

**Доказательство.** Допустимое множество  $U_k$  задачи (7.39) – (7.44), задаваемое условиями (7.40) – (7.44), непусто, так как точка  $(p, \sigma) = (0, 0)$  принадлежит  $U_k$ . Поэтому на выпуклом многограннике  $U_k$  линейная функция  $F(p, \sigma) = -\sigma$  достигает минимального значения  $-\sigma_k$  в некоторой точке  $(p^{(k)}, \sigma_k)$ .

Рассмотрим сначала случай  $\sigma_k > 0$ .

1. Убедимся в том, что решение  $(p^{(k)}, \sigma_k)$  задачи (7.39) – (7.44) при  $\sigma_k > 0$  задает возможное в точке  $x^{(k)}$  направление  $p^{(k)}$ . Из условий (7.40) и (7.41) при  $\sigma = \sigma_k > 0$  следует, что  $p^{(k)} \neq 0$ . Кроме того, для  $\sigma_k > 0$  с учетом (7.41) и (7.42) имеем:  $\langle g'_i(x^{(k)}), p^{(k)} \rangle < 0$ ,  $i \in I_k^N$ ,  $\langle g'_i(x^{(k)}), p^{(k)} \rangle \leq 0$ ,  $i \in I_k^L$ , т.е.  $p^{(k)}$  – возможное направление.

2. Покажем, что при  $\sigma_k > 0$  возможное направление  $p^{(k)}$  является направлением убывания функции  $f(x)$ . В самом деле, для  $\sigma > 0$  с учетом (7.40) можно записать:

$$f(x^{(k)} + \alpha p^{(k)}) - f(x^{(k)}) = \alpha \langle f'(x^{(k)}), p^{(k)} \rangle + o(\alpha) \leq \alpha (-\sigma_k + o(\alpha)/\alpha).$$

Поэтому найдется достаточно малое число  $\sigma_0 > 0$  такое, что

$$f(x^{(k)} + \alpha p^{(k)}) - f(x^{(k)}) < 0 \quad \text{для всех } \alpha \in (0, \alpha_0).$$

Рассмотрим теперь случай  $\sigma_k = 0$ .

Из соотношений (7.40) – (7.42) следует, что при  $\sigma_k = 0$  не существует вектора  $p \neq 0$  такого, чтобы одновременно выполнялись неравенства:

$$\langle f'(x^{(k)}), p \rangle < 0,$$

$$\langle g'_i(x^{(k)}), p \rangle < 0, \quad i \in I_k^N; \quad (7.45)$$

$$\langle g'_i(x^{(k)}), p \rangle \leq 0, \quad i \in I_k^L, \quad (7.46)$$

т.е. для всех возможных в точке  $x^{(k)}$  направлений  $p$ , определяемых неравенствами (7.45) и (7.46), выполняется условие  $\langle f'(x^{(k)}), p \rangle \geq 0$ , т.е. эти направления не являются направлениями убывания  $f(x)$  в точке  $x^{(k)}$  (см. теорему 5.8). А это означает, что в точках допустимого множества  $U$ , достаточно близких к  $x^{(k)}$ , выполняется неравенство  $f(x) \geq f(x^{(k)})$ , т.е.  $x^{(k)}$  – точка локального минимума  $f(x)$  на множестве  $U$ . Но для выпуклой функции  $f(x)$  на выпуклом множестве  $U$  этот минимум является и глобальным (см. теорему 5.4). Таким образом, точка  $x^{(k)}$  есть решение задачи (7.36) – (7.37).

После определения возможного направления убывания  $p^{(k)}$  из точки  $x^{(k)}$  находится величина перемещения  $\alpha_k$ :

$$\alpha_k = \min\{\alpha_{k1}, \dots, \alpha_{km}, \alpha_k^*\},$$

где  $\alpha_k^*$  определяется из условия  $f(x^{(k)} + \alpha^* p^{(k)}) = \min_{\alpha > 0} f(x^{(k)} + \alpha p^{(k)})$ ,

а  $\alpha_{ki}$ ,  $i = \overline{1, m}$  – максимально возможное перемещение из точки  $x^{(k)}$  вдоль направления  $p^{(k)}$  с учетом  $i$ -го ограничения (7.37), которое найдено из условия

$$g_i(x^{(k)} + \alpha_{ki} p^{(k)}) = 0. \quad (7.47)$$

Критерием достижения заданной точности решения  $\varepsilon > 0$  задачи (7.36), (7.37) методом возможных направлений служит выполнение хотя бы одного из неравенств  $\|f'(x^{(k)})\| \leq \varepsilon$ ,  $\sigma_k \leq \varepsilon$ . При выполнении этого условия полагают  $x^* = x^{(k)}$ ,  $f^* = f(x^{(k)})$ .

Любое из равенств  $\|f'(x^{(k)})\| = 0$ ,  $\sigma_k = 0$  означает, что точка минимума  $x^*$  функции  $f(x)$  на множестве  $U$  найдена точно:  $x^* = x^{(k)}$ .

**Алгоритм метода возможных направлений** ( $k$ -ая итерация) решения задачи (7.36), (7.37):

1. Найти  $f'(x^{(k)})$ . Если  $\|f'(x^{(k)})\| \leq \varepsilon$ , где  $\varepsilon$  – заданный параметр точности, то перейти к шагу (2), иначе – перейти к шагу 3.

2. Вычисления прекратить, положить  $x^* = x^{(k)}$ ,  $f_{\min} = f(x^*)$ .

3. Подставить  $x^{(k)}$  в неравенства (7.37) и определить множества индексов  $I_k^N$  и  $I_k^L$ .

4. Если  $I_k^N = I_k^L = \emptyset$ , то найти вектор  $p^{(k)}$  по формуле (7.38), в противном случае найти решение  $(p^{(k)}, \sigma_k)$  задачи линейного программирования (7.39) – (7.44). Если  $p^{(k)} = 0$  или  $\sigma_k \leq \varepsilon$ , то перейти к шагу 2, иначе – продолжить вычисления.

5. Для всех  $i = \overline{1, m}$ ,  $i \notin I_k^L$  найти максимально допустимые величины шага перемещения  $\alpha_{ki}$  вдоль направления  $p^{(k)}$  с учетом ограничений (7.37), используя условия (7.47) (для  $i \in I_k^L$ , очевидно,  $\alpha_{ki} = +\infty$ ).

6. Определить величину  $\alpha_k^*$  из условия исчерпывающего спуска вдоль направления  $p^{(k)}$  без учета ограничений (7.37), т.е.

$$F_k(\alpha_k^*) = \min_{\alpha > 0} F_k(\alpha), \text{ где } F_k(\alpha) = f(x^{(k)} + \alpha p^{(k)}).$$

7. Положив  $\alpha_k = \min\{\alpha_{ki}, \alpha_k^*\}$ , вычислить очередное приближение  $x^{(k+1)}$  к точке минимума  $x^*$  по формуле (7.15).

**Пример 7.2.** Методом возможных направлений решить следующую задачу нелинейного программирования:

$$f(x) = -x_1 - x_2 \rightarrow \min;$$

$$g_1(x) = x_1^2 + x_2^2 - 25 \leq 0,$$

полагая  $\varepsilon = 0.1$  и начальное значение  $x^{(0)} = (0, 1)$ . Очевидно,  $x^{(0)} \in U$ .

**Решение.** Функции  $f(x)$  и  $g_1(x)$  выпуклы и дифференцируемы в  $R^2$ .

*Итерация 1*

1. Найдем  $f'(x^{(0)}) = (-1, -1)$ . Так как  $\|f'(x^{(0)})\| > \varepsilon$ , переходим к шагу 3.

3. Так как  $x^{(0)}$  внутренняя точка множества  $U$  ( $g_1(x^{(0)}) < 0$ ), то  $I_0 = I_0^N = \emptyset$ .

4. Найдем  $p^{(0)}$  по формуле (7.38):  $p^{(0)} = -f'(x^{(0)}) = (1, 1)$ .

5. Найдем максимально возможное перемещение  $\alpha_{01}$  из условия  $g_1(x^{(0)} + \alpha_{01}p^{(0)}) = 0$ :  $(0 + \alpha_{01})^2 + (1 + \alpha_{01})^2 - 25 = 0$ , откуда  $\alpha_{01} = 3$ .

6. Так как функция  $F_0(\alpha) = f(x^{(0)} + \alpha p^{(0)}) = -2\alpha - 1$  неограниченно убывает при  $\alpha \rightarrow +\infty$ , следует положить  $\alpha_0^* = +\infty$ .

7. Величина перемещения по направлению  $p^0$  на первой итерации  $\alpha_0 = \min\{3, +\infty\} = 3$ . Поэтому следующее приближение  $x^{(1)} = x^{(0)} + \alpha_0 p^{(0)} = (0, 1) + 3(1, 1) = (3, 4)$ .

### Итерация 2

1. Найдем  $f'(x^{(1)}) = (-1, -1)$ . Так как  $\|f'(x^{(1)})\| > \varepsilon$ , то переходим к шагу (3).

3. Точка  $x^{(1)} = (3, 4)$  является граничной на множестве  $U$  ( $g_1(x^{(1)}) = 0$ ) и  $I_1 = \{1\}$ .

4. Вектор  $p^{(1)}$  находим из решения задачи линейного программирования (7.39) – (5.44):

$$\begin{aligned} F_1(p^{(1)}, \sigma) &= -\sigma \rightarrow \min ; \\ \begin{cases} -p_1^{(1)+} + p_1^{(1)-} - p_2^{(1)+} + p_2^{(1)-} + \sigma \leq 0, \\ 6p_1^{(1)+} - 6p_1^{(1)-} + 8p_2^{(1)+} - 8p_2^{(1)-} + \sigma \leq 0, \\ p_1^{(1)+} + p_1^{(1)-} + p_2^{(1)+} + p_2^{(1)-} \leq 1, \end{cases} \\ \sigma, p_j^{(1)+}, p_j^{(1)-} &\geq 0, \quad p_j^{(1)+} \cdot p_j^{(1)-} = 0, \quad j = 1, 2. \end{aligned}$$

Решив ее симплекс-методом, получим  $\sigma_1 = 1/8$ ,  $p_1^{(1)+} = 9/16$ ,  $p_2^{(1)-} = 7/16$ ,  $p_1^{(1)-} = p_2^{(1)+} = 0$ ,  $p^{(1)} = (9/16, -7/16)$ . Так как условия остановки вычислительного процесса не выполнены, продолжим вычисления.

5. Максимально допустимое перемещение  $\alpha_{11}$  в направлении  $p^{(1)}$  с учетом ограничения  $g_1(x) \leq 0$  равно  $\alpha_{11} = 16/65$ .

6. Величина  $\alpha_1^*$ , как и на предыдущей итерации, не ограничена, т.е.  $\alpha_1^* = +\infty$ .

7. Окончательно величина перемещения на второй итерации  $\alpha_1 = \min\{\alpha_{11}, \alpha_1^*\} = 16/65$ . Поэтому следующее приближение  $x^{(2)} = x^{(1)} + \alpha_1 p^{(1)} = (3, 4) + (16/65)(9/16, -7/16) = (3.138, 3.892)$ .

### Итерация 3

Так как  $g_1(x^{(2)}) = 0$ , то точка  $x^{(2)}$  является граничной. Находя по общему правилу  $p^{(2)}$  и  $\alpha_2$ , получаем  $\sigma_2 = 0.125$ ,  $p^{(2)} = (0.547, -0.453)$ ,  $\alpha_2 = 0.186$ . Отсюда  $x^{(3)} = (3.240, 3.809)$ .

### Итерация 4

На этой итерации, решая задачу линейного программирования для направления  $p^{(3)}$ , получаем, что  $\sigma_3 = 0.094 < 0.1$ , т.е. требуемая точность достигнута. Поэтому  $x^* = x^{(3)} = (3.240, 3.809)$ ,  $f_{\min} = f(x^{(3)}) = -7.049$ .

Отметим, что задача из этого примера допускает графическое решение, результат которого  $x^* = (5\sqrt{2}/2, 5\sqrt{2}/2) \approx (3.536, 3.536)$  и  $f^* = -5\sqrt{2} \approx -7.071$ .

Метод возможных направлений решения задач выпуклого программирования является довольно мощной оптимизационной процедурой, особенно для допустимых множеств простой структуры. Однако, если учитывать только активные в точке  $x^{(k)}$  ограничения, то метод «не чувствует» ограничений, которые в точке  $x^{(k)}$  выполняются почти как равенства (т.е.  $|g_i(x^{(k)})|$  мало), а их присутствие резко ограничивает длину шага из точки  $x^{(k)}$ . Поэтому может произойти «остановка» метода в окрестности точки, не являющейся решением задачи. Чтобы этого избежать, вводят понятие  $\varepsilon$ -активных ограничений, т.е. таких, которые с точностью до параметра  $\varepsilon \geq 0$  обращаются в равенства:  $I_k = \{i \mid g_i(x^{(k)}) \geq -\varepsilon_k\}$ . В процессе решения величину  $\varepsilon_k$  устремляют к нулю.

Недостатком этого метода является необходимость выбора начальной точки внутри допустимого множества, что может представлять собой самостоятельную довольно трудную задачу.

### 7.3. Градиентные методы

Приведем описание некоторых модификаций градиентных методов безусловной минимизации, учитывающих наличие ограничений на переменные в задаче выпуклого программирования

$$f(x) \rightarrow \min, \quad x \in R^n, \quad (7.48)$$

$$g_i(x) \leq 0, \quad i = \overline{1, m}, \quad (7.49)$$

где функции  $f(x)$  и  $g_i(x)$  выпуклы и дифференцируемы.

#### Метод проекции градиента

На каждой итерации этого метода предусмотрена процедура возврата очередного приближения градиентного спуска  $x^{(k+1)} = x^{(k)} - \alpha_k f'(x^{(k)})$  на допустимое множество  $U$ , если  $x^{(k+1)} \notin U$ . Такой возврат производится *проектированием* точки  $x^{(k+1)}$  на множество  $U$ .

**Определение 7.1.** *Проекцией* точки  $x \in R^n$  на замкнутое множество  $U \subset R^n$  называется ближайшая к  $x$  точка  $\hat{x}$  множества  $U$ , т.е.

$$\rho(x, \hat{x}) = \|x - \hat{x}\| = \min_{y \in U} \|x - y\|,$$

если такая точка существует.

Проекцию  $\hat{x}$  точки  $x$  на множество  $U$  будем обозначать через  $P_U(x)$ . Если  $x \in U$ , то, очевидно,  $P_U(x) = x$ .

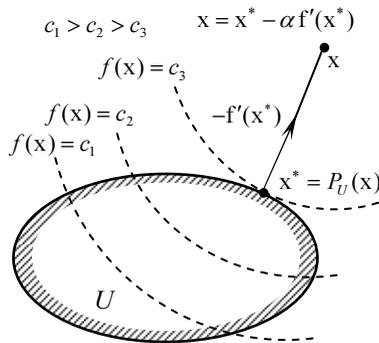


Рис. 7.3

#### Замечания

1. Можно показать, что если  $U$  – выпуклое замкнутое множество из  $R^n$ , то для любой точки  $x \in R^n$  существует единственная проекция на это множество.

2. Необходимым и достаточным условием того, чтобы точка  $x^*$  была решением задачи (7.48) и (7.49), является равенство (рис. 7.3):

$$x^* = P_U \left[ x^* - \alpha f'(x^*) \right] \quad \text{для всех } \alpha > 0. \quad (7.50)$$

Геометрически условие (7.50) представлено на рис 7.3.

Таким образом, в методе проекции градиента последовательные приближения  $x^{(k)}$  к точке минимума  $x^*$  целевой функции  $f(x)$  на множестве  $U$  вычисляются по формулам

$$x^{(k+1)} = P_U \left[ x^{(k)} - \alpha_k f'(x^{(k)}) \right], \quad k = 0, 1, \dots, \quad x^{(0)} \in U, \quad (7.51)$$

где величина  $\alpha_k$  выбирается одним из способов, рассмотренных при описании градиентных методов безусловной минимизации.

Вычисления по формуле (7.51) завершаются при выполнении одного из неравенств  $\|f'(x^{(k)})\| \leq \varepsilon$  или  $\|x^{(k)} - x^{(k-1)}\| \leq \varepsilon$ , где величина  $\varepsilon > 0$  определяет точность решения задачи. При этом полагают  $x^* = x^{(k)}$  и  $f^* = f(x^{(k)})$ .

Отметим, что задача проектирования на множество  $U$  в итерационном процессе (7.51) в свою очередь также является задачей нелинейного программирования:

$$\Phi_k = \|y^{(k)} - x\|^2 \rightarrow \min, \quad x \in U, \quad y^{(k)} = x^{(k)} - \alpha_k f'(x^{(k)}), \quad (7.52)$$

и ее решение может вызвать значительные затруднения. Однако если множество  $U$  задается лишь линейными ограничениями–неравенствами, то (7.52) является задачей квадратичного программирования и может быть решена за конечное число шагов.

Особый интерес при использовании метода проекции градиента представляют такие множества  $U$ , для которых задача проектирования решается в явном виде. Рассмотрим некоторые из таких множеств.

**Пример 7.3.** Пусть  $U = \{x \mid x_j \geq 0, j = \overline{1, n}\}$  – неотрицательный октант пространства  $R^n$ . Тогда  $P_U(x) = \hat{x}$ , где  $\hat{x}_j = \max\{0, x_j\}$ ,  $j = \overline{1, n}$ .

**Пример 7.4.** Пусть  $U = \{x \mid a_j \leq x_j \leq b_j, a_j < b_j, j = \overline{1, n}\}$  –  $n$ -мерный параллелепипед. Тогда  $P_U(x) = \hat{x}$ , где координаты проекции  $\hat{x}$  определяются формулой:

$$\hat{x}_j = \begin{cases} a_j, & \text{если } x_j < a_j; \\ b_j, & \text{если } x_j > b_j; \\ x_j, & \text{если } a_j \leq x_j \leq b_j. \end{cases}$$

**Пример 7.5.** Пусть  $U = \{x \mid \|x - x^{(0)}\| \leq R\}$  – шар радиуса  $R > 0$  с центром в точке  $x^{(0)}$ . Проекция точки  $x \in R^n$  может быть найдена по формуле:

$$P_U(x) = \begin{cases} x, & \text{если } x \in U, \\ x^{(0)} + R \frac{x - x^{(0)}}{\|x - x^{(0)}\|}, & \text{если } x \notin U. \end{cases}$$

**Пример 7.6.** Пусть  $U = \{x \mid \langle a, x \rangle = b, a \neq 0\}$  – гиперплоскость в  $R^n$  с нормальным вектором  $a$ . Для любой точки  $x \in R^n$  имеем:

$$P_U(x) = x - \frac{\langle a, x \rangle - b}{\|a\|^2} a. \quad (7.53)$$

Действительно, при проектировании на плоскость  $U$  точки  $x \notin U$  необходимо, чтобы векторы  $\hat{x} - x$  и  $a$  были коллинеарны. Поэтому ищем проекцию  $\hat{x}$  в виде  $\hat{x} = x + \alpha a$ . Постоянную  $\alpha$  находим из условия  $\hat{x} \in U$ , т.е.  $\langle a, \hat{x} \rangle = b$ . Отсюда  $\alpha = -\frac{\langle a, x \rangle - b}{\|a\|^2}$ . Отметим, что если  $x \in U$ , то  $P_U(x) = x$ .

**Пример 7.7.** Пусть  $U = \{x \mid Ax = b, A - \text{матрица размера } m \times n \text{ с рангом } r = m < n; b \in R^m; x \in R^n\}$  – пересечение гиперплоскостей  $\langle a^{(i)}, x \rangle = b_i, i = \overline{1, m}$  в пространстве  $R^n$ , где  $a^{(i)}$  – векторы-строки матрицы  $A$ . В этом случае проекция любой точки  $x \in R^n$  может быть найдена по формуле:

$$\hat{x} = P_U(x) = x - A^T (AA^T)^{-1} (Ax - b).$$

Действительно, ищем проекцию  $\hat{x}$  в виде  $\hat{x} = x + A^T \alpha, \alpha \in R^m$ .

Из условия  $\hat{x} \in U$  для компонент неизвестного вектора  $\alpha$  получаем систему уравнений:

$$Ax + AA^T \alpha = b.$$

Так как ранг матрицы  $A$  по условию равен  $m$ , квадратная матрица  $AA^T$  невырождена и существует обратная матрица  $(AA^T)^{-1}$ , поэтому  $\alpha = -(AA^T)^{-1}(Ax - b)$ .

Рассмотрим пример решения задачи нелинейного программирования методом проекции градиента.

**Пример 7.8.** Решить задачу

$$f(x) = -x_1 + x_2^2 \rightarrow \min ;$$

$$x_1^2 + x_2^2 \leq 1$$

методом проекции градиента. Вычисления прекратить при выполнении одного из условий:  $\|f'(x^{(k)})\| \leq 0.01$  или  $\|x^{(k)} - x^{(k-1)}\| < 0.01$ .

**Решение.** Выберем в качестве начального приближения точку  $x^{(0)} = (0, 0.5) \in U$ . Она является внутренней точкой множества  $U$ .

Целевая функция квадратичная с матрицей  $A = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}$ . Наибольшее ее собственное значение  $L = 2$ . Поэтому в формуле (7.51) можно положить  $\alpha \in (0, 1)$ , например,  $\alpha = 0.75$  (см. [3]).

*Итерация 1*

В точке  $x^{(0)}$  градиент  $f'(x^{(0)}) = (-1, 1)$ . Используя формулу (7.51), получаем

$$x^{(1)} = P_U [x^{(0)} - \alpha f'(x^{(0)})] = P_U [(0, 0.5) - 0.75(-1, 1)] = P_U [(0.75, -0.25)].$$

Точка  $(0.75, -0.25)$  принадлежит множеству  $U$ , поэтому  $x^{(1)} = (0.75, -0.25)$ . Требуемая точность не достигнута, так как  $\|f'(x^{(1)})\| = 1.12 > 0.01$  и  $\|x^{(1)} - x^{(0)}\| = 0.79 > 0.01$ .

*Итерация 2*

Как и на предыдущей итерации, находим

$$x^{(2)} = P_U [x^{(1)} - \alpha f'(x^{(1)})] = P_U [(0.75, 0.25) - 0.75(-1, -0.5)] =$$

$$= P_U [(1.5, 0.125)].$$

Точка  $(1.5, 0.125) \notin U$ . Используя результат примера 7.4, получаем

$$x^{(2)} = P_U [(1.5, 0.125)] = \frac{(1.5, 0.125)}{\sqrt{1.5^2 + 0.125^2}} = (0.996, 0.083).$$

Требуемая точность не достигнута, так как  $\|f'(x^{(2)})\|=1.003 > 0.01$  и  $\|x^{(2)} - x^{(1)}\|=0.298 > 0.01$ .

Результаты остальных итераций метода проекции градиента приведены в табл. 7.1.

Таблица 7.1

$k$	$x^{(k)}$	$\ x^{(k)} - x^{(k-1)}\ $	$f'(x^{(k)})$	$x^{(k)} - \alpha f'(x^{(k)})$
3	(0.9997, -0.0238)	0.107	(-1, 0.0476)	(1.7497, 0.0119)
4	(0.99998, 0.00679)	0.031	(-1, 0.0136)	(1.74998, 0.00339)
5	(0.99998, -0.00194)	0.0087 – точность достигнута		

Из табл. 7.1 следует, что  $x^* = x^{(5)} = (0.99998, -0.00194)$ ,  $f_{\min} = f(x^{(5)}) = -1$ . Графическое решение этой задачи дает точный ответ:  $x^* = (1, 0)$ ,  $f_{\min} = -1$ .

### Метод условного градиента

В этом методе решения гладкой задачи выпуклого программирования (7.48), (7.49) строится итерационный процесс, на каждом шаге которого решается вспомогательная задача минимизации на множестве  $U$  линейной функции и одномерная задача минимизации.

Пусть  $x^{(k)} \in U$  – очередное приближение к решению задачи. Если  $f'(x^{(k)}) = 0$ , то  $x^* = x^{(k)}$ . В противном случае в окрестности точки  $x^{(k)}$  функция  $f(x)$  представима в виде ряда

$$f(x) = f(x^{(k)}) + \langle f'(x^{(k)}), x - x^{(k)} \rangle + o(\|x - x^{(k)}\|),$$

и линейная функция

$$\bar{f}_k(x) = f(x^{(k)}) + \langle f'(x^{(k)}), x - x^{(k)} \rangle$$

является приближением  $f(x)$  с точностью до величины  $o(\|x - x^{(k)}\|)$  в некоторой окрестности точки  $x^{(k)}$ .

Поставим вспомогательную задачу минимизации на множестве  $U$  линейной функции  $\bar{f}_k(x)$ , т.е.

$$\bar{f}_k(x) \rightarrow \min, \quad x \in U. \quad (7.54)$$

Пусть  $\bar{x}^{(k)}$  – решение задачи (7.54). Следующее приближение  $x^{(k+1)}$  к точке минимума  $x^*$  функции  $f(x)$  на множестве  $U$  найдем по формуле

$$x^{(k+1)} = x^{(k)} + \alpha_k (\bar{x}^{(k)} - x^{(k)}), \quad \alpha_k \in [0, 1]. \quad (7.55)$$

Очевидно, вследствие выпуклости множества  $U$  точка  $x^{(k+1)} \in U$ . Величина  $\alpha_k$  из (7.55) в различных вариантах метода условного градиента вычисляется по-разному. Приведем два способа определения  $\alpha_k$ .

1. Величина  $\alpha_k$  находится как решение одномерной задачи минимизации функции  $\Phi_k(\alpha) = f[x^{(k)} + \alpha(\bar{x}^{(k)} - x^{(k)})]$  на отрезке  $[0, 1]$ . На рис. 7.4 поясняется определение точек  $\bar{x}^{(k)}$  и  $x^{(k+1)}$  в методе условного градиента для  $x \in R^2$ .

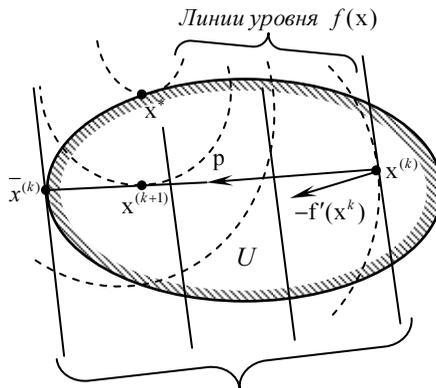
2. В формуле (7.55) полагается  $\alpha_k = 1$ , после чего проверяется условие

$$f(x^{(k+1)}) < f(x^{(k)}). \quad (7.56)$$

Если оно нарушается, то  $\alpha_k$  уменьшают (дробят), например в два раза, и повторно проверяют условие (7.56). Дробление  $\alpha_k$  и вычисления

$x^{(k+1)}$  производят до выполнения условия (7.56), после чего переходят к следующей итерации.

Условие окончания вычислений по методу условного градиента можно выбрать, например, таким  $\|x^{(k)} - x^{(k-1)}\| < \varepsilon$ , или  $\|f'(x^{(k)})\| < \varepsilon$ , или  $|f(x^{(k)}) - f(x^{(k-1)})| < \varepsilon$ .



Линии уровня

Обозначения:  $p = \bar{x}^{(k)} - x^{(k)}$ ;  $-f'(x^{(k)}) = -\bar{f}'(x^{(k)})$ .

Рис. 7.4

**Алгоритм метода условного градиента** ( $k$ -я итерация) решения задачи (7.48), (7.49):

1. Решив вспомогательную задачу минимизации с линейной целевой функцией  $\bar{f}_k(\mathbf{x}) \rightarrow \min, \mathbf{x} \in U$ , найти  $\bar{\mathbf{x}}^{(k)}$ .

2. Решить задачу одномерной минимизаций  $\Phi_k(\alpha) = f[\mathbf{x}^{(k)} + \alpha(\bar{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)})] \rightarrow \min, \alpha \in [0, 1]$ , т.е. найти  $\alpha^*$  и положить  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^*(\bar{\mathbf{x}}^{(k)} - \mathbf{x}^{(k)})$ .

3. Проверить условие достижения заданной точности. Если оно выполнено, то вычисления прекратить, положить  $\mathbf{x}^* = \mathbf{x}^{(k+1)}$ ,  $f_{\min} = f(\mathbf{x}^{(k+1)})$ . Иначе – перейти к шагу 1.

Отметим, что вспомогательная задача (7.54) является, вообще говоря, задачей нелинейного программирования и может оказаться немногим проще, чем исходная. Поэтому метод условного градиента применяется, когда решение задачи (7.54) находится достаточно просто. Укажем случаи, когда поиск ее решения  $\bar{\mathbf{x}}^{(k)}$  не представляет затруднений.

1. Допустимое множество  $U$  задано *линейными ограничениями*. Тогда (7.53) – это задача линейного программирования, и ее решение можно найти с помощью симплекс-метода (см. разд. 2.3).

2. Допустимое множество  $U = \{\mathbf{x} \in \mathbb{R}^n \mid a_j \leq x_j \leq b_j, j = \overline{1, n}\}$  –  $n$ -мерный параллелепипед. Тогда

$$\bar{x}_j^{(k)} = \begin{cases} a_j, & \text{если } \partial f(\mathbf{x}^{(k)})/\partial x_j > 0; \\ b_j, & \text{если } \partial f(\mathbf{x}^{(k)})/\partial x_j < 0; \\ (a_j + b_j)/2, & \text{если } \partial f(\mathbf{x}^{(k)})/\partial x_j = 0. \end{cases}$$

3. Допустимое множество  $U = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{y}^{(0)}\| \leq R\}$  – шар радиуса  $R$  с центром в точке  $\mathbf{y}^{(0)}$ . Тогда:

$$\bar{x}_j^{(k)} = y_j^{(0)} - R \frac{f'_j(\mathbf{x}^{(k)})}{\|\mathbf{f}'(\mathbf{x}^{(k)})\|}.$$

**Пример 7.9.** В задаче

$$f(\mathbf{x}) = \frac{1}{2} \left( x_1 - \frac{3}{4} \right)^2 + \frac{1}{2} \left( x_2 - \frac{3}{4} \right)^2 \rightarrow \min,$$

$$x_1 + x_2 \leq 1, \quad x_1, x_2 \geq 0$$

выполнить одну итерацию метода условного градиента из начальной точки  $x^{(0)} = (0, 0)$ .

**Решение**

1. Составим вспомогательную задачу (7.54) для точки  $x^{(0)}$ :

$$\bar{f}_0(x) = -\frac{3}{4}x_1 - \frac{3}{4}x_2 \rightarrow \min,$$

$$x_1 + x_2 \leq 1, \quad x_1, x_2 \geq 0.$$

Это задача линейного программирования. Симплекс-методом найдем ее решение  $\bar{x}^* = (0, 1)$ .

2. Составим задачу одномерной минимизации  $\Phi_0(\alpha) = f[x^{(0)} + \alpha(\bar{x}^{(0)} - x^{(0)})] = 0.5(-3/4)^2 + 0.5(\alpha - 3/4)^2 \rightarrow \min$ . Из условия  $\Phi'_0(\alpha) = 0$  найдем  $\alpha^* = 3/4$  и, следовательно,  $x^{(1)} = (0, 3/4)$ .

3. На этом шаге проверяют условие останова и либо прекращают вычисления, либо продолжают поиск точки минимума.

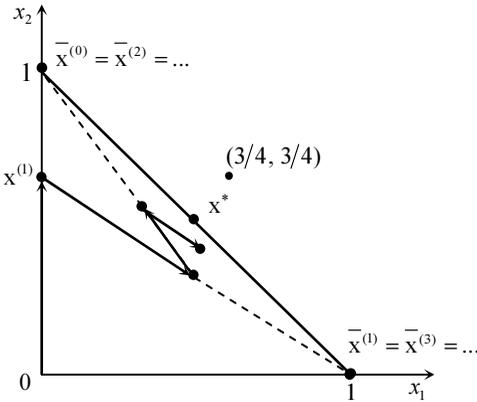


Рис. 7.5

На рис. 7.5 дана геометрическая иллюстрация решения примера 7.9 методом условного градиента. Обращает на себя внимание характерное зигзагообразное изменение направлений спуска. Это связано с тем, что при решении вспомогательных задач симплекс-методом в качестве  $\bar{x}^{(k)}$  будут получаться лишь угловые точки множества  $U$ , поэтому направления

$\bar{x}^{(k)} - x^{(k)}$  поиска очередной точки  $x^{(k+1)}$  могут сильно отличаться от направления на точку минимума  $x^*$ .

## 7.4. Методы последовательной безусловной минимизации

Основная идея этой группы методов состоит в преобразовании задачи нелинейного программирования  $f(x) \rightarrow \min, x \in U \subset R^n$  в последовательность задач безусловной минимизации

$$f_k(x) = f(x) + \varphi_k(x) \rightarrow \min, x \in R^n, k = 1, 2, \dots, \quad (7.57)$$

где  $\varphi_k(x)$  – функции, которые с ростом  $k$  во все большей степени учитывают ограничения, определяющие допустимое множество  $U$  исходной задачи.

В качестве приближенного решения исходной задачи берется решение  $x^{(k)}$  вспомогательной задачи (7.57), соответствующее достаточно большому значению  $k$ .

### Метод штрафных функций

В этом методе функции  $\varphi_k(x)$  подбираются так, чтобы при больших  $k$  функция  $f_k(x)$  из (7.57) мало отличалась от  $f(x)$  при  $x \in U$  и быстро возрастала при удалении точки  $x \notin U$  от допустимого множества. Идею метода штрафных функций иллюстрирует рис. 7.6, соответствующий, для наглядности, одномерному случаю.

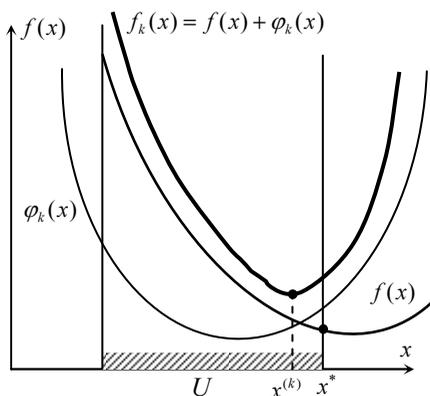


Рис. 7.6

**Определение 7.2.** Пусть  $U \subset R^n$  – заданное множество. Последовательность функций  $\{\varphi_k(x)\}$ , определенных в  $R^n$  и обладающих свойством

$$\lim_{k \rightarrow \infty} \varphi_k(x) = \begin{cases} 0, & \text{если } x \in U, \\ \infty, & \text{если } x \notin U \end{cases} \quad (7.58)$$

называется *последовательностью штрафных функций множества*  $U$ .

Пусть  $\{A_k\}$  – какая-либо возрастающая числовая последовательность с положительными членами и  $\lim_{k \rightarrow \infty} A_k = +\infty$ , а функция  $\varphi(x)$  обращается в нуль при  $x \in U$  и принимает положительные значения при  $x \notin U$ . Тогда последовательность  $\varphi_k(x) = A_k \varphi(x)$  удовлетворяет условию (7.58).

На практике часто полагают  $A_k = k$ ,  $k = 1, 2, \dots$ .

В качестве функции  $\varphi(x)$  можно взять, например,  $\rho(x, U)$  – расстояние от точки  $x$  до множества  $U$ . Тогда последовательность штрафных функций примет вид:

$$\varphi_k(x) = k \rho(x, U). \quad (7.59)$$

Вычисление расстояния  $\rho(x, U)$ , а следовательно, и значений штрафной функции  $A_k \rho(x, U)$  может быть затруднительным, поэтому часто применяют штрафные функции более удобного вида с учетом явного задания границ множества  $U$ .

Пусть множество  $U$  задано неравенствами

$$g_i(x) \leq 0, \quad i = \overline{1, m}, \quad (7.60)$$

где  $g_i(x)$  – выпуклые функции. Тогда в качестве  $\varphi(x)$  в выражении  $\varphi_k(x) = A_k \varphi(x)$  можно взять

$$\varphi(x) = \sum_{i=1}^m \Phi(g_i(x)),$$

где  $\Phi(t)$  – непрерывная функция, причем  $\Phi(t) = 0$ , если  $t \leq 0$  и  $\Phi(t) > 0$  при  $t > 0$ . Если  $\lim_{k \rightarrow \infty} A_k = +\infty$  и  $A_k > 0$ , то для последовательности  $\{\varphi_k(x)\}$ ,  $\varphi_k(x) = A_k \varphi(x)$  условия (7.58) выполняются, т.е. она является последовательностью штрафных функций. Функцию  $\Phi(t)$  можно выбрать так, чтобы функции  $\varphi_k(x)$  обладали свойствами, упрощающими решение вспомогательных задач минимизации (7.57), например, такими как выпуклость, существование производных, простота вычислений и т.д. Часто полагают

$$\varphi(x) = \sum_{i=1}^m [g_i^+(x)]^2,$$

где  $g_i^+(x) = \max\{0, g_i(x)\} = \frac{1}{2}(g_i(x) + |g_i(x)|)$ .

**Пример 7.10.** Решить задачу

$$f(x) = x_1^2 + (x_2 - 4)^2 \rightarrow \min ;$$

$$\begin{cases} -x_1 + x_2 - 2 \leq 0, \\ -x_1^2 + 2x_2 \leq 0 \end{cases}$$

методом штрафных функций.

**Решение.** Запишем последовательность вспомогательных задач минимизации:

$$f_k(x) = x_1^2 + (x_2 - 4)^2 + k \cdot [g_1^+(x)]^2 + k \cdot [g_2^+(x)]^2 \rightarrow \min ,$$

где

$$g_1^+ = \frac{1}{2}(-x_1 + x_2 - 2 + |-x_1 + x_2 - 2|), \quad g_2^+ = \frac{1}{2}(-x_1^2 + 2x_2 + |-x_1^2 + 2x_2|).$$

В табл. 7.2 приведены решения этих задач для некоторых  $k$ .

Точное решение задачи можно получить графическим методом.

Это решение:  $x^* = (1, 3)$ ,  $f(x^*) = f_{\min} = 2$ .

Графические построения здесь не приводятся.

Таблица 7.2

$k$	$x_1^{(k)}$	$x_2^{(k)}$
0	0	4
1	0.66	3.33
5	0.91	3.09
10	0.95	3.05
100	0.995	3.004

#### Замечания

1. В качестве критерия остановки вычислений в методе штрафных функций можно использовать неравенство  $\|x^{(k)} - x^{(k/2)}\| \leq \varepsilon$ , где  $\varepsilon > 0$  параметр точности,  $k$  – четное число. При его выполнении полагают  $x^* = x^{(k)}$ ,  $f^* = f(x^{(k)})$ .

2. Для оценки точности решения, полученного методом штрафных функций, можно использовать следующий факт. Пусть  $z$  – какая-либо внутренняя точка допустимого множества  $U$  задачи выпуклого программирования  $f(x) \rightarrow \min$ ,  $g_i(x) \leq 0$ ,  $i = \overline{1, m}$  (т.е.  $g_i(z) < 0$  при всех  $i$ ), а  $x^{(k)}$  – решение  $k$ -й вспомогательной задачи (7.57). Тогда справедлива следующая двухсторонняя оценка минимального значения функции  $f(x)$  на множестве  $U$ :

$$f(x^{(k)}) \leq f^* \leq f(\bar{x}^{(k)}),$$

где  $\bar{x}^{(k)} = \alpha_k z + (1 - \alpha_k)x^{(k)}$ , а  $\alpha_k$  – наименьшее из значений  $\alpha \in [0, 1]$ , при которых  $g_i(\alpha_k z + (1 - \alpha_k)x^{(k)}) \leq 0$  для всех  $i = \overline{1, m}$ .

На практике величину  $\alpha_k$  можно найти методом одномерного поиска.

3. Недостатком метода штрафных функций является плохая обусловленность вспомогательной минимизируемой функции  $f_k(x)$  при больших  $k$ .

В методе штрафных функций начальная точка может быть выбрана произвольно. Следует добавить, что этот метод является достаточно простым и в то же время эффективным методом решения задач мини-

мизации с ограничениями. Для минимизации вспомогательной целевой функции  $f_k(x)$  могут быть использованы мощные методы безусловной минимизации (см. гл. 5). Однако с его помощью трудно получить решение с высокой точностью. Дело в том, что, когда с ростом  $k$  целевая функция  $f_k(x)$  становится плохо обусловленной, замедляется скорость сходимости алгоритмов безусловной минимизации.

### Метод барьерных функций

В этом методе исходная задача нелинейного программирования также сводится к последовательности задач безусловной минимизации (7.57), но функции  $\varphi_k(x)$  выбираются таким образом, чтобы при больших  $k$  функции  $f_k(x)$  из (7.57) мало отличались от  $f(x)$  во внутренних точках  $x$  допустимого множества  $U$  и в то же время при приближении точки  $x \in U$  к границе множества  $U$  эти функции неограниченно возрастали.

**Определение 7.3.** Последовательность функций  $\{\varphi_k(x)\}$ , определенная во всех внутренних точках множества  $U$ , называется *последовательностью барьерных функций этого множества, если выполняются условия:*

$$1) \lim_{k \rightarrow \infty} \varphi_k(x) = 0 \text{ для любой фиксированной внутренней точки}$$

$x$  множества  $U$ ;

$$2) \lim_{r \rightarrow \infty} \varphi_k(x^{(r)}) = +\infty \text{ для любой последовательности } \{x^{(r)}\}$$

внутренних точек множества  $U$ , сходящейся к какой-либо граничной точке этого множества.

Таким образом, влияние барьерной функции  $\varphi_k(x)$  при больших  $k$  состоит в создании «барьера» с крутыми склонами вдоль границы допустимого множества.

Пусть  $\{B_k\}$  – какая-либо числовая последовательность с положительными членами и  $\lim_{k \rightarrow \infty} B_k = 0$ , а функция  $\varphi(x)$  определена во всех внутренних точках множества  $U$  и стремится к  $+\infty$  при приближении точки  $x$  к границе множества  $U$ . Тогда последовательность  $\varphi_k(x) = B_k \varphi(x)$  удовлетворяет определению 7.3.

На практике часто полагают  $B_k = 1/k$ ,  $k = 1, 2, \dots$ .

Если допустимое множество  $U$  задачи нелинейного программирования определяется системой неравенств

$$g_i(x) \leq 0, \quad i = \overline{1, m}, \quad (7.61)$$

то в качестве функции  $\varphi(x)$  можно взять, например,

$$\varphi(x) = -\sum_{i=1}^m \frac{1}{g_i(x)}. \quad (7.62)$$

Тогда вспомогательная функция  $f_k(x)$  из (7.57) принимает вид:

$$f_k(x) = f(x) - \frac{1}{k} \sum_{i=1}^m \frac{1}{g_i(x)}. \quad (7.63)$$

Хотя метод барьерных функций используется в задачах многомерной минимизации, рассмотрим для наглядности иллюстрацию этого метода в одномерном случае.

**Пример 7.11.** Решить задачу

$$\begin{aligned} f(x) = x &\rightarrow \min, \\ -x + 2 &\leq 0 \end{aligned}$$

методом барьерных функций, используя в качестве  $\varphi(x)$  функцию из (7.63).

**Решение.** Рассмотрим последовательность задач безусловной минимизации

$$f_k(x) = x + \frac{1}{k} \cdot \frac{1}{x-2}.$$

Начиная из произвольной точки допустимого множества, например  $x^{(0)} = 5$ , найдем ее решения при различных  $k$ :

$$x^{(1)*} = 3; \quad x^{(9)*} = 7/3; \quad x^{(100)*} = 21/10; \quad \dots$$

Нетрудно видеть, что при  $k \rightarrow \infty$   $x^{(k)*} \rightarrow x^* = 2$ .

Барьерные функции для допустимого множества  $U$  из (7.61) могут быть заданы и другими способами. Например,

$$\varphi(x) = \sum_{i=1}^m |g_i(x)|^{-p}, \quad p > 0$$

или 
$$\varphi(x) = -\sum_{i=1}^m \ln[-g_i(x)]. \quad (7.64)$$

Как и в методе штрафных функций, в качестве условия завершения вычислений можно использовать неравенство  $\|x^{(k)} - x^{(k/2)}\| \leq \varepsilon$ , где  $\varepsilon > 0$  параметр точности,  $k$  – четное число

**Пример 7.12.** Решить задачу

$$\begin{cases} f(x) = x_1^2 + 2x_2^2 \rightarrow \min; \\ g_1(x) = -x_1 + x_2 \leq 0, \\ g_2(x) = 1 - x_1 - x_2 \leq 0, \end{cases}$$

используя барьерные функции (7.64). Вычисления прекратить при  $\|x^{(k)} - x^{(k/2)}\| \leq 0.002$ .

**Решение.** Последовательность задач безусловной минимизации принимает вид:

$$f_k(x) = x_1^2 + 2x_2^2 - \frac{1}{k} [\ln(x_1 - x_2) + \ln(x_1 + x_2 - 1)] \rightarrow \min.$$

Решая их методом Ньютона из начальной точки  $x^{(0)} = (2, 0)$ , получим последовательность точек  $x^{(k)*}$ . При  $k = 500$  и  $k = 1000$  имеем соответственно

$$x^{(500)} = (0.6696, 0.3319) \text{ и } x^{(1000)} = (0.6682, 0.3326).$$

Так как  $\|x^{(1000)} - x^{(500)}\| = 1.65 \cdot 10^{-3} < 0.002$ , полагаем  $x^* = x^{(1000)} = (0.6682, 0.3326)$  и  $f^* = f(x^{(1000)}) = 0.6687$ . Точка  $x^*$  расположена вблизи границы множества  $U$ , задаваемого вторым ограничением.

Метод барьерных функций идейно близок к методу штрафных функций. Его иногда называют методом внутренних штрафов, так как поиск точки минимума ведется из внутренней точки допустимого множества, и величина штрафа возрастает при приближении к границе допустимого множества изнутри.

Для углубленного изучения вопросов минимизации можно использовать следующие пособия и монографии: [3, 7, 8, 11, 13].

## ЛИТЕРАТУРА

1. *Абрамов Л. М., Капустин В. Ф.* Математическое программирование. – Л.: Изд-во Ленингр. ун-та, 1981. – 328 с.
2. *Акулич И. Л.* Математическое программирование в примерах и задачах. – М.: Высш. шк., 1986. – с.
3. *Васильев Ф. П.* Численные методы решения экстремальных задач. – М.: Наука, 1980. – 520 с.
4. *Деордица Ю. С., Нефедов Ю. М.* Исследование операций в планировании и управлении. – Киев: Вища шк., 1991. – 270 с.
5. *Зайченко Ю. П.* Исследование операций. – К.: Вища шк., 1988. 552 с.
6. *Зуховицкий С. И., Авдеева Л. И.* Линейное и выпуклое программирование. – М.: Наука, 1964. – 348 с.
7. *Карманов В. Г.* Математическое программирование. – М.: Наука, 1986. – 288 с.
8. *Ляшенко И. Н.* и др. Линейное и нелинейное программирование. – Киев: Вища шк., 1975. – 372 с.
9. *Моисеев Н. Н., Иванюков Ю. П., Столярова Е. М.* Методы оптимизации. – М.: Наука, 1978. – 350 с.
10. *Мухачева, Э. А., Рубинштейн Г.Ш.* Математическое программирование. – Новосибирск: Наука, 1987. – 274 с.
11. *Сей Ж.* Оптимизация. Теория и алгоритмы. – М.: Мир, 1973. – 244 с.
12. *Сергиенко И. В.* Математические модели и методы решения задач дискретной оптимизации. – Киев: Наук. думка, 1985. – 384 с.
13. *Сухарев А. Г., Тимохов А. В., Федоров В. В.* Курс методов оптимизации. – М.: Наука, 1986. – 328 с.
14. *Химмельблау Д.* Прикладное нелинейное программирование. – М.: Мир, 1975. – 534 с.

Учебное пособие

НЕФЁДОВ Юрий Михайлович

## **МЕТОДЫ ОПТИМИЗАЦИИ**

Авторский оригинал-макет

Подписано к печати \_\_\_\_\_  
Формат 60С841/16 Бумага типограф. Гарнитура Times.  
Печать офсетная. Усл. печ. стр. \_\_\_\_\_. Уч.-изд. л. \_\_\_\_\_.  
Тираж \_\_\_\_ экз. Изд. № \_\_\_\_\_. Заказ № \_\_\_. Цена договорная.

Издательство Луганского национального университета  
имени Владимира Даля

**Адрес издательства:** 91034, г. Луганск, кв. Молодежный, 20а  
**Телефон:** 8 (0642) 41-34-12, **факс:** 8 (0642) 41-31-60  
**http:** www.dahluniver.ru