

519

Л59

МИФИ

МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)



ФАКУЛЬТЕТ
КИБЕРНЕТИКИ

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ В СОВРЕМЕННЫХ ЗАДАЧАХ ОПТИМИЗАЦИИ

Учебное пособие

Москва 2008

63666

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

**ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ
В СОВРЕМЕННЫХ ЗАДАЧАХ ОПТИМИЗАЦИИ**

Учебное пособие

Москва 2008

УДК 519.85(075)

ББК 22.18я7

Л159

Линейное программирование в современных задачах оптимизации. Учебное пособие / Ю.В.Бородакий, А.М. Загребаев, Н.А. Крицына, Ю.П. Кулябичев, Ю.Ю. Шумилов. – М.: МИФИ, 2008. – 188 с.

Приведены теоретические основы методов линейного и целочисленного математического программирования. Представлено большое количество практических задач, решение которых основано на использовании методов линейного и целочисленного программирования.

Предназначено для студентов и практикантов МИФИ, обучающихся по специальности «Прикладная математика и информатика», а также будет полезно инженерам и аспирантам, работающим в области оптимизации параметров технических систем различного назначения.

Рецензент д-р техн. наук, проф. А.Д. Модяев

Рекомендовано редсоветом МИФИ в качестве учебного пособия

ISBN 978-5-7262-0915-9

Всего 784609

72400

УДАА 56

© Московский инженерно-физический институт
(государственный университет), 2008

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
ОСНОВНЫЕ ОБОЗНАЧЕНИЯ И ПОНЯТИЯ	7
Глава I. ОСНОВЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ	13
1.1. Задача линейного программирования и ее геометрический смысл	13
1.2. Симплекс-метод	18
1.2.1. Порядок решения задач линейного программирования симплекс-методом	18
1.2.2. Алгоритм решения задач линейного программирования с использованием симплекс-таблиц	23
1.2.3. Примеры решения практических задач линейного программирования	25
1.3. Вырожденные задачи линейного программирования	33
1.3.1. Понятия вырожденности и заикливания решения задач линейного программирования	33
1.3.2. Антициклон	37
1.4. Метод искусственного базиса	39
1.4.1. Особенности метода	39
1.4.2. Примеры решения задач с использованием метода искусственного базиса	43
1.5. Двойственные задачи линейного программирования	53
1.5.1. Основные положения	53
1.5.2. Двойственный симплекс-алгоритм	58
1.5.3. Примеры решения двойственных задач линейного программирования	60
1.6. Транспортная задача линейного программирования	65
1.6.1. Постановка задачи	66
1.6.2. Нахождение первого опорного плана	68
1.6.3. Метод потенциалов	70
1.6.4. Транспортные задачи с неправильным балансом	77
1.7. Дискретное программирование	82
1.7.1. Постановка задачи	82
1.7.2. О решении задач линейного целочисленного программирования	86
1.7.3. Метод отсечения. Первый алгоритм Гомори	88
1.7.4. Метод ветвей и границ	92
1.7.5. Метод зондирования решений	95
1.7.6. Примеры решения задач целочисленного программирования	99

Глава 2. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ В ПРИКЛАДНЫХ ЗАДАЧАХ ОПТИМИЗАЦИИ	107
2.1. Применение линейного программирования в теоретико-игровых методах исследования сложных систем	107
2.1.1. Теоретические основы матричных игр	107
2.1.2. Сведение матричной игры к задаче линейного программирования	110
2.2. Использование методов линейного программирования при решении задач перебазирования средств истребительной авиации	113
2.2.1. Математическая постановка задачи перебазирования	113
2.2.2. Преобразование задачи целочисленного программирования к задаче линейного программирования	118
2.2.3. Тестовый пример задачи оптимального перегруппирования средств истребительной авиации	122
2.3. Оптимальное распределение запасов реактивности при работе системы ядерных реакторов в переменном суточном графике нагрузки	124
2.3.1. Физическая постановка задачи	124
2.3.2. Оптимальное распределение запасов реактивности в системе двух реакторов с линейной зависимостью возможной степени снижения мощности от запаса реактивности	128
2.4. Оптимизация размещения радиолокационных станций	135
2.4.1. Модели радиолокационных станций и основные этапы решения задачи	135
2.4.2. Используемые системы координат	139
2.4.3. Метод решения задачи оптимизации и результаты расчетов	145
2.5. Оптимальная кластеризация как задача линейного программирования	151
2.5.1. Математическая постановка задачи кластерного анализа	151
2.5.2. Математические критерии оптимальной кластеризации	157
2.5.3. Методы оптимальной кластеризации	160
2.5.4. Приближенные методы кластеризации	165
2.5.5. Зависимость времени и качества кластеризации от количества объектов, кластеров и размерности признакового пространства	173
СПИСОК ЛИТЕРАТУРЫ	184

ВВЕДЕНИЕ

Первооснову математического программирования составляют методы нахождения экстремума (максимума или минимума) функций, заданных на некоторой области изменения переменных. Задачи такого рода встречаются в самых разных областях человеческой деятельности: в экономике, технике, военном деле и т.д. Математическое программирование является одним из разделов науки, имеющей название «исследование операций».

Задачи, которые решаются с помощью методов математического программирования возникают там, где необходим выбор одного из возможных образов действий (или программ действий). Именно отсюда название «программирование» – в смысле выбор программы действий. Название не имеет ни какого отношения к программированию в смысле написания программ для ЭВМ и возникло раньше, чем ЭВМ. Хотя надо отметить, что все методы, в конечном счете, рассчитаны на использование ЭВМ. Первые задачи линейного программирования, являющегося основой математического программирования, были рассмотрены русским математиком Канторовичем ещё в 1939 г. (задача выбора наилучшей программы загрузки группы лущильных станков). Однако их практическое решение требовало большого объема вычислений. Создание высоко производительных ЭВМ послужило стимулом развития теории и практики как математического программирования, так и других разделов исследования операций. Суть у всех этих разделов одна – выбор наилучшей программы действий в задачах оптимизации с различными математическими моделями.

Так, в математическом программировании дается функция, которую нужно оптимизировать (т.е. найти экстремум) и ограничения на область изменения переменных; а в оптимальном управлении задаётся оптимизируемый функционал и процесс, описываемый с помощью дифференциальных уравнений и т.д.

Математическое программирование является одним из важнейших разделов, представляющих интерес для специалистов, занимающихся прикладной математикой, и изложено в той или иной мере в различных изданиях как в России, так и за рубежом, ряд из которых приведен в списке литературы.

Данная книга представляет собой результат многолетнего чтения курса «Математическое программирование» на кафедре «Математическое обеспечение систем» в Московском инженерно-физическом институте и помимо теоретических соображений, охватывающих разделы линейного, целочисленного и нелинейного программирования, содержит многоплановые примеры решений задач, иллюстрирующих применение методов математического программирования на практике.

В книге рассматривается многообразие прикладных задач оптимизации, решение которых обеспечивается методами математического программирования, независимо от их физического содержания. Так, особенностью книги является единый подход к оптимизации игровых задач, задач управления движением, задач распределения запасов реактивности в ядерных реакторах, а также задач, связанных с оптимальной кластеризацией и т.д.

Каждый раздел книги содержит большое количество доведенных до конечного результата примеров.

Книга предназначена для специалистов по прикладной математике, а также может быть полезна студентам вузов соответствующих специальностей.

ОСНОВНЫЕ ОБОЗНАЧЕНИЯ И ПОНЯТИЯ

Приведем ряд общепринятых обозначений и понятий, используемых при рассмотрении материала последующих глав.

1. Пусть \bar{x} и \bar{y} – n -мерные векторы

$$\bar{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \bar{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix},$$

$\bar{x}^T = (x_1, \dots, x_n)$ – транспонированный вектор,

тогда $\langle \bar{x}, \bar{y} \rangle = \bar{x}^T \bar{y} = \sum_{i=1}^n x_i y_i$ – скалярное произведение векторов \bar{x} и \bar{y} .

2. $A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$ – матрица размерности $m \times n$.

3. $f(\bar{x}) = f(x_1, \dots, x_n)$ – функция n переменных.

Функция $f(\bar{x})$ имеет 1-ю и 2-ю непрерывные производные на множестве D в n -мерном пространстве R^n .

Пусть некоторая точка $\bar{x}^0 \in D$. Тогда $f(\bar{x})$ может быть представлена в виде ряда Тейлора относительно этой точки:

$$\begin{aligned} f(x_1, \dots, x_n) &= f(x_1^0, \dots, x_n^0) + \sum_{i=1}^n \left. \frac{\partial f}{\partial x_i} \right|_{x_0} (x_i - x_i^0) + \\ &+ \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \left. \frac{\partial^2 f}{\partial x_i \partial x_j} \right|_{x_0} (x_i - x_i^0)(x_j - x_j^0) + O(\|\bar{x} - \bar{x}^0\|^2). \end{aligned}$$

4. $\nabla f(\bar{x}^0) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^T \Big|_{x_0}$ – градиент функции в точке

$\bar{x}^0 \in D$.

Градиент это вектор, который направлен в сторону наискорейшего роста функции и ортогонален в точке x^0 к поверхности (или линии) уровня $f(\bar{x}) = \text{const}$ (рис. В.1).



Рис. В.1. Линии уровня $f(\bar{x}) = \text{const}$ для функции двух переменных

5. $H(\bar{x}^0) = \begin{vmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{vmatrix}$ – вещественная симметричная

матрица (матрица Гессе).

Важность градиента и матрицы Гессе определяется их использованием во многих алгоритмах поиска экстремума функций.

Можно записать

$$f(\bar{x}) = f(\bar{x}^0) + \langle \nabla f(\bar{x}^0), (\bar{x} - \bar{x}^0) \rangle + \frac{1}{2} \langle (\bar{x} - \bar{x}^0),$$

$$H(\bar{x}^0)(\bar{x} - \bar{x}^0) \rangle + o(\|\bar{x} - \bar{x}^0\|^2).$$

6. Пусть

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1j}x_j + \dots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2j}x_j + \dots + a_{2n}x_n = b_2; \\ \dots \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ij}x_j + \dots + a_{in}x_n = b_i; \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mj}x_j + \dots + a_{mn}x_n = b_m \end{array} \right. -$$

система линейных уравнений в скалярном виде, тогда $A\bar{x} = \bar{b}$ – система уравнений в матричной форме, где A – матрица коэффициентов a_{ij} ($i = 1, \dots, m$; $j = 1, \dots, n$), \bar{b} – m -мерный вектор.

Система линейных уравнений называется *совместной*, если она имеет хотя бы одно решение.

Пусть r – ранг матрицы и представляет собой наибольший порядок отличного от нуля определителя матрицы A (число линейно независимых столбцов матрицы A).

$r(A) = r(Ab)$ – необходимое и достаточное условие совместности системы линейных уравнений, т.е. ранг матрицы A должен быть равен рангу распределенной матрицы системы.

Ранг не превосходит числа неизвестных n , $r \leq n$.

Тогда:

если $r = n$, то решение системы единственно;

если $r < n$, то система имеет бесчисленное множество решений.

7. $a_0 + a_1x_1 + \dots + a_nx_n \geq 0$ – линейное неравенство.

Совокупность точек пространства, координаты которых удовлетворяют этому неравенству, представляют собой полупространство; совокупность нескольких линейных неравенств определяет область решения, представляющую собой выпуклый многогранник.

8. Общая задача математического программирования формулируется так:

найти максимум (минимум) функции $f(x)$ при ограничениях

$$g_i(x_1, \dots, x_j, \dots, x_n) \leq b_i, \quad i = \overline{1, m}; \quad (B.1)$$

$$x_j \geq 0, \quad j = \overline{1, n}. \quad (B.2)$$

При этом:

функция $f(\bar{x})$ называется *целевой*;

система неравенств (B.1) и условия неотрицательности переменных (B.2) называются *системой ограничений задачи*;

всякое решение задачи с учетом системы ограничений, т.е. совокупность значений переменных x_1, \dots, x_n , удовлетворяющих условиям (B.1) и (B.2), называется *допустимым решением*;

совокупность точек n -мерного пространства, удовлетворяющих системе ограничений, образует так называемую *допустимую область*;

допустимое решение, максимизирующее (минимизирующее) целевую функцию, называется *оптимальным*.

Итак, задача математического программирования заключается в нахождении оптимального решения, обеспечивающего максимальное (минимальное) значение целевой функции с учетом системы ограничений на переменные.

Замечания. Постановка задачи математического программирования является достаточно общей. Если какая-то практическая задача, на первый взгляд, кажется иной, то, как правило, с помощью простых математических преобразований её можно свести к задаче рассматриваемого вида.

1. Если требуется найти минимум функции $f(\bar{x})$, то это эквивалентно поиску максимума функции $-f(\bar{x})$, т.е. $\min_x f(\bar{x}) = -\max_x [-f(\bar{x})]$.

2. Если заданы неравенства вида

$$g(\bar{x}) \geq \bar{b},$$

то простой переменной знака можно прийти к виду (B.1), а именно

$$-g(\bar{x}) \leq -\bar{b}.$$

3. Если на переменные x_j не наложено условие неотрицательности, а задано ограничение

$$x_j \geq x_{j, \min},$$

то, вводя замену переменных

$$x'_j = x_j - x_{j, \min},$$

получаем для новой переменной условие (B.1), т.е. $x'_j \geq 0$.

4. Функция $f(\bar{x})$ может иметь несколько экстремумов, а именно: локальные и глобальный экстремумы (пусть это будет максимум).

Функция $f(\bar{x})$, определенная на области D , достигает на ней глобального максимума в точке $\bar{x}^0 \in D$, если неравенство $f(\bar{x}) \leq f(\bar{x}^0)$ справедливо для любой точки $\bar{x} \in D$.

Функция $f(\bar{x})$ достигает локального максимума в точке $\bar{x}^0 \in D$, если неравенство $f(\bar{x}) \leq f(\bar{x}^0)$ справедливо для любой точки \bar{x} из некоторой окрестности точки \bar{x}^0 .

5. В математическом анализе для нахождения экстремума функций используются производные. Эти методы называются *классическими методами оптимизации*.

Различают задачи безусловного и условного экстремумов.

Задача на безусловный экстремум — задача максимизации функции $f(\bar{x})$ при отсутствии ограничений. Необходимые условия экстремума в этом случае записываются в виде системы уравнений:

$$\frac{\partial f(x_1, \dots, x_n)}{\partial x_j} = 0, \quad j = \overline{1, n}.$$

Решая данную систему уравнений, находят стационарные точки, а затем после их анализа — оптимальную точку.

Задача на условный экстремум предусматривает ограничения в виде равенств: т.е. минимизировать $f(\bar{x})$ при ограничениях $g(\bar{x}) = \bar{b}$.

В векторной форме g и \bar{b} — m -мерные векторы.

Эта задача сводится к задаче на безусловный экстремум с помощью множителей Лагранжа. С этой целью образуют функцию Лагранжа:

$$L(\bar{x}, \lambda) = f(\bar{x}) + \sum_{i=1}^m \lambda_i [b_i - g_i(x)],$$

где λ_i — множители Лагранжа.

Функция $L(\bar{x}, \lambda)$ зависит от $m+n$ переменных, на которые наложены ограничения.

Оптимальную точку находят в результате решения и последующего анализа системы уравнений:

$$\begin{cases} \frac{\partial L}{\partial x_j} = 0, \quad j = \overline{1, n}, & \frac{\partial f}{\partial x_j} + \sum_{i=1}^m \lambda_i \frac{\partial g_i}{\partial x_j} = 0; \\ \frac{\partial L}{\partial \lambda_i} = 0, \quad i = \overline{1, m}, & b_i - g_i(x) = 0. \end{cases}$$

6. Классические методы оптимизации применяют лишь для решения сравнительно простых задач математического программирования, так как они имеют ряд недостатков.

- Для использования методов нужно, чтобы функции $f(\bar{x})$ и $g(\bar{x})$ были непрерывны и имели частные производные, по крайней мере, второго порядка. Поскольку на практике функции не редко задаются не аналитически, а таблично или иными, более сложными способами, то это затрудняет вычисление и анализ производных.

- Применение классических методов связано с большим объемом вычислений. При этом, во-первых, необходимо решить систему алгебраических уравнений, которая в большинстве случаев нелинейна. Это, в свою очередь, достаточно сложная задача. Во-вторых, для анализа стационарных точек требуется вычислять вторые производные целевой функции, что часто затруднительно из-за громоздкости выражений.

- С помощью классических методов можно найти экстремум функции только внутри области.

Если оптимальная точка находится на границе области, то методы эти бессильны.

Итак, поскольку классические методы оптимизации не всегда приводят к желаемому результату, необходимы принципиально новые методы, составляющие предмет математического программирования.

Глава 1

ОСНОВЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

1.1. Задача линейного программирования и ее геометрический смысл

Постановка задачи. Найти максимум функции

$$f(\bar{x}) = c, \bar{x} > +c_0$$

при ограничениях

$$\begin{aligned} A\bar{x} &\leq \bar{b}; \\ \bar{x} &\geq 0, \end{aligned} \tag{1.1}$$

где \bar{x} — n -мерный вектор; \bar{c} — n -мерный вектор; A — матрица $m \times n$; \bar{b} — m -мерный вектор; c_0 — вещественное число.

Суть задачи состоит в том, что необходимо найти неотрицательные значения компонентов вектора \bar{x} , удовлетворяющие системе линейных неравенств, при которых линейная целевая функция достигает максимума.

Рассмотрим упрощенные модели некоторых прикладных задач, приводящих к задачам линейного программирования.

Пример 1.1. Пусть некоторая фирма выпускает n видов продукции и использует при этом m видов сырья. Причем ресурсы сырья ограничены: i -го вида сырья не более b_i единиц. На изготовление одной единицы продукции j -го вида (всего изготавливается $x_j \geq 0, j = 1, n$) тратится a_{ij} единиц i -го сырья, т.е. по i -му виду сырья должно выполняться условие:

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

Известно также, что на каждой единице продукции j -го вида фирма получает c_j единиц прибыли, т.е. общая прибыль составляет:

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

Требуется определить, сколько единиц x_j каждого вида продукции должна изготавливать фирма, чтобы получить максимальную прибыль.

Пример 1.2. У фермера имеется m участков земли, площадь каждого составляет b_i гектара ($i = \overline{1, m}$). Фермер собирается посеять n культур — x_{ij} ($j = \overline{1, n}$) — площадь j -й культуры на i -м участке.

Прибыль с одного центнера j -й культуры составляет c_j рублей.

Фермер хочет получить максимальную прибыль.

Известно, что средняя урожайность j -й культуры на i -м участке равна a_{ij} центнеров с гектара.

Кроме того, по каким-то причинам фермер должен получить каждой культуры не менее d_j центнеров (или не более).

Сформулируем задачу:

определить максимальную суммарную прибыль

$$\sum_{j=1}^n \left(c_j \sum_{i=1}^m a_{ij} x_{ij} \right)$$

при ограничениях на переменные x_{ij}

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

площадь i -го участка

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

и урожай j -й культуры

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

$$\left(\sum_{i=1}^m x_{ij} a_{ij} \geq d_j \right).$$

К задачам линейного программирования также сводятся и многие другие прикладные задачи технико-экономического содержания.

Следует заметить, что приведенные примеры представляют лишь приближенную, упрощенную математическую модель реальных задач, и некритическое использование получаемых на основе анализа этих моделей результатов может привести иногда к парадоксам.

Это относится не только к задачам линейного программирования, но и к другим математическим моделям. Вполне может оказаться, что принятая математическая модель, обычно составленная на основе приближенных данных о реальном моделируемом явлении, не охватывает какие-либо важные существенные стороны исследуемого явления, что приводит к результатам, существенно расходящимся с реальностью. Однако не нужно делать вывод, что линейные модели слишком просты и вовсе непригодны для исследования реальных задач. (Следует помнить, что существует противоречие: сложная адекватная математическая модель, учитывающая много факторов, с одной стороны, и возможность решения задачи, с другой.) Наоборот, практика показывает, что линейное программирование может быть успешно применено для исследования широкого класса реальных технических и экономических задач. Кроме того, задачи линейного программирования часто используются в качестве вспомогательных во многих методах решения более сложных нелинейных задач оптимизации.

Рассмотрим *геометрический смысл* задачи линейного программирования (на примере функции двух переменных).

Как отмечено выше, допустимая область задачи линейного программирования геометрически изображается пересечением полуплоскостей, определенных линейными неравенствами системы ограничений и представляет собой *выпуклый многоугольник* (рис. 1.1).

Линии уровня линейной целевой функции – это прямые линии $c_1 x_1 + c_2 x_2 + c_0 = \text{const}$, параллельные друг другу, они перпендикулярны вектору \bar{c} , который равен градиенту линейной функции.

Оптимальной является точка, лежащая на линии уровня, принимающей наибольшее значение.

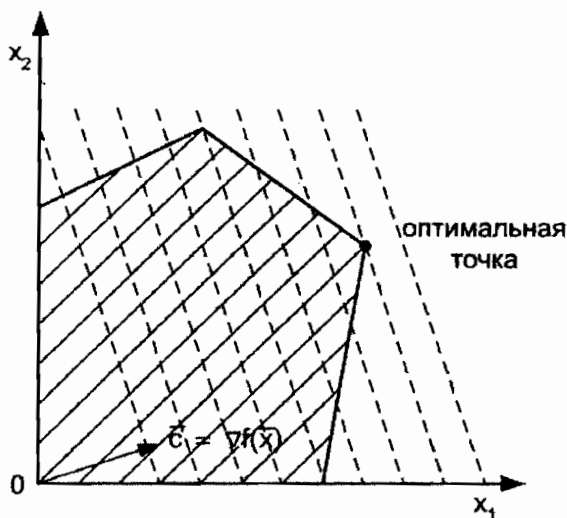


Рис. 1.1. Допустимая область – выпуклый многоугольник (оптимальная точка на пересечении ребер)

Такая точка не может быть внутренней, она всегда лежит на границе, причем соответствует именно вершине многоугольника, за исключением случая, когда линии уровня параллельны некоторому ребру многоугольника (рис. 1.2).

В этом случае задача имеет бесчисленное множество оптимальных точек, принадлежащих указанному ребру.

Задача линейного программирования не имеет решения, если система ограничений несовместна, т.е. допустимое множество пусто (рис. 1.3).

Задача не имеет решения также, если в направлении увеличения функции (рис. 1.4) допустимая область не замкнута.

Если же в направлении увеличения функции (см. рис. 1.4) допустимая область замкнута, то задача решение имеет.

Итак, из геометрического смысла задачи линейного программирования следует, что, в общем случае, оптимальное решение находится в одной из вершин допустимой многогранной области. При решении задачи осуществляется переход от одной вершины допус-

тимой области к другой, соседней с ней вершине, причем такой, в которой значение целевой функции больше. Такое движение составляет сущность симплекс-метода – основного метода решения задач линейного программирования.

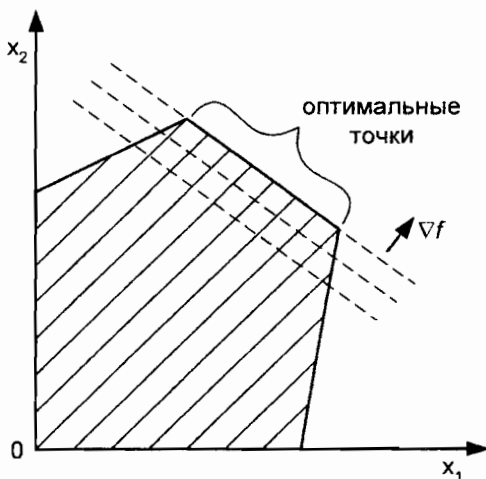


Рис. 1.2. Частный случай допустимой области (оптимальная точка на ребре)

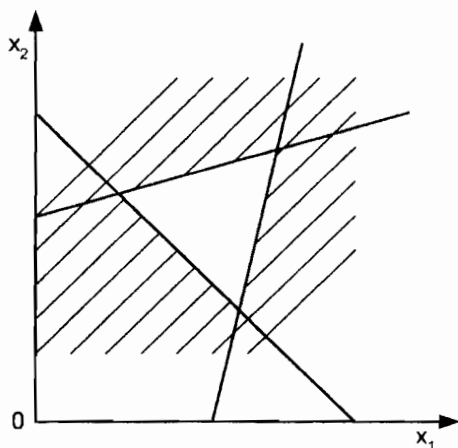


Рис. 1.3. Допустимая область – множество пусто

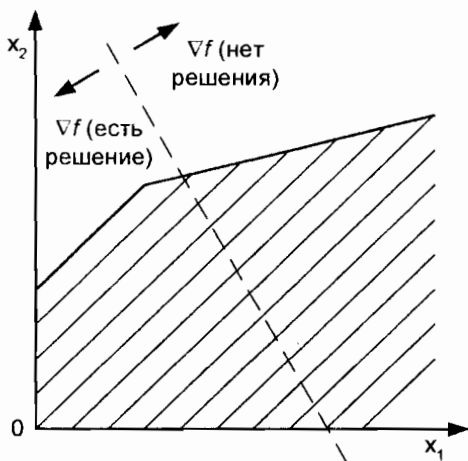


Рис. 1.4. Допустимая область не замкнута

1.2. Симплекс-метод

1.2.1. Порядок решения задач линейного программирования симплекс-методом

Для решения симплекс-методом задача (1.1) должна быть приведена к *стандартному (каноническому)* виду:

найти $\max f(\bar{x}) = \langle \bar{c}, \bar{x} \rangle + c_0$ при ограничениях

$$\begin{aligned} A\bar{x} &= \bar{b}, \\ x_{ij} &\geq 0, \quad i = \overline{1, n}, \quad j = \overline{1, m}. \end{aligned} \quad (1.2)$$

Приведение задачи к стандартному виду осуществляется с помощью введения дополнительных (ослабляющих) переменных:

$$1) \quad A\bar{x} \leq \bar{b} \rightarrow A\bar{x} + \bar{x}' = \bar{b};$$

$$2) \quad A\bar{x} \geq \bar{b} \rightarrow A\bar{x} - \bar{x}' = \bar{b},$$

где \bar{x}' – неотрицательный вектор.

Пример. 1.3.

$$f(\bar{x}) = 3x_1 - 2x_2 + x_3; \quad f(\bar{x}) = 3x_1 - 2x_2 + x_3 + 0x_4 + 0x_5;$$

$$\begin{cases} 2x_1 - 3x_3 \leq 1; \\ 3x_1 + 4x_2 + x_3 = 6; \\ x_1 + 2x_2 - x_3 - x_5 \geq 2; \\ x_j \geq 0, \quad j = \overline{1,3}. \end{cases} \rightarrow \begin{cases} 2x_1 - 3x_3 + x_4 = 1; \\ 3x_1 + 4x_2 + x_3 = 6; \\ x_1 + 2x_2 - x_3 - x_5 = 2; \\ x_j \geq 0, \quad j = \overline{1,5}. \end{cases}$$

В результате введения ослабляющих переменных получается задача, эквивалентная исходной, так как эти переменные не влияют на значение целевой функции – они входят в неё с нулевыми коэффициентами.

С практической точки зрения можно считать, что ранг системы ограничений (1.2) равен числу уравнений m , т.е. $r = m$. В противном случае хотя бы одно из уравнений представляет собой линейную комбинацию остальных и поэтому является лишним.

Кроме того, как было отмечено выше, ранг $r \leq n$ (чаще всего $r < n$) и система имеет бесчисленное множество решений (если, конечно, она вообще совместна).

Выделим среди n переменных первые r и назовем их *базисными переменными*. Остальные $n - r$ переменные назовем *свободными*.

Итак, x_1, x_2, \dots, x_r – базисные, $x_{r+1}, x_{r+2}, \dots, x_n$ – свободные переменные.

Систему ограничений можно разрешить относительно базисных переменных:

$$\begin{cases} x_1 = \beta_1 - (\alpha_{1r+1}x_{r+1} + \alpha_{1r+2}x_{r+2} + \dots + \alpha_{1n}x_n), \\ \dots \\ x_i = \beta_i - (\alpha_{ir+1}x_{r+1} + \alpha_{ir+2}x_{r+2} + \dots + \alpha_{in}x_n), \\ \dots \\ x_r = \beta_r - (\alpha_{rr+1}x_{r+1} + \alpha_{rr+2}x_{r+2} + \dots + \alpha_{rn}x_n). \end{cases} \quad (1.3)$$

Свободные переменные в выражении (1.3) могут принимать любые значения. Следовательно, система ограничений имеет бесчисленное множество решений. Задача заключается в том, чтобы среди этого бесчисленного множества выбрать такое решение, которое доставляет максимум целевой функции.

Следует отметить, что свободные переменные выбираются таким образом, чтобы коэффициенты β_i были неотрицательны, т.е. $\beta_i \geq 0$.

Во многих задачах разрешение системы ограничений относительно базисных переменных до вида (1.3) является тривиальным. В некоторых задачах это сделать сложно, но существует так называемый метод искусственного базиса, который помогает это сделать (будет рассмотрен ниже).

Выразим также целевую функцию через свободные переменные:

$$f(\bar{x}) = \gamma_0 - (\gamma_{r+1}x_{r+1} + \gamma_{r+2}x_{r+2} + \dots + \gamma_n x_n). \quad (1.4)$$

Возьмем все свободные переменные равными нулю:

$$x_{r+1} = x_{r+2} = \dots = x_n = 0,$$

тогда из выражения (1.3) следует, что значения базисных переменных соответственно равны:

$$x_1 = \beta_1, \dots, x_i = \beta_i, \dots, x_r = \beta_r.$$

Полученное таким образом решение системы ограничений

$$\bar{x}_{61} = (\beta_1, \beta_2, \dots, \beta_r, 0, \dots, 0)^T$$

является допустимым, так как все его компоненты неотрицательны ($\beta_i \geq 0$).

Такое решение называют базисным (в базисном решении отличны от нуля ровно r переменных). В этом случае значение целевой функции

$$f(\bar{x}_{61}) = \gamma_0.$$

При таком подходе решение задачи линейного программирования заключается в последовательном переходе от одного базисного решения к другому (базисные решения соответствуют вершинам допустимого многогранника), при котором значение целевой функции монотонно возрастает.

Если проанализировать выражение (1.4), то видно, что значение целевой функции можно увеличить только в случае, если имеются отрицательные коэффициенты γ_j . Если все коэффициенты $\gamma_j \geq 0$, то значение целевой функции увеличить нельзя. Поэтому *признаком оптимальности решения* поставленной задачи максимизации

является неотрицательность всех коэффициентов при свободных переменных в выражении (1.4).

Если среди коэффициентов есть хотя бы один отрицательный, например γ_j , тогда за счет повышения x_j можно увеличить значение $f(\bar{x})$. Однако это возможно в определенных пределах, так как выбор x_j влияет на базисные переменные, которые могут стать недопустимыми, т.е. отрицательными.

Выразим базисные переменные из системы (1.3), считая все свободные переменные, за исключением x_j , равными нулю:

$$\begin{cases} x_1 = \beta_1 - \alpha_{1j}x_j; \\ \dots \\ x_i = \beta_i - \alpha_{ij}x_j; \\ \dots \\ x_r = \beta_r - \alpha_{rj}x_j. \end{cases}$$

При этом $f(\bar{x}) = \gamma_0 - \gamma_j x_j$.

Если все коэффициенты $\alpha_{ij} \leq 0$, $i = \overline{1, r}$, то увеличение x_j может быть неограниченным, это приводит к неограниченному увеличению базисных переменных и они никогда не станут отрицательными. Но это приводит и к бесконечному возрастанию целевой функции, т.е. в этом случае задача не имеет решения.

Итак, признаком того, что задача не имеет решения является условие

$$\alpha_{ij} \leq 0, \quad i = \overline{1, r},$$

где индекс j соответствует $\gamma_j < 0$.

Рассмотрим случай, когда имеются $\alpha_{ij} > 0$. Тогда увеличение x_j приводит к тому, что базовая переменная x_i будет уменьшаться до тех пор, пока не станет равной нулю при некотором значении x_j . Это значение определяется уравнением

$$x_i = \beta_i - \alpha_{ij}x_j = 0.$$

Отсюда $x_j = \frac{\beta_i}{\alpha_{ij}}$.

Если несколько $\alpha_{ij} > 0$, то первой обратится в нуль переменная x_l , для которой отношение $\frac{\beta_l}{\alpha_{lj}}$ минимально, т.е.

$$\frac{\beta_l}{\alpha_{lj}} = \min_i \left\{ \frac{\beta_i}{\alpha_{ij}} \right\} = \rho \quad (\text{минимум берется по всем } i, \text{ для которых } \alpha_{ij} > 0).$$

Когда x_l обратится в нуль, остальные базовые переменные будут еще принимать неотрицательные значения.

Элемент α_{lj} называют *разрешающим*. Он определяет переменную x_l , которую выводят из базиса \bar{x}_{61} , и указывает свободную переменную x_j , вводимую в новый базис \bar{x}_{62} :

$$\begin{cases} x_{r+1} = \dots = x_n = 0, & \text{кроме } x_j = \rho, \\ x_l = \beta_l - \alpha_{lj}x_j, \\ \dots \\ x_l = 0, \\ \dots \\ x_r = \beta_r - \alpha_{rj}x_j. \end{cases}$$

Значение целевой функции $f(\bar{x}_{62}) = \gamma_0 - \gamma_j \rho > f(x_{61})$, так как $\gamma_j < 0$, а $\rho > 0$.

Может оказаться, что $\rho = 0$, тогда целевая функция сохраняет предыдущее значение, хотя переменные базиса меняются. Такой случай называется *вырожденным* и встречается редко, для него существуют специальные методы решения задачи.

Поскольку в базис введена новая переменная x_j и выведена x_l , выразим, как и прежде, базисную переменную x_j через свободные переменные. Для этого понадобится уравнение из системы (1.3)

$$x_l = \beta_l - (\alpha_{lr+1}x_{r+1} + \dots + \alpha_{lj}x_j + \dots + \alpha_{ln}x_n).$$

Из этого уравнения получим

$$x_j = \frac{\beta_l}{\alpha_{lj}} - \left(\frac{\alpha_{lr+1}}{\alpha_{lj}} x_{r+1} + \dots + \frac{1}{\alpha_{lj}} x_l + \dots + \frac{\alpha_{ln}}{\alpha_{lj}} x_n \right).$$

Подставив x_j в остальные уравнения системы (1.3), получим выражения для нового базиса:

$$x_i = \beta'_i - (\alpha'_{ir+1}x_{r+1} + \dots + \alpha'_{ij}x_l + \dots + \alpha'_{in}x_n),$$

где на j -м месте стоит x_l — новая свободная переменная.

Целевая функция через новые свободные переменные запишется в виде

$$f(\bar{x}) = \gamma'_0 - (\gamma'_{r+1}x_{r+1} + \dots + \gamma'_jx_l + \dots + \gamma'_nx_n).$$

Далее все рассуждения повторяются снова, т.е. исследуются знаки коэффициентов γ'_j и проводятся соответствующие преобразования до получения окончательного результата.

Для облегчения процесса поиска оптимального решения используются специальные таблицы, называемые *симплекс-таблицами*.

1.2.2. Алгоритм решения задач линейного программирования с использованием симплекс-таблиц

1. Каноническая система ограничений $A\bar{x} = \bar{b}$ приводится к виду, в котором базисные переменные выражены через свободные, т.е. к виду

$$\begin{cases} x_1 = \beta_1 - (\alpha_{1r+1}x_{r+1} + \alpha_{1r+2}x_{r+2} + \dots + \alpha_{1n}x_n); \\ \dots \\ x_r = \beta_r - (\alpha_{rr+1}x_{r+1} + \alpha_{rr+2}x_{r+2} + \dots + \alpha_{rn}x_n), \end{cases}$$

а целевая функция к виду (1.4):

$$f(\bar{x}) = \gamma_0 - (\gamma_{r+1}x_{r+1} + \gamma_{r+2}x_{r+2} + \dots + \gamma_nx_n).$$

2. Заполняется симплекс-таблица (табл. 1.1).

Базис	Свободные члены	Переменные									
		x_1	...	x_i	...	x_r	x_{r+1}	...	x_j	...	x_n
x_1	β_1	1	...	0	...	0	α_{1r+1}	...	α_{1j}	...	α_{1n}
...
x_i	β_i	0	...	1	...	0	α_{ir+1}	...	α_{ij}	...	α_{in}
...
x_r	β_r	0	...	0	...	1	α_{rr+1}	...	α_{rj}	...	α_{rn}
$f(\bar{x})$	γ_0	0	...	0	...	0	γ_{r+1}	...	γ_j	...	γ_n

Каждая строка табл. 1.1 соответствует уравнению, а последняя строка соответствует целевой функции.

3. В строке коэффициентов целевой функции (не считая γ_0) выбирается $\gamma_j < 0$, обычно максимальным по модулю.

Если все $\gamma_j \geq 0$, то оптимальное решение достигнуто, причем значения переменных определяются столбцом свободных членов, а оптимальное значение функции – клеткой, соответствующей свободному члену.

4. В j -м столбце среди положительных коэффициентов α_{ij} выбирается разрешающий элемент α_{lj} , т.е. элемент для которого минимально отношение $\frac{\beta_l}{\alpha_{lj}}$.

Если положительных коэффициентов нет, то задача не имеет решения.

5. Делятся все члены строки, содержащей разрешающий элемент, на α_{lj} . Полученная строка вносится на то же место в новой таблице.

6. Из каждой оставшейся i -й ($i \neq l$) строки вычитается получившаяся строка, умноженная на коэффициент при x_j в i -й стро-

ке. В результате в клетках, соответствующих j -му столбцу, появляются нули. Преобразованные строки записываются в новой таблице на место прежних.

Переменные x_j и x_l в новой таблице меняются местами, т.е. x_j вводится в базис вместо x_l .

Далее идет переход на шаг 3.

1.2.3. Примеры решения практических задач линейного программирования

Пример 1.4. Составить математическое описание следующих задач в виде задач линейного программирования.

Задача 1.1. Из одного города в другой ежедневно отправляются пассажирские и скорые поезда. В табл. 1.2 указан состав поездов каждого типа, количество вагонов в парке и максимальное число пассажиров, на которое рассчитан вагон каждого вида.

Таблица 1.2

Поезда	Вагоны				
	Багажный	Почтовый	Плацкартный	Купейный	Спальный
Скорый	1	1	5	6	3
Пассажирский	1	—	8	4	1
Число вагонов	12	8	81	70	26
Число пассажиров в вагоне	$\alpha_1 = 0$	$\alpha_2 = 0$	$\alpha_3 = 58$	$\alpha_4 = 40$	$\alpha_5 = 32$

Определить число скорых x_1 и пассажирских x_2 поездов, которые необходимо формировать ежедневно из имеющихся вагонов, чтобы число перевозимых пассажиров было максимально.

Решение. Целевая функция:

$$f(\bar{x}) = c_1 x_1 + c_2 x_2,$$

где c_1 — число пассажиров в скором вагоне; c_2 — число пассажиров в пассажирском вагоне.

Найти $\max f(\bar{x})$ при ограничениях:

$$1x_1 + 1x_2 \leq 12 \quad \text{— ограничения на число багажных вагонов;}$$

- $1x_1 + 0x_2 \leq 8$ – ограничения на число почтовых вагонов;
 $5x_1 + 8x_2 \leq 81$ – ограничения на число плацкартных вагонов;
 $6x_1 + 4x_2 \leq 70$ – ограничения на число купейных вагонов;
 $3x_1 + 1x_2 \leq 26$ – ограничения на число спальных вагонов.

При этом

$$c_1 = 1 \cdot \alpha_1 + 1 \cdot \alpha_2 + 5 \cdot \alpha_3 + 6 \cdot \alpha_4 + 3 \cdot \alpha_5;$$

$$c_2 = 1 \cdot \alpha_1 + 0 \cdot \alpha_2 + 8 \cdot \alpha_3 + 4 \cdot \alpha_4 + 1 \cdot \alpha_5.$$

Задача 1.2. В начале рабочего дня из автобусного парка выходят на линию x_1 автобусов, через час ещё x_2 автобусов, ещё через час – ещё x_3 автобусов.

Каждый автобус работает непрерывно 8 ч. Рабочий день составляет 10 ч.

Минимально необходимое число машин на линии в i -й час рабочего дня ($i = \overline{1, 10}$) равно b_i . Превышение этого числа приводит к дополнительным издержкам: c_i рублей в течение i -го часа на каждый дополнительный автобус.

Определить количество автобусов x_1, x_2, x_3 , выходящих на линию в первые часы рабочего дня, так, чтобы минимизировать дополнительные издержки в течение всего рабочего дня.

Решение. Составим табл. 1.3.

Таблица 1.3

Часы	1	2	3	...	8	9	10
Минимальное количество автобусов	b_1	b_2	b_3	...	b_8	b_9	b_{10}
Количество автобусов	x_1	x_1 + x_2	x_1 + x_2 + x_3	...		x_2 + x_3	x_3

С учетом табл. 1.3 целевая функция может быть представлена в виде

$$\begin{aligned}
 f(x) &= (x_1 - b_1)c_1 + (x_1 + x_2 - b_2)c_3 + (x_1 + x_2 + x_3 - b_3)c_3 + \dots + \\
 &+ (x_1 + x_2 + x_3 - b_8)c_8 + (x_2 + x_3 - b_9)c_9 + (x_3 - b_{10})c_{10} = \\
 &= x_1 \sum_{i=1}^8 c_i + x_2 \sum_{i=2}^9 c_i + \sum_{i=3}^{10} c_i - \sum_{i=1}^{10} b_i c_i.
 \end{aligned}$$

Необходимо найти $\min f(\bar{x})$ при ограничениях:

$$\begin{cases} x_1 \geq b_1; \\ x_1 + x_2 \geq b_2; \\ x_1 + x_2 + x_3 \geq \max_{3 \leq i \leq 8} b_i; \\ x_2 + x_3 \geq b_9; \\ x_3 \geq b_{10}. \end{cases}$$

Пример 1.5. Решение задач симплекс-методом.

Задача 1.3. Найти $\max f(\bar{x}) = 21x_1 + 11x_2$ при ограничениях:

$$\begin{cases} 7x_1 + 4x_2 \leq 13; \\ x_1 \leq 1; \\ 2x_1 + x_2 \leq 3; \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Решение. 1. Приведем систему ограничений к каноническому виду путем добавления ослабляющих переменных x_3, x_4 :

$$\begin{cases} 7x_1 + 4x_2 + x_3 = 13; \\ x_1 + x_4 = 1; \\ 2x_1 + x_2 + x_5 = 3. \end{cases}$$

2. Разрешим систему ограничений относительно базисных переменных. Ранг системы $r = 3$. В качестве базисных переменных примем переменные x_3, x_4, x_5 . Тогда переменные x_1, x_2 будут свободными.

$$\begin{cases} x_3 = 13 - (7x_1 + 4x_2); \\ x_4 = 1 - x_1; \\ x_5 = 3 - (2x_1 + x_2). \end{cases}$$

Целевая функция: $f(\bar{x}) = 0 - (-21x_1 - 11x_2)$.

3. Составим табл. 1.4.

Таблица 1.4

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_3	13	7	4	1	0	0
x_4	1	1	0	0	1	0
x_5	3	2	1	0	0	1
$f(\bar{x})$	0	-21	-11	0	0	0

$$\min \left\{ \frac{13}{7}, \frac{1}{1}, \frac{3}{2} \right\} = \frac{1}{1}, \quad \alpha_{ij} = 1 - \text{разрешающий элемент.}$$

4. Составим табл. 1.5. Для чего выведем элемент x_4 из базиса, а элемент x_1 введём в базис.

Таблица 1.5

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_3	6	0	4	1	-7	0
x_1	1	1	0	0	1	0
x_5	3	0	1	0	-2	1
$f(\bar{x})$	21	0	-11	0	21	0

$$\min \left\{ \frac{6}{4}, \frac{1}{1} \right\} = \frac{1}{1}.$$

5. Составим табл. 1.6. Для этого удалим из базиса элемент x_5 и введем в базис x_2 .

Таблица 1.6

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_3	2	0	0	1	1	-4
x_1	1	1	0	0	1	0
x_2	1	0	1	0	-2	1
$f(\bar{x})$	32	0	0	0	-1	11

6. Составим табл. 1.7, которая образуется при удалении из базиса элемента x_1 и ввода в базис элемента x_4 .

Таблица 1.7

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_3	1	-1	0	1	0	-4
x_4	1	1	0	0	1	0
x_2	3	2	1	0	0	1
$f(\bar{x})$	33	1	0	0	0	11

Поскольку в последней строке табл. 1.7 отрицательных коэффициентов нет, то процесс решения задач закончен. При этом $x_1 = 0$, $x_2 = 3$, $x_3 = 1$, $x_4 = 1$, $x_5 = 0$ (x_1, x_5 – свободные переменные).

Оптимальное значение целевой функции – $f(\bar{x}) = 33$.

Геометрическая интерпретация решения задачи представлена на рис. 1.5.

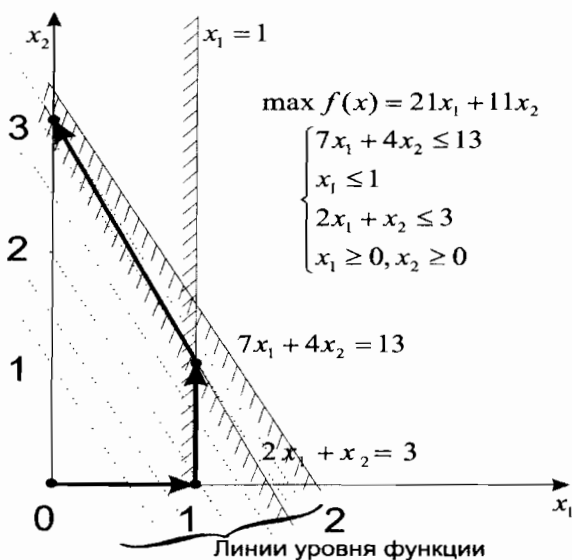


Рис. 1.5. Геометрическое представление задачи

Задача 1.4. Найти \max функции $f(\bar{x}) = 3 + 4x_1 + 6x_2$ при ограничениях:

$$\begin{cases} 2x_1 - 3x_2 \leq 0; \\ 2x_1 - 2x_2 \geq -4; \\ 4x_1 - 5x_2 \geq -20; \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Геометрическая интерпретация задачи 1.4 приведена на рис. 1.6.

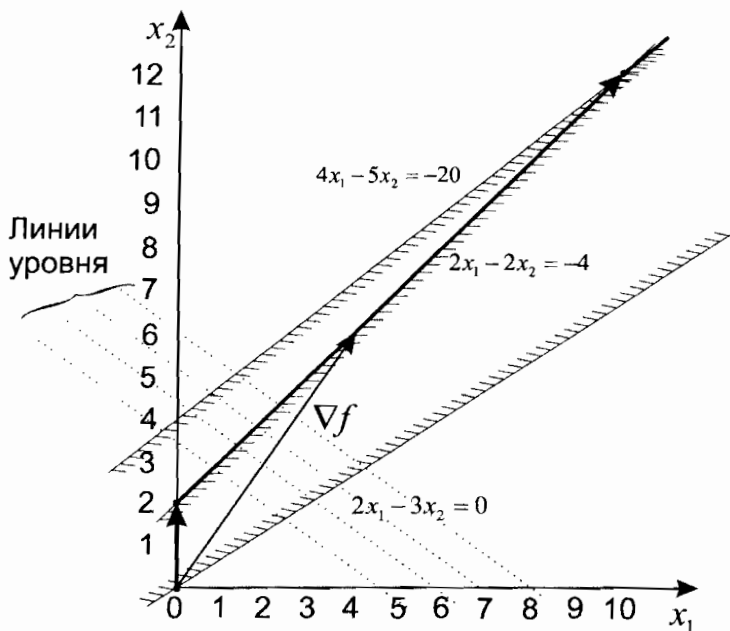


Рис. 1.6. Геометрическое представление задачи 2

Решение будем проводить в соответствии с порядком, изложенным при решении предыдущей задачи.

$$1. \quad \begin{cases} 2x_1 - 3x_2 + x_3 = 0, \\ 2x_1 - 2x_2 - x_4 = -4, \quad r = 3. \\ 4x_1 - 5x_2 - x_5 = -2; \end{cases}$$

2. x_3, x_4, x_5 – базисные; x_1, x_2 – свободные;

$$\begin{cases} x_3 = 0 - 2x_1 + 3x_2; \\ x_4 = 4 + 2x_1 - 2x_2; \\ x_5 = 20 + 4x_1 - 5x_2; \end{cases} \quad f(\bar{x}) = 3 + 4x_1 + 6x_2 - \text{целевая функция.}$$

Составим соответствующие табл. 1.8 – 1.10.

Таблица 1.8

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_3	0	2	-3	1	0	0
x_4	4	-2	2	0	1	0
x_5	20	-4	5	0	0	1
$f(\bar{x})$	3	-4	-6	0	0	0

Таблица 1.9

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_3	6	-1	0	1	3/2	0
x_2	2	-1	1	0	1/2	0
x_5	10	1	0	0	-5/2	1
$f(\bar{x})$	15	-10	0	0	3	0

Таблица 1.10

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_3	16	0	0	1	-1	1
x_2	12	0	1	0	-2	1
x_1	10	1	0	0	5/2	1
$f(\bar{x})$	115	0	0	0	-28	10

Как видно, в табл. 1.10 все $\alpha_{ij} < 0$ ($j=4$). Следовательно, задача не имеет решения.

Задача 1.5. Найти \max функции $f(\bar{x}) = 2 + 4x_1 + 2x_2 + 2x_3$ при ограничениях:

$$\begin{cases} 2x_1 - 3x_2 + 15x_3 \leq 3; \\ 2x_1 + 2x_2 + 8x_3 \leq 32; \\ 2x_1 - 4x_2 + 16x_3 \leq 2; \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{cases}$$

Решение будем проводить в соответствии с порядком, изложенным при решении задач 1.3 и 1.4.

$$1. \quad \begin{cases} 2x_1 - 3x_2 + 15x_3 + x_4 = 3; \\ 2x_1 + 2x_2 + 8x_3 + x_5 = 32; \\ 2x_1 - 4x_2 + 16x_3 + x_6 = 2, \end{cases} \quad r = 3.$$

$$2. \quad \begin{cases} x_4 = 3 - (2x_1 - 3x_2 + 15x_3); \\ x_5 = 32 - (2x_1 + 2x_2 + 8x_3); \\ x_6 = 2 - (2x_1 - 4x_2 + 16x_3), \end{cases} \quad f(\bar{x}) = 2 - (-4x_1 - 2x_2 - 2x_3).$$

Составим соответствующие табл. 1.11 – 1.14.

Таблица 1.11

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5	x_6
x_4	3	2	-3	15	1	0	0
x_5	32	2	2	8	0	1	0
x_6	2	2	-4	16	0	0	1
$f(\bar{x})$	2	-4	-2	-2	0	0	0

Таблица 1.12

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5	x_6
x_4	1	0	1	-1	1	0	-1
x_5	30	0	6	-8	0	1	-1
x_1	1	1	-2	8	0	0	1/2
$f(\bar{x})$	6	0	-10	30	0	0	2

Таблица 1.13

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5	x_6
x_2	1	0	1	-1	1	0	-1
x_5	24	0	0	-2	-6	1	-5
x_1	3	1	0	6	2	0	-3/2
$f(\bar{x})$	16	0	0	20	10	0	-8

Таблица 1.14

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5	x_6
x_2	19/5	0	1	-7/5	-1/5	1/5	0
x_6	24/5	0	0	-2/5	-6/5	1/5	1
x_1	51/5	1	0	32/5	1/5	3/10	0
$f(\bar{x})$	272/5	0	0	84/5	2/5	8/5	0

В соответствии с алгоритмом, приведенным выше, все $\gamma_j > 0$, значит, достигнуто оптимальное решение.

Таким образом, имеем:

$$x_1 = \frac{51}{5}, x_2 = \frac{19}{5}, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = \frac{24}{5}$$

(x_3, x_4, x_5 – свободные переменные).

Оптимальное значение целевой функции – $f(\bar{x}) = 54\frac{2}{5}$.

1.3. Вырожденные задачи линейного программирования

1.3.1. Понятия вырожденности и заклинивания решения задач линейного программирования

Определение 1.1. Если хотя бы одно базисное решение \bar{x}_{b_i} – вырожденное, то задача линейного программирования называется *вырожденной*, если все базисные решения \bar{x}_{b_i} – невырожденные, то задачу называют *невырожденной*.

Определение 1.2. Базисное решение $\bar{x}_{b_i} = (x_1, \dots, x_r, \dots, x_n)$ называется невырожденным, если оно имеет точно r ($r = m$ – ранг системы ограничений) положительных координат. Если число положительных координат базисного решения меньше r , то решение называется вырожденным.

Пример 1.6. Рассмотрим задачу 1.4 из примера 1.2, которая не имеет решения.

Найти $\max f(\bar{x}) = 3 + 4x_1 + 6x_2$.

Система ограничений имеет вид

$$\begin{aligned} x_3 &= 0 - (2x_1 - 3x_2); \\ x_4 &= 4 - (2x_1 - 3x_2); \\ x_5 &= 20 - (2x_1 - 3x_2); \quad r = 3. \end{aligned}$$

Первое базисное решение – вырожденное.

$$\left. \begin{aligned} x_3 &= 0; \\ x_4 &= 4; \\ x_5 &= 20 \end{aligned} \right\} \text{ -- базисное решение;}$$

$$\left. \begin{aligned} x_1 &= 0; \\ x_2 &= 0; \\ x_3 &= 0 \end{aligned} \right\} \text{ -- свободные переменные.}$$

Следовательно, по определению 1.1, задача – вырожденная (при решении этой задачи симплекс-методом, мы не столкнулись ни с какими особенностями; то что у задачи нет решения с вырожденностью никак не связано).

Рассмотрим ситуации, когда задача может обладать вырожденными базисными решениями.

1. Если ранг матрицы A меньше m ($r < m$), то все базисные решения задачи – вырожденные. Это не страшно, так как вырожденные задачи линейного программирования решаются не сложнее невырожденных. Такую ситуацию легко исключить, отбросив «лишние» уравнения.

2. Если ранг матрицы A равен m ($r = m$), то задача также может обладать вырожденными базисными решениями (см. пример 1.6),

причем это может обнаружиться не на первой итерации симплекс-метода, а позже.

К чему может привести применение симплекс-метода, если он попал в вырожденную точку?

Возможна (но необязательна) ситуация, когда симплекс-таблица принимает вид табл. 1.15.

Таблица 1.15

Базис	Свободный член	x_1	...	x_j	...	x_n	Примечание
x_1	$\beta_1 > 0$	α_{11}	...	α_{1j}	...	α_{1n}	Коэффициент α_{ij} будет разрешающим элементом, так как отношение $(\beta_i / \alpha_{ij} = 0)$ минимально
.	
.	
.	
x_i	$\beta_i = 0$	$\alpha_{ij} > 0$...	α_{in}	
.	
.	
x_r	$\beta_r > 0$	α_{rj}	...	α_{rn}	
$f(\bar{x})$	γ_0	γ_j	...	γ_j	...	γ_n	$\gamma_j < 0$ -- коэффициент, который выберем

В результате получим следующее текущее базисное решение:

$$x_1 = \beta_1, \dots, x_i = 0, \dots, x_r = \beta_r, x_{r+1} = x_j = \dots = x_n = 0.$$

После изменения таблицы в соответствии с симплекс-методом базисное решение имеет вид

$$x_1 = \beta_1, \dots, x_j = 0, \dots, x_r = \beta_r, x_{r+1} = 0, \dots, x_i = 0, \dots, x_n = 0.$$

Таким образом, новое базисное решение совпадает с предыдущим (в столбце «Свободный член» ничего не изменилось). Резуль-

татом итерации явилось то, что изменился базис точки. При этом значение целевой функции ($f(\bar{x}) = \gamma_0$), естественно, не увеличилось. (Коэффициенты α_{ij} и γ_j изменились.)

Вычисления можно продолжать, надеясь, что в ходе очередной итерации удастся увеличить значение целевой функции, при этом возможно повторение нескольких «холостых» итераций.

Поскольку число различных базисов любого базисного решения конечно, то после некоторого числа «холостых» итераций либо:

1) выяснится, что базисное решение, на котором мы «застряли» оптимально;

2) обнаружится, что целевая функция задачи не ограничена сверху на допустимом множестве;

3) получится новое базисное решение;

4) придем к базису текущей точки, который уже фигурировал на одной из предыдущих итераций.

В отличие от благополучных случаев 1, 2 и 3, случай 4 означает, что вернемся к состоянию (с той же самой симплекс-таблицей), в котором уже были, и, следовательно, проходя снова и снова через тот же самый ряд пустых итераций, будем возвращаться в это же положение. Такое явление называют *зацикливанием* процесса решения задачи.

Из опыта применения симплекс-метода для исследования моделей реальных явлений сложилось убеждение, что вероятность зацикливания ничтожно мала. Известно лишь несколько специально разработанных задач линейного программирования, в процессе решения которых возникает зацикливание. Поэтому, поскольку зацикливание хотя и маловероятно, но все же возможно, необходимо уметь его устранять.

Оказывается, зацикливание можно предупредить, если внести в симплекс-алгоритм определенные уточнения относительно правила выбора разрешающего элемента. Любое правило выбора разрешающего элемента, с помощью которого можно избежать зацикливания, называют *антициклином*. Имеется несколько, причем достаточно простых, антициклинов. Остановимся на одном из них.

1.3.2. Антициклон

Легко доказывается тот факт, что в невырожденной задаче минимум отношения $\frac{\beta_i}{\alpha_{ij}}$ достигается на единственном номере l .

Следовательно, номер переменной, которая выводится из числа базисных переменных — x_l , и разрешающий элемент — α_{lj} в невырожденных задачах определяется однозначно. В вырожденных задачах это не так, т.е. может быть несколько минимальных отношений $\frac{\beta_i}{\alpha_{ij}}$. Предположим, что на очередной итерации симплекс-

таблица имеет вид табл. 1.16.

Таблица 1.16

Базис	Свободный член	x_1	...	x_r	x_{r+1}	...	x_j	...	x_n
x_1	$\beta_1 > 0$	1	...	α_{1r}	α_{1r+1}	...	α_{1j}	...	α_{1n}
.
.
.
x_{l_1}	$\beta_{l_1} > 0$	0	...	0	$\alpha_{l_1 r+1}$...	$\alpha_{l_1 j} > 0$...	$\alpha_{l_1 n}$
.
.
.
x_{l_2}	$\beta_{l_2} > 0$	0	...	0	$\alpha_{l_2 r+1}$...	$\alpha_{l_2 j} < 0$...	$\alpha_{l_2 n}$
.
.
.
x_r	$\beta_r > 0$	0	...	1	α_{rr+1}	...	α_{rj}	...	α_{rn}
$f(\bar{x})$	γ_0	0	...	0	γ_{r+1}	...	$\gamma_j < 0$...	γ_n

Предположим, что

$$\frac{\beta_{l_1}}{\alpha_{l_1 j}} = \frac{\beta_{l_2}}{\alpha_{l_2 j}} = \min_{\alpha_{ij} > 0} \frac{\beta_i}{\alpha_{ij}}.$$

Следовательно, нет однозначного выбора разрешающего элемента. Предположим, что мы произвольно выбрали $l = l_1$. Тогда после преобразования таблицы на месте свободного члена β_{l_2} появится нуль:

$$\beta_{l_2} - \frac{\beta_{l_1}}{\alpha_{l_1 j}} \alpha_{l_2 j} = \left(\frac{\beta_{l_2}}{\alpha_{l_2 j}} - \frac{\beta_{l_1}}{\alpha_{l_1 j}} \right) \alpha_{l_2 j} = 0.$$

Таким образом, появится вырожденное базисное решение и станет возможным заикливание. (Теперь ясно, почему в невырожденной задаче может быть только одно минимальное отношение.)

Доказано [1], что заикливание не может возникнуть, если на каждой итерации номер l -й переменной, выводимой из базиса, в случае неоднозначности, выбирать по следующему правилу.

Пусть имеем:

$$Q^0 = \min_{\alpha_{ij} > 0} \frac{\beta_i}{\alpha_{ij}} = \frac{\beta_{l_1}}{\alpha_{l_1 j}} = \dots = \frac{\beta_{l_k}}{\alpha_{l_k j}}$$

(т.е. минимум достигается на номерах $k > 1$).

Надо вычислить отношения $\frac{\alpha_{i1}}{\alpha_{ij}}$ для $i = l_1, \dots, l_k$ (т.е. вместо β_i

из столбца свободных членов брать α_{i1} из первого столбца, который соответствует x_1) и найти среди них $\min Q^1 = \min_{i=l_1 \dots l_k} \frac{\alpha_{i1}}{\alpha_{ij}}$.

Если минимум Q^1 достигается только для одного номера из l_1, \dots, l_k (например, l_2), то вопрос решен — x_{l_2} выводится из базиса, $\alpha_{l_2 j}$ — разрешающий элемент.

Если минимум Q^1 достигается более чем для одного номера, то для всех этих номеров $l_1 \dots l_s$ ($s \leq k$) находится $Q^2 = \min_{i=l_1 \dots l_s} \frac{\alpha_{i2}}{\alpha_{ij}}$ (коэффициенты из второго столбца) и т.д.

Гарантируется, что найдется такое q ($1 \leq q \leq n$), для которого минимальное значение Q^q достигается только для одного номера l .

Следовательно, приведенное правило всегда однозначно определяет переменную, выводимую из базиса.

Необходимо заметить, что хотя среди прикладных задач линейного программирования вырожденные задачи встречаются довольно часто, тем не менее заикливание бывает крайне редко. Использование антициклина на каждом шаге симплекс-метода приводит к заметному увеличению времени, которое требуется для решения задачи на ЭВМ. Поэтому на практике чаще всего пользуются *упрощенным правилом* выбора разрешающего элемента в случае неоднозначности, т.е. если минимум достигается на нескольких отношениях $\frac{\beta_i}{\alpha_{ij}}$, то берут в качестве l *наименьший* из номеров i .

В том случае, если обнаружится заикливание, принимают описанный или какой-нибудь другой антициклон, а после того, как заикливание преодолено, снова возвращаются к упрощенному правилу выбора разрешающего элемента.

1.4. Метод искусственного базиса

1.4.1. Особенности метода

Для решения задачи линейного программирования симплекс-методом, как было отмечено в разд. 1.2, необходимо выразить базисные переменные через свободные, т.е. получить выражение (1.3). *Причем в этом выражении свободные члены должны быть неотрицательны* ($\beta_j \geq 0, j = \overline{1, r}$). Во многих задачах эта проблема решается тривиально, но не во всех. В случае, если проблема не решается тривиально, используют метод искусственного базиса. Метод позволяет либо выразить базисные переменные через свободные, либо установить, что система ограничений несовместна. (Процедура Жордана – Гаусса [2] гарантирует $\beta_j \geq 0$ и отличается от симплекс-метода, а метод искусственного базиса сводится к тому же симплекс-методу.)

Запишем основную задачу линейного программирования:
найти $\max f(\bar{x}) = c_1 x_1 + \dots + c_n x_n + c_0$ при ограничениях

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{r1}x_1 + \dots + a_{rn}x_n = b_r. \end{cases} \quad (1.5)$$

Предположим, что все коэффициенты $b_i \geq 0$, $i = \overline{1, r}$. Если какое-то $b_i < 0$, то умножением i -го уравнения на -1 можно обеспечить неотрицательность b_i .

Введем вспомогательные переменные $y_i \geq 0$, $i = \overline{1, r}$:

$$\begin{cases} y_1 = b_1 - a_{11}x_1 - \dots - a_{1n}x_n; \\ y_2 = b_2 - a_{21}x_1 - \dots - a_{2n}x_n; \\ \dots \\ y_r = b_r - a_{r1}x_1 - \dots - a_{rn}x_n. \end{cases} \quad (1.6)$$

Решение системы линейных уравнений (1.5) эквивалентно решению системы (1.6) при дополнительных условиях

$$y_1 = y_2 = \dots = y_r = 0.$$

Введем также дополнительную целевую функцию:

$$\Phi = y_1 + y_2 + \dots + y_r.$$

Итак, поставлена вспомогательная задача линейного программирования, которая принимает следующий вид:

найти $\min \Phi$ при ограничениях (1.6) и условиях

$$x_1 \geq 0, \dots, x_n \geq 0, \quad y_1 \geq 0, \dots, y_r \geq 0.$$

Для её решения можно воспользоваться симплекс-методом. При этом начальный базис уже найден. Действительно, если за базисные переменные принять y_1, \dots, y_r , а за свободные — x_1, \dots, x_n , то в системе (1.6) базисные переменные уже выражены через свободные, причем $b_i \geq 0$, $i = \overline{1, r}$.

Предположим, что вспомогательная задача решена и найден минимум целевой функции Φ . Возможны два варианта:

1) $\min \Phi = 0$. Очевидно, что это возможно, лишь когда все вспомогательные переменные равны нулю, а именно, когда $y_1^* = \dots = y_r^* = 0$ (* – оптимальное решение).

Если выписать из последней симплекс-таблицы оптимальные значения переменных x_1^*, \dots, x_n^* , то очевидно, что они являются решениями системы уравнений (1.5);

2) $\min \Phi > 0$. Это означает, что система (1.6) не имеет решений, для которых $y_1^* = \dots = y_r^* = 0$.

Отсюда следует, что система (1.5) несовместна.

Рассмотрим более подробно первый случай, когда $\min \Phi = 0$. Предположим, что в последней симплекс-таблице вспомогательной задачи все вспомогательные переменные y_i находятся в группе свободных переменных (табл. 1.17).

Таблица 1.17

Базис	Свободные члены	x_1	x_2	x_3	x_4	y_1	y_2	y_3
x_1	2	1	0	0	1	-2	3	1
x_2	1	0	1	0	2	3	-2	4
x_3	3	0	0	1	-3	1	3	2
$-\Phi$	0	0	0	0	5	7	1	4

Из табл. 1.17 следует, что базисные переменные x_1, x_2, x_3 можно выразить через свободные – x_4, y_1, y_2, y_3 следующим образом:

$$\begin{cases} x_1 = 2 - (x_4 - 2y_1 + 3y_2 + y_3); \\ x_2 = 1 - (2x_4 + 3y_1 - 2y_2 + 4y_3); \\ x_3 = 3 - (-3x_4 + y_1 + y_3 + y_2). \end{cases}$$

В общем виде это можно записать так:

$$\begin{cases} x_j = \beta_j - (\alpha_{jr+1}x_{r+1} + \dots + \alpha_{jn}x_n + \alpha_{j1}y_1 + \dots + \alpha_{jr}y_r); \\ j = \overline{1, r}. \end{cases} \quad (1.7)$$

Обратим внимание на то, что все $\beta_j \geq 0$, $j = \overline{1, r}$. Это справедливо, так как в столбце свободных членов находятся оптимальные значения базисных переменных, а они всегда неотрицательны.

Полагая теперь в выражении (1.7) $y_1 = \dots = y_r = 0$, получаем искомый начальный базис:

$$\begin{cases} x_j = \beta_j - (\alpha_{jr+1}x_{r+1} + \dots + \alpha_{jn}x_n); \\ \beta_j \geq 0. \end{cases}$$

Далее необходимо выразить целевую функцию через свободные переменные x_{r+1}, \dots, x_n и решать основную задачу максимизации функции $f(\bar{x})$ с системой ограничений (1.5). Начальная симплекс-таблица для этой задачи будет лишь немного отличаться от последней симплекс-таблицы для вспомогательной задачи, т.е. будут отсутствовать столбцы, соответствующие переменным y_j , и будет другой строка целевой функции, а основная часть таблицы будет та же.

Однако может сложиться такая ситуация, что в табл. 1.17 вспомогательной задачи не все переменные y_j окажутся в числе свободных, некоторые из них могут оказаться в числе базисных (табл. 1.18).

Таблица 1.18

Базис	Свободные члены	x_1	x_2	x_3	x_4	y_1	y_2	y_3
x_1	2	1	0	1	0	0	1	2
x_2	1	0	1	-1	2	0	0	1
y_1	0	0	0	1	0	1	-1	0
-Ф	0	0	0	2	1	0	0	3

Здесь базисные переменные — x_1, x_2, y_1 , а x_3, x_4, y_2, y_3 — свободные, причем свободный член, соответствующий y_1 , должен быть обязательно равен нулю, так как в оптимальном решении $y_1^* = 0$ (т.е. имеем дело с вырожденной задачей). В этом случае

необходимо произвести дополнительные действия и перевести вспомогательные переменные y_j из базисных в число свободных.

Возможны два случая.

1. Если ранг матрицы A в выражении (1.5) равен m , т.е. в исходной системе ограничений нет «лишних» (линейно-зависимых) уравнений, то *всегда* можно вывести *все* переменные y_j из числа базисных. Для этого в строке, соответствующей y_j , надо найти положительный коэффициент при свободной переменной x_k и, условно приняв его за разрешающий элемент, осуществить переход к следующей симплекс-таблице обычным способом. Решение вспомогательной задачи при этом не изменится, так как свободный член в строке y_j равен нулю и, следовательно, элементы столбца свободных членов при преобразовании таблицы не изменятся. В последнем примере «1» в заштрихованном прямоугольнике таблицы является разрешающим элементом, поэтому y_1 выводится из базиса в число свободных переменных, а x_3 вводится в базис.

2. Если ранг матрицы A ($r < m$), то оказывается, что $(m - r)$ переменных y_j не удастся вывести из числа базисных способом, описанным выше. Это означает, что ограничения задачи, номера которых соответствуют индексам оставшихся векторов y_j , можно отбросить – они «лишние».

Таким образом, решение вспомогательной задачи не только позволяет найти начальный базис исходной задачи, но и дает возможность выявить линейно независимые ограничения данной задачи.

1.4.2. Примеры решения задач с использованием метода искусственного базиса

Пример 1.7. Найти $\max f(\bar{x}) = 2x_1 + x_2 - 3x_3 + x_4$ при ограничениях:

$$\begin{cases} -2x_1 + 4x_2 + 3x_3 = -4; \\ -x_1 + 2x_2 - 2x_4 = 2 \end{cases}$$

и условиях: $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0$.

Решение. Сформулируем вспомогательную задачу. Для этого введём две вспомогательные переменные y_1 и y_2 .

Необходимо найти $\min \Phi(\bar{y}) = y_1 + y_2$ при ограничениях:

$$\begin{cases} y_1 = 4 - (2x_1 - 4x_2 - 3x_3); \\ y_2 = 2 - (-x_1 + 2x_2 - 2x_4) \end{cases}$$

и условиях неотрицательности переменных $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, y_1 \geq 0, y_2 \geq 0$.

Выразим функцию $\Phi(\bar{y})$ через свободные переменные x_1, x_2, x_3, x_4 (y_1, y_2 – базисные):

$$\Phi = 6 - x_1 + 2x_2 + 3x_3 + 2x_4.$$

Будем искать $\max(-\Phi)$ симплекс-методом, при этом целевую функцию приведем к виду

$$-\Phi = -6 - (-x_1 + 2x_2 + 3x_3 + 2x_4).$$

Составим симплекс-таблицу (табл. 1.19).

Таблица 1.19

Базис	Свободные члены	x_1	x_2	x_3	x_4	y_1	y_2
y_2	4	2	-4	3	0	1	0
y_1	2	-1	2	0	-2	0	1
$-\Phi$	-6	-4	2	3	2	0	0

В соответствии с симплекс-методом получим табл. 1.20.

Таблица 1.20

Базис	Свободные члены	x_1	x_2	x_3	x_4	y_1	y_2
x_1	2	1	-2	-3/2	0	1/2	0
y_2	4	0	0	-3/2	2	1/2	1
$-\Phi$	-4	0	0	3/2	2	1/2	0

Из последней таблицы следует, что решением вспомогательной задачи являются:

$$\Phi = 4, \quad y_1 = 0, \quad y_2 = 4.$$

Отсюда вывод – система ограничений основной задачи несовместна.

Пример 1.8. Этот пример дает возможность рассмотреть особенности использования метода искусственного базиса в вырожденных задачах линейного программирования.

Итак, необходимо найти $\max f(\bar{x}) = x_1 + 3x_2 + 2x_3 + 4x_4 + x_5$ при ограничениях:

$$\begin{cases} -x_1 + 4x_3 + 3x_4 = 2; \\ 2x_1 + 3x_2 + 3x_3 + 5x_4 - x_5 = 3; \\ x_1 + 3x_2 + x_3 + 2x_4 + x_5 = 2; \\ 2x_1 + 6x_2 + 8x_3 + 10x_4 = 7; \\ x_j \geq 0, \quad j = \overline{1,5}. \end{cases}$$

Как видно, в данной системе ограничений последнее уравнение является суммой трех предыдущих.

Вспомогательная задача имеет вид:

найти $\min \Phi(\bar{y}) = y_1 + y_2 + y_3 + y_4$ при ограничениях:

$$\begin{cases} y_1 = 2 - (-x_1 + 4x_3 + 3x_4); \\ y_2 = 3 - (2x_1 + 3x_2 + 3x_3 + 5x_4 - x_5); \\ y_3 = 2 - (x_1 + 3x_2 + x_3 + 2x_4 + x_5); \\ y_4 = 7 - (2x_1 + 6x_2 + 8x_3 + 10x_4). \end{cases}$$

При этом целевая функция описывается выражением:

$$\Phi = 14 - (4x_1 + 12x_2 + 16x_3 + 20x_4),$$

или

$$-\Phi = -14 - (-4x_1 - 12x_2 - 16x_3 - 20x_4).$$

Составим симплекс-таблицу (табл. 1.21).

Таблица 1.21

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4
y_1	2	-1	0	4	3	0	1	0	0	0
y_2	3	2	3	3	5	-1	0	1	0	0
y_3	2	1	3	1	2	1	0	0	1	0
y_4	7	2	6	8	10	0	0	0	0	1
-Ф	-14	-4	-12	-16	-20	0	0	0	0	0

Для нахождения разрешающего элемента используем другое правило выбора столбца, а именно, берем первый столбец, у которого $\gamma_j < 0$, а не максимально по модулю, и преобразуем табл. 1.21 в соответствии с симплекс-методом, на основе которого вводим в базис переменную x_1 вместо y_2 .

Таблица 1.22

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4
y_1	7/2	0	3/2	11/2	11/2	-1/2	1	1/2	0	0
x_1	3/2	1	3/2	3/2	5/2	-1/2	0	1/2	0	0
y_3	1/2	0	3/2	-1/2	-1/2	3/2	0	-1/2	1	0
y_4	4	0	3	5	5	1	0	-1	0	1
-Ф	-8	0	-6	-10	-10	-2	0	2	0	0

Исходя из указанного выше правила, в табл. 1.22 выбираем разрешающий элемент в столбце x_2 .

На основании симплекс-метода вводим в базис переменную x_2 вместо y_3 . При этом получим табл. 1.23.

При выборе разрешающего элемента в табл. 1.23 (столбец x_3) имеем минимальные отношения

$$\min\left\{\frac{3}{6}, \frac{1}{2}, \frac{3}{6}\right\} = \frac{1}{2}$$

в трех строках, соответствующих элементам y_1, x_1, y_4 . Следовательно, задача – вырожденная, однозначного выбора разрешающего элемента нет.

Таблица 1.23

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4
y_1	3	0	0	6	6	-2	1	1	-1	0
x_1	1	1	0	2	3	-2	0	1	-1	0
x_2	1/3	0	1	-1/3	-1/3	1	0	-1/3	2/3	0
y_4	3	0	0	6	6	-2	0	0	-2	1
-Φ	-6	0	0	-12	-12	-4	0	0	4	0

Применяем антициклин. Для этого составим таблицу, элементами которой являются отношения элементов табл. 1.23, указанных выше строк к коэффициентам $\alpha_{13}, \alpha_{23}, \alpha_{43}$ соответственно.

Таблица 1.23а

α_{y_1}/α_{13}	1/2	0	0	1	1	-1/3	1/6	1/6	-1/6	0
α_{x_1}/α_{23}	1/2	1/2	0	1	3/2	-1	0	1/2	-1/2	0
α_{y_4}/α_{43}	1/2	0	0	1	1	-1/3	0	0	-1/3	1/6

Анализируя каждый из столбцов табл. 1.23а в соответствии с подходом, изложенным в разд. 1.3.2, получим, что из базиса необходимо вывести переменную y_4 .

В результате получим табл. 1.24.

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_3	y_4
y_1	0	0	0	0	0	0	1	1	1	-1
x_1	0	1	0	0	1	-4/3	0	1	-1/3	-1/3
x_2	1/2	0	1	0	0	8/9	0	-1/3	5/9	1/18
x_3	1/2	0	0	1	1	-1/3	0	0	-1/3	1/6
-Φ	0	0	0	0	0	0	0	0	0	2

Табл. 1.24 – оптимальная, и $\Phi=0$. Следовательно, задача имеет непустое допустимое множество.

Удалить y_1 из базисных переменных невозможно, так как коэффициенты этой строки при свободных переменных x_4, x_5 равны нулю. Это значит, что ограничение первое – лишнее, его можно отбросить и ранг матрицы ограничений станет равным числу строк ($r = 3$).

При этом начальный базис для основной задачи запишется в виде:

$$\begin{cases} x_1 = 0 - \left(x_4 - \frac{4}{3}x_5 \right); \\ x_2 = \frac{1}{2} - \left(\frac{8}{9}x_5 \right); \\ x_3 = \frac{1}{2} - \left(x_4 - \frac{1}{3}x_5 \right). \end{cases}$$

Выразим целевую функцию $f(\bar{x})$ через свободные переменные x_4, x_5 :

$$\begin{aligned} f(\bar{x}) &= x_1 + 3x_2 + 2x_3 + 4x_4 + x_5 = \left(-x_4 + \frac{4}{3}x_5 \right) + \\ &+ \left(\frac{3}{2} - \frac{8}{3}x_5 \right) + \left(1 - 2x_4 + \frac{2}{3}x_5 \right) + 4x_4 + x_5 = \frac{5}{2} - \left(-x_4 - \frac{1}{3}x_5 \right). \end{aligned}$$

Тогда начальная симплекс-таблица для основной задачи принимает вид табл. 1.25.

Таблица 1.25

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5
x_1	0	1	0	0	1	-4/3
x_2	1/2	0	1	0	0	8/9
x_3	1/2	0	0	1	1	-1/3
$f(\bar{x})$	5/2	0	0	0	-1	-1/3

В первой строке табл. 1.25 есть вырожденность. Используя симплекс-метод, вводим в базис переменную x_4 , при этом переменную x_1 исключаем из базиса. В результате табл. 1.25 приводится к табл. 1.25а.

Таблица 1.25а

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5
x_4	0	1	0	0	1	-4/3
x_2	1/2	0	1	0	0	8/9
x_3	1/2	-1	0	1	0	1
$f(\bar{x})$	5/2	1	0	0	0	5/3

Поскольку решение задачи на данном этапе не получено, проведем дальнейшие преобразования последней таблицы. Выбираем в качестве разрешающего элемента коэффициент, расположенный в третьей строке таблицы, в соответствии с указанным выше правилом. Получим табл. 1.26, которая требует дальнейших преобразований.

В соответствии с симплекс-методом, выбирая разрешающий элемент в столбце x_1 , табл. 1.26 приводим к табл. 1.26а.

Таблица 1.26

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5
x_4	2/3	-1/3	0	4/3	1	0
x_2	1/18	8/9	1	-8/9	0	0
x_5	1/2	-1	0	1	0	1
$f(\bar{x})$	10/3	-2/3	0	5/3	0	0

Таблица 1.26а

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5
x_4	11/16	0	3/8	1	1	0
x_1	1/16	1	9/8	-1	0	0
x_5	9/16	0	9/8	-1	1	1
$f(\bar{x})$	27/8	0	3/4	1	0	0

Таким образом, как следует из последней таблицы, оптимальным решением исходной задачи будут:

$$x_1 = 1/16, \quad x_2 = 0, \quad x_3 = 0, \quad x_4 = 11/16, \quad x_5 = 9/16, \quad f(\bar{x}) = 27/8.$$

Пример 1.9. Найти $\max f(\bar{x}) = 2x_1 + x_3 - x_4 + x_5$ при ограничениях:

$$\begin{cases} -x_1 + 2x_2 + x_3 = 2; \\ 3x_1 + 5x_2 + x_3 + x_4 + 2x_5 = 14; \\ x_1 - x_2 + x_5 = 1. \end{cases}$$

Запишем вспомогательную задачу:

найти $\min \{\Phi(\bar{y}) = y_1 + y_2 + y_3\}$ при ограничениях:

$$\begin{cases} y_1 = 2 - (-x_1 + 2x_2 + x_3); \\ y_2 = 14 - (3x_1 + 5x_2 + x_3 + x_4 + 2x_5); \\ y_3 = 1 - (x_1 - x_2 + x_5). \end{cases}$$

Подставляя значения \bar{y} в формулу для целевой функции вспомогательной задачи, получим:

$$\Phi(\bar{x}) = 17 - (3x_1 + 6x_2 + 2x_3 + x_4 + 3x_5).$$

Будем искать $\max \{-\Phi(x)\}$.

Исходная симплекс-таблица имеет вид табл. 1.27.

Таблица 1.27

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_5
y_1	2	-1	2	1	0	0	1	0	0
y_2	14	3	5	1	1	2	0	1	0
y_3	1	1	-1	0	0	1	0	0	1
$-\Phi$	-17	-3	-6	-2	-1	-3	0	0	0

После ряда последовательных преобразований, в соответствии с симплекс-методом, получим последовательность симплекс-таблиц, отражающих процесс нахождения решения исходной задачи (табл. 1.27а – 1.27в).

Таблица 1.27а

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_5
1	-1/2	1	1/2	0	0	1/2	0	0	
9	11/2	0	-3/2	1	2	-5/2	1	0	
2	1/2	0	1/2	0	1	1/2	0	1	
-11	-6	0	1	-1	-3	3	0	0	

Если в табл. 1.27б взять элемент 7/11 в качестве разрешающего, то процесс сойдется быстрее и окончательная симплекс-таблица примет вид табл. 1.27в.

Таблица 1.27б

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_5
x_2	20/11	0	1	4/11	1/11	2/11	3/11	1/11	0
x_1	18/11	1	0	-3/11	2/11	4/11	-5/11	2/11	0
y_3	13/11	0	0	7/11	-1/11	9/11	8/11	-1/11	1
$-\Phi$	-13/11	0	0	-7/11	1/11	-9/11	3/11	12/11	0

Таблица 1.27в

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5	y_1	y_2	y_5
x_2	8/7	0	1	0	1/7	-2/7	-1/7	1/7	-4/7
x_1	15/7	1	0	0	1/7	5/7	1/7	1/7	3/7
x_3	13/7	0	0	1	-1/7	9/7	8/7	-1/7	11/7
$-\Phi$	0	0	0	0	0	0	1	1	1

В результате найдено оптимальное решение вспомогательной задачи.

Поскольку $\min \Phi = 0$ и $y_1 = y_2 = y_3 = 0$, то начальный базис для основной задачи запишется следующим образом:

$$\begin{cases} x_2 = \frac{8}{7} - \left(\frac{1}{7}x_4 - \frac{2}{7}x_5 \right); \\ x_1 = \frac{15}{7} - \left(\frac{1}{7}x_4 + \frac{5}{7}x_5 \right); \\ x_3 = \frac{13}{7} - \left(-\frac{1}{7}x_4 + \frac{9}{7}x_5 \right). \end{cases}$$

Выразим целевую функцию через x_4 и x_5 :

$$\begin{aligned} f(\bar{x}) &= 2x_1 + x_3 - x_4 + x_5 = \\ &= \frac{30}{7} - \frac{2}{7}x_4 - \frac{10}{7}x_5 + \frac{13}{7} + \frac{1}{7}x_4 - \frac{9}{7}x_5 - x_4 + x_5 = \frac{43}{7} - \frac{8}{7}x_4 - \frac{12}{7}x_5. \end{aligned}$$

С учетом полученного результата составим исходную симплекс-таблицу для основной задачи (табл. 1.28).

Таблица 1.28

Базис	Свободные члены	x_1	x_2	x_3	x_4	x_5
x_2	8/7	0	1	0	1/7	-2/7
x_1	15/7	1	0	0	1/7	5/7
x_3	13/7	0	0	1	-1/7	9/7
$f(\bar{x})$	43/7	0	0	0	8/7	12/7

В нижней строке табл. 1.28, соответствующей целевой функции, нет отрицательных коэффициентов, следовательно, получено оптимальное решение поставленной вначале задачи, а именно:

$$x_1 = \frac{15}{7}, \quad x_2 = \frac{8}{7}, \quad x_3 = \frac{13}{7}, \quad x_4 = 0, \quad x_5 = 0, \quad f(\bar{x}) = \frac{43}{7}.$$

1.5. Двойственные задачи линейного программирования

1.5.1. Основные положения

Понятие двойственности в задачах линейного программирования расширяет границы применения симплексных алгоритмов.

Пусть дана задача линейного программирования, которую назовем исходной или первой задачей:

Задача 1.6 (исходная). Найти

$$\max f(\bar{x}) = \sum_{j=1}^n c_j x_j + c_0$$

при ограничениях

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

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

Сформулируем другую задачу, которую назовем двойственной по отношению к исходной.

Задача 1.7 (двойственная). Найти

$$\min \varphi(\bar{y}) = \sum_{i=1}^m b_i y_i + c_0$$

при ограничениях

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

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

В матричном виде эти задачи можно записать так.

Задача 1.8. Найти $\max f(\bar{x}) = \langle \bar{c}, \bar{x} \rangle + c_0$ при ограничениях

$$A\bar{x} \leq \bar{b}; \quad \bar{x} \geq 0.$$

Задача 1.9. Найти $\min \varphi(\bar{y}) = \langle \bar{b}, \bar{y} \rangle + c_0$ при ограничениях

$$A^T \bar{y} \geq \bar{c};$$

$$\bar{y} \geq 0.$$

Итак, при переходе от исходной задачи к соответствующей двойственной задаче производят следующие преобразования:

- 1) заменяют максимизацию целевой функции минимизацией;
- 2) смысл неравенств-ограничений меняется на противоположный, если в исходной задаче неравенство меньше или равно нулю, то в двойственной задаче – больше или равно нулю;
- 3) число переменных x в исходной задаче равно числу ограничений в двойственной, и наоборот;
- 4) матрица A системы ограничений исходной задачи транспонируется;
- 5) векторы \bar{b} и \bar{c} меняются местами, т.е. коэффициенты целевой функции исходной задачи (\bar{c}) становятся столбцом свободных членов системы ограничений двойственной задачи, а столбец свободных членов системы ограничений исходной задачи (\bar{b}) становится коэффициентами целевой функции двойственной задачи.

Пример 1.10. Исходная задача:

найти $\max f(\bar{x}) = 12 + 8x_1 + 36x_2 + 4x_3 + 12x_4$ при ограничениях

$$\begin{cases} 88x_1 - 20x_2 + 32x_3 + 14x_4 \leq 12; \\ -16x_1 + 4x_2 + 48x_3 + 4x_4 \leq 10; \\ -24x_1 - 10x_2 + 6x_3 + 8x_4 \leq 44; \\ x_i \geq 0, \quad i = \overline{1,4}. \end{cases}$$

Двойственная задача:

найти $\min \varphi(\bar{y}) = 12 + 12y_1 + 10y_2 + 44y_3$ при ограничениях

$$\begin{cases} 88y_1 - 16y_2 - 24y_3 \geq 8; \\ -20y_1 + 4y_2 - 10y_3 \geq 36; \\ 32y_1 + 48y_2 + 6y_3 \geq 4; \\ 14y_1 + 4y_2 + 8y_3 \geq 12; \\ y_j \geq 0, \quad j = \overline{1,3}. \end{cases}$$

Легко показать, что если за исходную принять задачу 1.7 и применить к ней аналогичные преобразования, то получим задачу 1.6. Поэтому рассмотренные задачи называются взаимно двойственными.

Взаимозависимость исходной и двойственной задач определяется рядом теорем, которые рассмотрим без доказательств.

Лемма. Если \bar{x}^0 — произвольное допустимое решение задачи 1.6, а \bar{y}^0 — произвольное допустимое решение задачи 1.7, то справедливо соотношение $f(\bar{x}^0) \leq \varphi(\bar{y}^0)$. Если для \bar{x}^0 и \bar{y}^0 значения целевых функций совпадают, т.е. $f(\bar{x}^0) = \varphi(\bar{y}^0)$, то \bar{x}^0 — оптимальное решение задачи 1.6, а \bar{y}^0 — оптимальное решение задачи 1.7.

Доказательство. 1. Так как \bar{x}^0 является допустимым решением задачи 1.6, то имеет место система неравенств

$$\sum_{j=1}^n a_{ij}x_j^0 \leq b_i, \quad i = \overline{1, m}.$$

Умножим i -е неравенство на i -ю компоненту вектора \bar{y}^0 , т.е. на y_i^0 (так как $y_i^0 \geq 0$, то знак неравенства не изменится) и просуммируем по всем i .

В результате получим:

$$\sum_{i=1}^m \sum_{j=1}^n a_{ij} x_j^0 y_i^0 \leq \sum_{i=1}^m b_i y_i^0.$$

Аналогично, для вектора \bar{y}^0 справедливо неравенство

$$\sum_{i=1}^m a_{ij} y_i^0 \geq c_j, \quad j = \overline{1, n}.$$

Умножим j -е неравенство на $x_j^0 \geq 0$ и просуммируем по всем j :

$$\sum_{j=1}^n \sum_{i=1}^m a_{ij} y_i^0 x_j^0 \geq \sum_{j=1}^n c_j x_j^0.$$

Объединяя полученные неравенства, получим

$$\sum_{j=1}^n c_j x_j^0 \leq \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_j^0 y_i^0 \leq \sum_{i=1}^m b_i y_i^0.$$

Отсюда

$$\sum_{j=1}^n c_j x_j^0 + c_0 \leq \sum_{i=1}^m b_i y_i^0 + c_0,$$

т.е. $f(\bar{x}^0) \leq \varphi(\bar{y}^0)$.

2. Покажем, что если $f(\bar{x}^0) = \varphi(\bar{y}^0)$, то \bar{x}^0 и \bar{y}^0 – оптимальные решения.

А. Предположим противное, т.е. что существует такая точка \bar{x}' , что $f(\bar{x}') > f(\bar{x}^0)$, но в этом случае получим $f(\bar{x}') > \varphi(\bar{y}^0)$, а это противоречит доказанному в п. 1.

Б. Аналогично, предположим, что найдется такая точка \bar{y}' , что $\varphi(\bar{y}') < \varphi(\bar{y}^0)$, но в этом случае получим $\varphi(\bar{y}') < f(\bar{x}^0)$, а это противоречит доказанному условию.

Первая теорема двойственности. Если одна из задач (1.6 или 1.7) имеет оптимальное решение, то и другая также имеет оптимальное решение, причем их оптимумы совпадают, т.е. $\max f(\bar{x}) = \min \varphi(\bar{y})$.

Если целевая функция одной из задач не ограничена, то другая задача не имеет допустимых решений.

Вторая теорема двойственности (теорема о дополняющей нежесткости). Для того чтобы допустимые векторы \bar{x}^* и \bar{y}^* были оптимальными, необходимо выполнение следующих условий:

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

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

Эту теорему надо понимать так, что если оптимальное решение исходной задачи подставить в систему ограничений и если i -я строка обратится в строгое неравенство, то i -я компонента оптимального решения двойственной задачи равна нулю.

Или так, если i -я компонента оптимального решения двойственной задачи положительна, то i -е ограничение исходной задачи выполняется как строгое равенство.

Исходя из рассмотренных свойств взаимно двойственных задач можно сформулировать *правило решения двойственной задачи*. Оказывается, если решена исходная задача, то двойственную задачу решать не нужно. Оптимальным решением двойственной задачи являются коэффициенты при ослабляющих переменных в строке целевой функции в последней симплекс-таблице исходной задачи.

Отметим, что коэффициенты при остальных x_j в строке целевой функции последней симплекс-таблицы равны разности между левой и правой частями j -го ограничения двойственной задачи, т.е.

величине $\sum_{i=1}^m a_{ij} y_i^* - c_j$.

1.5.2. Двойственный симплекс-алгоритм

В связи с постановкой двойственных задач и приведением их к стандартному виду возникает проблема решения линейных задач, когда оптимальный базис является недопустимым.

Например, если привести двойственную задачу последнего примера к стандартному виду, получим:

найти $\max \{-\varphi(\bar{y}) = -12 - (12y_1 + 10y_2 + 44y_3)\}$ при ограничениях:

$$\begin{cases} -88y_1 + 16y_2 + 24y_3 + y_4 = -8; \\ 20y_1 - 4y_2 + 10y_3 + y_5 = -36; \\ -32y_1 - 48y_2 - 6y_3 + y_6 = -4; \\ -14y_1 - 4y_2 - 8y_3 + y_7 = -12. \end{cases}$$

Начальный базис для данной задачи имеет вид:

$$\begin{aligned} y_4 &= -8 - (-88y_1 + 16y_2 + 24y_3); \\ y_5 &= -36 - (20y_1 - 4y_2 + 10y_3); \\ y_6 &= -4 - (-32y_1 - 48y_2 - 6y_3); \\ y_7 &= -12 - (-14y_1 - 4y_2 - 8y_3). \end{aligned}$$

Этот базис будет недопустимым, так как $\beta_i < 0$, т.е. свободные члены в системе ограничений отрицательны.

Подобные задачи могут возникать не только при решении двойственных задач, но и в самом общем случае.

Однако существует метод, позволяющий решать задачи с недопустимым базисом. Этот метод называется двойственным симплекс-алгоритмом.

Алгоритм двойственного симплекс-метода можно получить, проведя рассуждения, аналогичные выводу простого симплекс-метода. Все пункты рассматриваемого алгоритма такие же, как у простого симплекс-метода, но имеется ряд отличий (предполагается, что решается задача максимизации):

1) возможность применения обычного симплекс-метода определялась положительностью коэффициентов столбца свободных членов β_i в системе ограничений, разрешенной относительно базиса, признаком допустимости в двойственном симплекс-алгоритме является неотрицательность коэффициентов целевой функции γ_j (не считая свободного члена);

2) признаком оптимальности полученного решения является неотрицательность всех β_i коэффициентов столбца свободных членов (не считая значения целевой функции);

3) переменная x_i , выводимая из базиса, определяется максимальным по модулю отрицательным коэффициентом в столбце свободных членов;

4) разрешающий элемент выбирается в i -й строке среди отрицательных коэффициентов $\alpha_{ij} < 0$ и соответствует максимальному отношению коэффициентов строки целевой функции к элементам i -й строки, т.е. $\frac{\gamma_j}{\alpha_{ij}}$ (табл. 1.29).

Таблица 1.29

Базис	Свободные члены	x_1	...	x_i	...	x_r	x_{r+1}	...	x_j	...	x_n
x_1	β_1	1	...	0	...	0	α_{1r+1}	...	α_{1j}	...	α_{1n}
.
.
.
x_i	β_i	0	...	1	...	0	α_{ir+1}	...	α_{ij}	...	α_{in}
.
.
.
x_r	β_r	0	...	0	...	1	α_{rr+1}	...	α_{rj}	...	α_{rn}
$f(\bar{x})$	γ_0	0	...	0	...	0	γ_{r+1}	...	γ_j	...	γ_n

1.5.3. Примеры решения двойственных задач линейного программирования

Пример 1.11. Исходная задача: найти $\max \{f(\bar{x}) = 21x_1 + 11x_2\}$ при ограничениях:

$$\begin{cases} 7x_1 + 4x_2 \leq 13; \\ x_1 \leq 1; \\ 2x_1 + x_2 \leq 3; \\ x_1, x_2 \geq 0. \end{cases}$$

Двойственная задача: найти $\min \{\varphi(\bar{y}) = 13y_1 + y_2 + 3y_3\}$ при ограничениях:

$$\begin{cases} 7y_1 + y_2 + 2y_3 \geq 21; \\ 4y_1 + y_3 \geq 11; \\ y_1, y_2, y_3 \geq 0. \end{cases}$$

Решение двойственной задачи имеет вид:

$$y_1 = 0, \quad y_2 = 0, \quad y_3 = 11; \\ \varphi_{\min}(\bar{y}) = 33.$$

Полученное решение находится из оптимальной таблицы исходной задачи.

Пример 1.12. Исходная задача: найти $\max \{f(\bar{x}) = 3 + 4x_1 + 6x_2\}$ при ограничениях:

$$\begin{cases} 2x_1 - 3x_2 \leq 0; \\ -2x_1 + 2x_2 \leq 4; \\ -4x_1 + 5x_2 \leq 20; \\ x_1, x_2 \geq 0. \end{cases}$$

Двойственная задача: найти $\min \{\varphi(\bar{y}) = 3 + 4y_2 + 20y_3\}$ при ограничениях:

$$\begin{cases} 2y_1 - 2y_2 - 4y_3 \geq 4; \\ -3y_1 + 2y_2 + 5y_3 \geq 6; \\ y_1, y_2, y_3 \geq 0. \end{cases}$$

Так как в исходной задаче целевая функция неограниченно возрастает, двойственная задача также не имеет решения (причем ее допустимое множество пусто).

Пример 1.13. Исходная задача: найти

$$\max \{f(\bar{x}) = 2 + 4x_1 + 2x_2 + 2x_3\}$$

при ограничениях

$$\begin{cases} 2x_1 - 3x_2 + 15x_3 \leq 3; \\ 2x_1 + 2x_2 + 8x_3 \leq 32; \\ 2x_1 - 4x_2 + 16x_3 \leq 2; \\ x_1, x_2, x_3 \geq 0. \end{cases}$$

Двойственная задача: найти

$$\min \{\varphi(\bar{y}) = 2 + 3y_1 + 32y_2 + 2y_3\}$$

при ограничениях:

$$\begin{cases} 2y_1 + 2y_2 + 2y_3 \geq 4; \\ 3y_1 + 2y_2 - 4y_3 \geq 2; \\ 15y_1 + 8y_2 + 16y_3 \geq 2; \\ y_1, y_2, y_3 \geq 0. \end{cases}$$

Решение двойственной задачи имеет вид:

$$y_1 = \frac{2}{5}, \quad y_2 = \frac{8}{5}, \quad y_3 = 0;$$

$$\varphi(\bar{y}) = 54\frac{2}{5}.$$

В рассматриваемом примере решением являются коэффициенты при ослабляющих переменных x_4, x_5, x_6 в строке целевой функции оптимальной таблицы исходной задачи.

Пример 1.14. Решить задачу на основе двойственного симплекс-алгоритма.

Исходная задача: найти

$$\max \{f(\bar{x}) = 12 + 8x_1 + 36x_2 + 4x_3 + 12x_4\}$$

при ограничениях:

$$\begin{cases} 88x_1 - 20x_2 + 32x_3 + 14x_4 \leq 12; \\ -16x_1 + 4x_2 + 48x_3 + 4x_4 \leq 10; \\ -24x_1 - 10x_2 + 6x_3 + 8x_4 \leq 44; \\ x_1, x_2, x_3, x_4 \geq 0. \end{cases}$$

Двойственная задача: найти

$$\min \{ \varphi(\bar{y}) = 12 + 12y_1 + 10y_2 + 44y_3 \}$$

при ограничениях:

$$\begin{cases} 88y_1 - 16y_2 - 24y_3 \geq 8; \\ -20y_1 + 4y_2 - 10y_3 \geq 36; \\ 32y_1 + 48y_2 + 6y_3 \geq 4; \\ 14y_1 + 4y_2 + 8y_3 \geq 12; \\ y_1, y_2, y_3 \geq 0. \end{cases}$$

Добавим ослабляющие переменные y_4, y_5, y_6, y_7 . В результате получим начальный базис:

$$\begin{cases} y_4 = -8 - (-88y_1 + 16y_2 + 24y_3); \\ y_5 = -36 - (20y_1 - 4y_2 + 10y_3); \\ y_6 = -4 - (-32y_1 - 48y_2 + 6y_3); \\ y_7 = -12 - (-14y_1 - 4y_2 - 8y_3). \end{cases}$$

Это недопустимый базис, так как свободные члены в уравнениях начального базиса — отрицательные. Следовательно, необходимо использовать двойственный симплекс-алгоритм.

Будем искать $\max \{ (-\varphi(\bar{y})) = -12 - (12y_1 + 10y_2 + 44y_3) \}$.

Соответствующая последовательность симплекс-таблиц имеет вид:

Базис	Свободные члены	y_1	y_2	y_3	y_4	y_5	y_6	y_7
y_4	-8	-88	16	24	1	0	0	0
y_5	-36	20	-4	10	0	1	0	0
y_6	-4	-32	-48	-6	0	0	1	0
y_7	-12	-14	-4	-8	0	0	0	1
$-\Phi$	-12	12	10	44	0	0	0	0

Базис	Свободные члены	y_1	y_2	y_3	y_4	y_5	y_6	y_7
y_4	-152	-8	0	64	1	4	0	0
y_2	9	-5	1	-5/2	0	-1/4	0	0
y_6	428	-272	0	-126	0	-12	1	0
y_7	24	-34	0	-18	0	-1	0	1
$-\Phi$	-102	62	0	69	0	2.5	0	0

Базис	Свободные члены	y_1	y_2	y_3	y_4	y_5	y_6	y_7
y_1	19	1	0	-8	-1/8	-1/2	0	0
y_2	104	0	1	-85/2	-5/8	-11/4	0	0
y_6	5596	0	0	-2302	-34	-148	1	0
y_7	670	0	0	-290	-17/4	-18	0	1
$-\Phi$	-1280	0	0	565	7.75	67/2	0	0

Как следует из последней симплекс-таблицы, оптимальным решением будут:

$$y_1 = 19, y_2 = 104, y_3 = 0;$$

$$\Phi_{\min}(\bar{y}) = 1280.$$

Пример 1.15. Решить задачу с использованием двойственного симплекс-алгоритма.

Пусть после приведения некоторой задачи к стандартному виду вместе с ограничениями и замены требования минимизации

целевой функции на максимизацию мы пришли к следующей задаче:

найти $\max \{-\varphi(\bar{y}) = -8y_1 - 12y_2 - 18y_3\}$ при ограничениях:

$$\begin{cases} y_1 + 3y_2 \geq 2; \\ y_1 + 3y_2 \geq 1; \\ y_1, y_2, y_3 \geq 0. \end{cases}$$

Запишем начальный базис:

$$\begin{cases} y_4 = -2 - (y_1 + 3y_2); \\ y_5 = -1 - (y_1 + 3y_2). \end{cases}$$

Отрицательные свободные члены в начальном базисе преобразованной задачи указывают на необходимость применения двойственного симплекс-алгоритм, при этом

$$-\varphi(\bar{y}) = 0 - (8y_1 + 12y_2 + 18y_3).$$

Соответствующая симплекс-таблица имеет вид:

Базис	Свободные члены	y_1	y_2	y_3	y_4	y_5
y_1	2	-1	0	3	1	0
y_5	-1	-1	-3	0	0	1
$-\varphi$	0	8	12	18	0	0

Как следует из таблицы,

$$\max \left\{ \frac{\gamma_k}{\alpha_{ik}} \right\} = \max \left\{ -\frac{8}{1}, -\frac{18}{3} \right\} = \max \{-8, -6\} = -6.$$

Соответственно, преобразованная таблица имеет вид:

Базис	Свободные члены	y_1	y_2	y_3	y_4	y_5
y_3	2/3	1/3	0	1	-1/3	0
y_5	-1	-1	-3	0	0	1
$-\varphi$	-12	2	12	0	6	0

Аналогично, из полученной таблицы следует:

$$\max\{-2, -4\} = -2.$$

Тогда, преобразованная таблица представится в виде:

Базис	Свободные члены	y_1	y_2	y_3	y_4	y_5
y_3	1/3	0	-1	1	-1/3	1/3
y_1	1	1	3	0	0	-1
$-\varphi$	-14	0	6	0	6	2

В результате из последней таблицы следует, что оптимальным решением задачи будут:

$$y_1 = 1, y_2 = 0, y_3 = 1/3;$$

$$\varphi_{\min} = 14.$$

1.6. Транспортная задача линейного программирования

Раннее было рассмотрено симплекс-метод, который является методом решения задач линейного программирования общего вида. Симплекс-метод является конечным, а не итерационным, тем не менее, число его итераций может быть весьма большим. Многие классы широко распространенных на практике задач линейного программирования обладают особенностями, выражающимися в специфическом строении матрицы A коэффициентов системы ограничений, которые позволяют существенно упростить общий метод решения применительно к рассматриваемой в данном разделе задаче. При этом процесс получения оптимального решения ускоряется. В других случаях удастся построить принципиально новый метод решения, приспособленный к определенному классу задач.

Одним из примеров задач со специальной структурой матрицы ограничений является транспортная задача.

Рассмотрим постановку простейшей транспортной задачи и один из наиболее популярных методов ее решения, принципиально отличающийся от симплекс-метода.

1.6.1. Постановка задачи

Пусть на базах A_1, A_2, \dots, A_m сосредоточены определенные ресурсы в количестве a_1, a_2, \dots, a_m . Эти ресурсы должны быть доставлены на объекты B_1, B_2, \dots, B_n в количестве b_1, b_2, \dots, b_n . Стоимость перевозки единицы груза из A_i в B_j обозначим через c_{ij} . Через x_{ij} обозначим количество ресурсов, которое доставляется из A_i в B_j .

Задача 1.10. Найти матрицу перевозок $\|x_{ij}\|$, $i = \overline{1, m}$, $j = \overline{1, n}$, которая минимизирует целевую функцию $f(\bar{x}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$, удовлетворяющую ограничениям

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

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

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

Первое и второе ограничения означают, что суммарное количество груза, вывозимого с базы A_i , точно равно имеющимся там запасам a_i , а ввозимый груз на базу B_j точно равен заявкам этой базы.

Последнее условие называется *балансным* и означает равенство всего количества имеющихся запасов и потребностей.

Поставленная транспортная задача – это задача линейного программирования, но записанная особым образом. Для её решения разработаны специальные методы более эффективные, чем симплекс-метод.

В представленной задаче $m \times n$ неизвестных.

Введем ряд понятий.

Матрица $\|x_{ij}\|$ размерности $m \times n$ называется *планом перевозок*,

x_{ij} – *перевозкой*, $\|c_{ij}\|$ – *матрицей издержек*.

План называется *допустимым*, если он удовлетворяет указанным выше ограничениям.

План называется *оптимальным*, если он минимизирует функцию $f(\bar{x})$.

Переменные x_{ij} удовлетворяют $m+n$ уравнениям системы ограничений, ранг которой r равен $m+n-1$. Если расположить переменные x_{ij} следующим образом:

$$x_{11}, x_{12}, \dots, x_{1n}, x_{21}, \dots, x_{2n}, \dots, x_{m1}, \dots, x_{mn},$$

то матрица ограничений транспортной задачи примет вид

$$A = \left(\begin{array}{c|c|c|c} \overbrace{1 \ 1 \ \dots \ 1}^n & \overbrace{0 \ 0 \ \dots \ 0}^n & \dots & \overbrace{0 \ 0 \ \dots \ 0}^n \\ \overbrace{0 \ 0 \ \dots \ 0}^n & \overbrace{1 \ 1 \ \dots \ 1}^n & \dots & \overbrace{0 \ 0 \ \dots \ 0}^n \\ \dots & \dots & \dots & \dots \\ \overbrace{0 \ 0 \ \dots \ 0}^n & \overbrace{0 \ 0 \ \dots \ 0}^n & \dots & \overbrace{1 \ 1 \ \dots \ 1}^n \\ \overbrace{1 \ 0 \ \dots \ 0}^n & \overbrace{1 \ 0 \ \dots \ 0}^n & \dots & \overbrace{1 \ 0 \ \dots \ 0}^n \\ \overbrace{0 \ 1 \ \dots \ 0}^n & \overbrace{0 \ 1 \ \dots \ 0}^n & \dots & \overbrace{0 \ 1 \ \dots \ 0}^n \\ \dots & \dots & \dots & \dots \\ \overbrace{0 \ 0 \ \dots \ 1}^n & \overbrace{0 \ 0 \ \dots \ 1}^n & \dots & \overbrace{0 \ 0 \ \dots \ 1}^n \end{array} \right) \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} m \\ n \end{array}.$$

Нетрудно видеть, что ранг r матрицы A равен $m+n-1$. Действительно, всего в матрице $m+n$ строк, и все строки линейно зависимы. Чтобы в этом убедиться, достаточно сложить первые m строк и вычесть сумму последних n , при этом получится нулевой вектор. С другой стороны, нетрудно заметить, что любые $m+n-1$ строк матрицы A линейно независимы.

Итак, ранг системы ограничений равен $m+n-1$. Следовательно, можно выделить $m+n-1$ базисных переменных.

Существует понятие «*опорный план*» – это план, в котором отличны от нуля не более чем $m + n - 1$ переменных x_{ij} .

Опорный план – это фактически базис.

1.6.2. Нахождение первого опорного плана

Как и в задаче линейного программирования, в транспортной задаче необходимо сначала найти первый допустимый базис. Для этого существует несколько методов, на некоторых из них остановимся ниже.

Зададим транспортную задачу табл. 1.30.

Таблица 1.30

A	B				
	B_1	B_2	...	B_n	Запасы a_i
A_1	c_{11} x_{11}	c_{12} x_{12}	...	c_{1n} x_{1n}	a_1
A_2	c_{21} x_{21}	c_{22} x_{22}	...	c_{2n} x_{2n}	a_2
...
A_m	c_{m1} x_{m1}	c_{m2} x_{m2}	...	c_{mn} x_{mn}	a_m
Заявки b_j	b_1	b_2	...	b_n	$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$

В табл. 1.30 A – пункт отправления, B – пункт назначения.

Диагональный метод нахождения опорного плана состоит в следующем. В верхнюю левую клетку таблицы ставится минимальное из чисел a_1 и b_1 , т.е. $\min\{a_1, b_1\}$.

Пусть $a_1 < b_1$. Следовательно, поставщик A_1 полностью использовал свои запасы, и при установлении остальных перевозок его можно не учитывать (строка, соответствующая a_1 , из таблицы вычеркивается). Теперь потребность 1-го потребителя будет составлять $b_1 - a_1$.

Если наоборот, $a_1 > b_1$, то соответствующий столбец из таблицы вычеркивается.

Далее процесс повторяется.

Пример 1.15. Рассмотрим табл. 1.31. В этом примере $m = 3, n = 4$.

В результате получили $m + n - 1 = 6$ отличных от нуля переменных, т.е. это опорный план, в соответствии с которым

$$f(\bar{x}) = \sum_{i=1}^3 \sum_{j=1}^4 c_{ij} x_{ij} = 1630.$$

Таблица 1.31

A	B				
	B_1	B_2	B_3	B_4	Запасы a_i
A_1	5 60	7 20	1	3	80 / 20
A_2	4	8 20	4 70	2	90 / 70
A_3	1	5	6 20	9 70	90
Заявки b_j	60	40 / 20	90 / 20	70	260

Можно показать, что диагональный метод всегда приводит к опорному допустимому решению. Однако следует иметь в виду, что поскольку данный метод не учитывает стоимость перевозок, то опорные планы, полученные с его помощью, значительно отличаются от оптимальных.

Метод минимального элемента дает возможность учесть заданные стоимости. От диагонального метода он отличается только тем, что в таблице выбирается не верхняя левая клетка, а клетка с минимальной стоимостью c_{ij} , и как в диагональном методе в эту клетку записывается минимальное из двух чисел a_i и b_j , т.е. $\min\{a_i, b_j\}$. Затем соответствующий столбец или строка из таблицы вычеркивается и т.д.

Пример 1.16. Рассмотрим табл. 1.32.

Таблица 1.32

A	B				
	B ₁	B ₂	B ₃	B ₄	Запасы a _i
A ₁	5	7	1	3	80
A ₂	4	8	4	2	90 / 20 / 10
A ₃	1	5	6	9	90 / 30
Заявки b _j	60	40 / 10	90 / 10	70	260

В соответствии с опорным планом, полученным данным методом, значение функции

$$f(\bar{x}) = \sum_{i=1}^3 \sum_{j=1}^4 c_{ij} x_{ij} = 550.$$

Опыт решения транспортных задач показал, что далеко не всегда использование метода минимального элемента вместо диагонального дает лучшее приближение к оптимальному решению.

1.6.3. Метод потенциалов

Это основной метод решения транспортной задачи. Метод базируется на ряде теорем, которые рассмотрим ниже.

Введем понятие *псевдостоимости*.

Пусть за перевозку единицы груза с базы A_i вносится плата α_i, а за перевозку единицы груза на базу B_j – плата β_j. *Псевдостоимостью* перевозки из A_i в B_j называется сумма $\tilde{c}_{ij} = \alpha_i + \beta_j$.

Совокупность величин (α_i, β_j), i = 1, m, j = 1, n, называется *платежами, или потенциалами*.

Теорема о платежах. Для заданной совокупности платежей суммарная псевдостоимость перевозки при любом допустимом плане x_{ij} сохраняет одно и то же значение, т. е.

$$\tilde{f}(\bar{x}) = \sum_{i=1}^m \sum_{j=1}^n \tilde{c}_{ij} x_{ij} = \text{const}.$$

Доказательство.

$$\begin{aligned} \tilde{f}(\bar{x}) &= \sum_{i=1}^m \sum_{j=1}^n \tilde{c}_{ij} x_{ij} = \sum_{i=1}^m \sum_{j=1}^n (\alpha_i + \beta_j) x_{ij} = \sum_{i=1}^m \sum_{j=1}^n \alpha_i x_{ij} + \sum_{i=1}^m \sum_{j=1}^n \beta_j x_{ij} = \\ &= \sum_{i=1}^m \alpha_i \sum_{j=1}^n x_{ij} + \sum_{j=1}^n \beta_j \sum_{i=1}^m x_{ij} = \sum_{i=1}^m \alpha_i a_i + \sum_{j=1}^n \beta_j b_j = \text{const}. \end{aligned}$$

Так как план – допустимый, то выполняются соотношения:

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

$$\sum_{j=1}^n x_{ij} = b_j,$$

при этом суммарная псевдостоимость перевозки не зависит от плана x_{ij} , т.е. $\tilde{f}(\bar{x}) = \text{const}$. Таким образом, теорема доказана.

Теорема об оптимальности. Если для всех базисных клеток ($x_{ij} > 0$) псевдостоимость равна стоимости, т.е. $\tilde{c}_{ij} = c_{ij}$, а для свободных клеток ($x_{ij} = 0$) выполняется неравенство $\tilde{c}_{ij} \leq c_{ij}$, то план является оптимальным и никаким способом не может быть улучшен.

Доказательство. Для базисных клеток имеем

$$f(\bar{x}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = \sum_{i=1}^m \sum_{j=1}^n \tilde{c}_{ij} x_{ij} = \tilde{f}(\bar{x}) = \text{const}.$$

Заменим план x_{ij} на любой другой допустимый план x'_{ij} . Тогда получим другое значение целевой функции

$$f(\bar{x}') = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x'_{ij}.$$

В базисных клетках, которые сохранились от старого плана, стоимость $c_{ij} = \tilde{c}_{ij}$, а в базисных клетках нового плана, которые в плане x_{ij} были свободными, стоимость $c_{ij} \geq \tilde{c}_{ij}$. Следовательно,

$$f(\bar{x}') \geq \sum_{i=1}^m \sum_{j=1}^n \tilde{c}_{ij} x'_{ij} = \text{const} = \tilde{f}(\bar{x}'),$$

т.е. $f(\bar{x}') \geq \tilde{f}(\bar{x}')$.

Таким образом, при любом плане, отличном от x_{ij} , функция $f(\bar{x}')$ может быть только больше. Значит, x_{ij} — оптимальный план.

Теорема доказана.

Из условия равенства $c_{ij} = \tilde{c}_{ij}$ в базисных клетках можно определить потенциалы α_i и β_j . В опорном плане $(n+m-1)$ базисных клеток; причем значения c_{ij} известны. Для нахождения потенциалов составим систему уравнений, используя табл. 1.32:

$$\alpha_1 + \beta_3 = 1;$$

$$\alpha_2 + \beta_3 = 4;$$

$$\alpha_2 + \beta_2 = 8;$$

$$\alpha_3 + \beta_2 = 5;$$

$$\alpha_3 + \beta_1 = 1;$$

$$\alpha_2 + \beta_4 = 2.$$

Система имеет $(n+m-1)$ уравнений и $(n+m)$ неизвестных потенциалов. Для получения решения необходимо произвольно выбрать один из потенциалов, например $\alpha_1 = 1$. Остальные потенциалы однозначно определяются при решении системы уравнений: $\beta_3 = 0$, $\alpha_2 = 4$, $\beta_2 = 4$, $\alpha_3 = 1$, $\beta_1 = 0$, $\beta_4 = -2$.

Следует отметить, что в системе потенциалов могут быть и отрицательные элементы.

Определив потенциалы, находят псевдостоимости $\tilde{c}_{ij} = \alpha_i + \beta_j$ в остальных небазисных клетках и проставляют их в левом углу небазисных клеток (табл. 1.33).

Таблица 1.33

A	B				
	B ₁	B ₂	B ₃	B ₄	α _i
A ₁	1 < 5	6 < 7	80 1	-1 < 3	1
A ₂	4 ≤ 4	10 8	10 4	70 2	4
A ₃	60 1	30 5	1 < 6	-1 < 9	1
β _j	0	4	0	-2	

Далее проверяют, является ли опорный план оптимальным. В рассматриваемом случае план является оптимальным, так как во всех небазисных клетках $\tilde{c}_{ij} \leq c_{ij}$.

Возьмем другую таблицу, построенную диагональным методом (табл. 1.34).

Таблица 1.34

A	B				
	B ₁	B ₂	B ₃	B ₄	α _i
A ₁	60 5	20 7	3 > 1	6 > 3	1
A ₂	6 > 4	20 8	70 4	7 > 2	2
A ₃	8 > 1	10 > 5	20 6	70 9	4
β _j	4	6	2	5	

Опорный план не является оптимальным, так как имеются клетки, где $\tilde{c}_{ij} > c_{ij}$.

Выбирается клетка с максимальной разностью ($\tilde{c}_{31} - c_{31} = 7$). Эта клетка берется за начальную для построения так называемого цикла пересчета, с помощью которого можно найти новый опорный план с меньшим значением целевой функции.

Циклом пересчета для данной свободной клетки называется замкнутая линия, все вершины которой, за исключением данной, лежат в базисных клетках. Вершинам цикла пересчета присваиваются знаки «+» или «-». Знаки последовательных вершин чередуются, причем свободная клетка имеет положительный знак. Доказано, что для каждой свободной клетки существует и притом единственный цикл пересчета.

Ценой цикла называется приращение стоимости перевозок при перемещении единицы груза по циклу, она равна алгебраической сумме стоимостей, стоящих в вершинах цикла, а, в конечном счете, равна:

$$\gamma_{ij} = c_{ij} - \tilde{c}_{ij}.$$

Цена может быть как положительной, так и отрицательной. Очевидно, $\gamma_{ij} < 0$ при $\tilde{c}_{ij} > c_{ij}$ и в этом случае стоимость перевозок уменьшается.

Некоторые циклы пересчета представлены в табл. 1.35.

Таблица 1.35

A	B					
	B ₁	B ₂	B ₃	B ₄	B ₅	B ₆
A ₁	c ₁₁	- c ₁₂	c ₁₃	c ₁₄	+ + c ₁₅	- c ₁₆
A ₂	c ₂₁	+ c ₂₂	- c ₂₃	c ₂₄	c ₂₅	c ₂₆
A ₃	+ c ₃₁	- c ₃₂	+ c ₃₃	c ₃₄	- c ₃₅	c ₃₆
A ₄	c ₄₁	c ₄₂	- c ₄₃	c ₄₄	c ₄₅	+ c ₄₆
A ₅	- c ₅₁	+ c ₅₂	+ c ₅₃	c ₅₄	- c ₅₅	c ₅₆

Покажем справедливость того, что цена $\gamma_{ij} = c_{ij} - \tilde{c}_{ij}$.

В соответствии с табл. 1.35 цена цикла для клетки (5, 3) определяется выражением:

$$\begin{aligned}\gamma_{53} &= c_{53} - c_{43} + c_{46} - c_{16} + c_{15} - c_{55} = \\ &= c_{53} - (\alpha_4 + \beta_3) + (\alpha_4 + \beta_6) - (\alpha_1 + \beta_6) + (\alpha_1 + \beta_5) - (\alpha_5 + \beta_5) = \\ &= c_{53} - (\beta_3 + \alpha_5) = c_{53} - \tilde{c}_{53},\end{aligned}$$

что соответствует указанному выше утверждению.

Рассмотрим пример, представленный табл. 1.36, элементы которой соответствуют элементам табл. 1.34.

Выберем клетку с максимальной разностью $\tilde{c}_{ij} - c_{ij}$, т.е. клетку (3, 1), и построим для нее цикл пересчета. Прямая, выходящая из клетки, совершает повороты в базисных клетках и возвращается в начальную.

Теперь нужно перераспределить базисные переменные, находящиеся в вершинах цикла, так, чтобы в свободной клетке появилось положительное значение, а одна из базисных переменных приняла нулевое значение. При этом должно сохраниться требование, что все $x_{ij} \geq 0$. Доказано, что при таком перемещении будет получено новое решение транспортной задачи с уменьшенным значением целевой функции, если цена цикла $\gamma_{ij} < 0$.

Таблица 1.36

A	B			
	B ₁	B ₂	B ₃	B ₄
A ₁	5 60 ← -	7 20 ↑ +	1 3	3 6
A ₂	4 6 ↓ +	8 20 ← -	4 70 ↑ +	2 7
A ₃	1 8 ↑ +	5 10 ← -	6 20 ↓ -	9 70

$$f(\bar{x}) = 1630.$$

В примере наименьшее значение базисной переменной, которую можно переместить по циклу, равно 20. Таким образом, это число

нужно прибавить к базисным переменным вершин цикла, имеющих знак «+», и вычесть из переменных в вершинах цикла со знаком «-». В результате, в клетке (3, 1) появится переменная $x_{31} = 20$. Значение переменной более 20 выбрать нельзя, так как это приведет к появлению отрицательных значений x_{ij} .

После перемещения по циклу получим табл. 1.37.

Таблица 1.37

A	B				
	B_1	B_2	B_3	B_4	a_i
A_1	40	40			80
A_2		0 (ϵ)	90		90
A_3	20		0	70	90
b_j	60	40	90	70	

$$f(\bar{x}) = 1490.$$

Новый опорный план соответствует меньшему значению целевой функции, причем уменьшение составляет величину $\gamma_{ij}x_{ij}$ ($7 \cdot 20 = 140$), где x_{ij} – перемещенное по циклу число.

Следует отметить одну особенность полученной таблицы: вместо одной переменной нулю стали равны две переменные x_{22} , x_{33} . Это означает, что задача – вырожденная, так как число базисных переменных меньше $(n + m - 1)$. Для устранения вырожденности предположим, что в одной из клеток, например в клетке (A_2, B_2) , имеется бесконечно малая величина $\epsilon > 0$, которую в окончательном плане примем равной нулю.

Подводя итоги, сформулируем алгоритм решения транспортной задачи.

1. В качестве первого приближения берется допустимый план, полученный одним из методов нахождения начального опорного плана.

2. Для данного плана определяется система платежей исходя из условия, что в любой базисной клетке стоимость равна псевдостоймости.

3. Подсчитывается псевдостоймость для всех свободных клеток.

Если окажется, что для всех клеток плана $\tilde{c}_{ij} \leq c_{ij}$, то задача решена, найденный план оптимален.

В противном случае, т.е. если существуют клетки, в которых $\tilde{c}_{ij} > c_{ij}$, переходим к шагу 4.

4. Для свободной клетки с максимальной разностью $\tilde{c}_{ij} - c_{ij} > 0$ строится цикл пересчета, после чего составляется новый опорный план и осуществляется переход на шаг 2.

Отметим, что логические операции, составляющие содержание процедуры поиска цикла, гораздо более экономны в смысле затрат времени ЭВМ по сравнению с арифметическими операциями симплекс-метода.

В заключение обратим внимание на любопытную *особенность транспортной задачи*: если все числа a_i, b_j ($i = \overline{1, m}, j = \overline{1, n}$) – целые, то любой опорный план транспортной задачи (в том числе оптимальный) – целочисленный.

Действительно, когда все числа a_i, b_j – целые, начальный опорный план, полученный диагональным или другим методом, является целочисленным. Кроме того, на каждой итерации метода потенциалов перемещаемое число x_{ij} является целым, поэтому любой следующий опорный план – целочисленный.

1.6.4. Транспортные задачи с неправильным балансом

Ранее был рассмотрен метод решения *простейшей транспортной задачи* в случае выполнения условия $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$,

т.е. когда суммарный объем возможных поставок равнялся суммарному объему потребностей (*закрытая* транспортная задача). Между тем чаще возникают транспортные задачи, в которых условие баланса нарушено в ту или иную сторону (*открытые* транспортные задачи).

Так в случае, когда $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$, задачу можно сформулировать следующим образом:
найти

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

при ограничениях:

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

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

Это означает, что у некоторых поставщиков остается неотправленная продукция, суммарный объем которой равен

$$\left(\sum_{i=1}^m a_i - \sum_{j=1}^n b_j \right).$$

• Если безразлично, у каких именно поставщиков и в каких количествах останется продукция, то задача с неправильным балансом может быть сведена к эквивалентной ей задаче с правильным балансом, т.е. к закрытой транспортной задаче. Для этого вводится еще один фиктивный потребитель с заявкой

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$$

и стоимостью перевозок от каждого поставщика

$$c_{in+1} = 0, i = \overline{1, m}.$$

Тогда, поставленная задача может быть записана в виде:
найти

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

при ограничениях:

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

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

Полученная задача действительно эквивалентна исходной, так как фиктивные перевозки не влияют на значение целевой функции, поскольку $c_{in+1} = 0$.

Совершенно аналогично можно поступить в случае $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$, т.е. когда продукции недостаточно для полного удовлетворения потребностей. В этом случае вводится в рассмотрение фиктивный поставщик с объемом возможных поставок, равным недостатку продукции $\left(\sum_{j=1}^n b_j - \sum_{i=1}^m a_i \right)$. Стоимости перевозок от фиктивного поставщика ко всем потребителям принимаем также равными нулю, т.е. $c_{m+1j} = 0, \quad j = \overline{1, n}$.

• Если неслучайно у какого поставщика и сколько останется продукции (или какой потребитель и сколько недополучит), то для сведения задачи к закрытой транспортной задаче надо прибегать к более сложным приемам.

Пусть, например, суммарный объем продукции недостаточен; т.е. $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$. И пусть убыток, к которому приводит недополучение продукции j -м потребителем, оценен в r_j единиц тех же, в которых измерены c_{ij} . Тогда, естественно, данную задачу математически можно записать так:

найти

$$\min f(\bar{x}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n r_j y_j. \quad (1.8)$$

В соотношение (1.8)

$$y_j = b_j - \sum_{i=1}^m x_{ij}$$

представляет собой объем неудовлетворенного спроса j -го потребителя при ограничениях:

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = \overline{1, m}; \quad (1.9)$$

$$\sum_{i=1}^m x_{ij} + y_j = b_j, \quad j = \overline{1, n}. \quad (1.10)$$

Эта задача принимает вид обычной закрытой транспортной задачи, если ввести фиктивного $(m+1)$ -го поставщика с объемом поставок

$$a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$$

и принять

$$c_{m+1j} = r_j, \quad \text{а} \quad x_{m+1j} = y_j.$$

Недостающее ограничение $\sum_{j=1}^n x_{m+1j} = a_{m+1}$ получается как сумма по j ограничений (1.10):

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

или

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

Откуда

$$\sum_{j=1}^n x_{m+1j} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i = a_{m+1}.$$

Таким образом, получаем задачу:
найти

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

при ограничениях:

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

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

В заключение следует отметить, что:

метод потенциалов является неединственным методом решения простейшей транспортной задачи;

к аналогичному виду математической модели может привести задача, по своему содержанию никак не связанная с транспортом и с планированием перевозок.

Замечания. Многие практические задачи, связанные с планированием перевозок, не укладываются в рамки рассмотренной задачи, так как осложнены теми или иными дополнительными ограничениями.

1. Например, транспортные магистрали, связывающие источники продукции с потребителями, имеют *ограниченные* (в ограниченный промежуток времени) *пропускные способности*, что может не позволить реализовать оптимальный план перевозок, найденный без их учета. Поэтому ограничения, связанные с пропускными способностями, необходимо включать в задачу.

Так вместо условия $x_{ij} \geq 0$ нужно использовать ограничение $(0 \leq x_{ij} \leq d_{ij})$, где d_{ij} максимальный объем продукции, который может быть перевезен по этой магистрали (из i в j) за рассматриваемый промежуток времени (т.е. пропускная способность d_{ij} может быть равна бесконечности).

Ясно, что задача принципиально отличается от простейшей транспортной задачи, для ее решения имеется метод, являющийся видоизменением метода потенциалов.

2. Задача может быть осложнена наличием *промежуточных транспортных узлов*, в которых производится обработка груза (например, перевалка на другой вид транспорта). Для таких задач также имеются более или менее эффективные методы решения.

3. Принципиально меняется характер модели задачи даже в простейшем случае, если за критерий оптимальности выбрано время, необходимое для обеспечения всех потребителей нужным объемом продукции, и требуется это время минимизировать – *транспортная задача по критерию времени*. Математическая модель этой задачи оказывается нелинейной за счет нелинейной целевой функции. Методы решения такой задачи известны.

4. Часто требуется составить оптимальный план перевозок не одного, а нескольких видов продукции. В постановке задачи появляется объем перевозок x_{ijk} k -го вида продукции от i -го поставщика к j -му потребителю. В остальном формулировка задачи аналогична простейшей. Получается так называемая *трехиндексная транспортная задача*.

Бывают задачи и с большим количеством индексов.

Например, x_{ijkl} – объем перевозок k -го вида продукции l -м видом транспорта от i -го поставщика j -му потребителю, $k = \overline{1, p}$, $l = \overline{1, q}$.

Следовательно, существует класс *многоиндексных транспортных задач*. Можно отметить, что:

простых методов решения таких задач, подобных методу потенциалов, нет и быть не может;

при их решении надо использовать симплекс-метод, как для обычной задачи линейного программирования, но при этом могут возникать проблемы, связанные с большой размерностью задачи при больших значениях m, n, p, q и т.д.

1.7. Дискретное программирование

Дискретное программирование – особый класс задач линейного программирования.

1.7.1. Постановка задачи

Математическая модель общей задачи дискретного программирования может быть представлена следующим образом:

найти

$$\max \left\{ f(\bar{x}) = \sum_{j=1}^n c_j x_j + c_0 \right\}$$

при ограничениях:

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

$$x_j \in A_j = \{0, A_{j1}, A_{j2}, \dots, A_{jk_j}\}.$$

Итак, в этой задаче для каждой переменной x_j заранее определен набор значений, которые она может принимать.

Частным случаем задач дискретного программирования являются задачи *целочисленного программирования*:

найти

$$\max \left\{ f(\bar{x}) = \sum_{j=1}^n c_j x_j + c_0 \right\}$$

при ограничениях:

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

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

$$x_j - \text{целые}, \quad j = \overline{1, n_1}.$$

Если $n_1 < n$, то задача называется *частично целочисленной*, если $n_1 = n$ — *полностью целочисленной*.

Частным случаем задач целочисленного программирования являются задачи с *булевыми переменными*.

Постановка задачи точно такая же, только на x накладывается ограничение

$$x_j = \begin{cases} 0, \\ 1, \end{cases} \quad j = \overline{1, n}.$$

Как и в линейном программировании, вектор \bar{x} , удовлетворяющий всем ограничениям, называется *допустимым*, а допустимый вектор, оптимизирующий целевую функцию, — *оптимальным*.

Рассмотрим ряд классических задач целочисленного программирования.

1. Задача о ранце. Собравшийся в поход путешественник должен упаковать в ранец различные полезные предметы, всего n наименований, причем могут потребоваться несколько одинаковых предметов. Имеется m ограничений, например: на вес, на объем, на линейные размеры и т.д. — т.е. имеется m характеристик, по которым возможности ранца (или путешественника) ограничены.

Пусть:

a_{ij} — i -я характеристика предмета j -го наименования;

b_i — ограничивающая величина для i -й характеристики;

c_j — полезность одного j -го предмета;

x_j — количество предметов j -го наименования, запланированное к загрузке в ранец.

Тогда математическая формулировка задачи о ранце имеет вид:
найти

$$\max \left\{ f(\bar{x}) = \sum_{j=1}^n c_j x_j \right\}$$

при ограничениях:

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

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

К задачам подобного вида приводится большое число практических задач, когда в качестве x_j фигурируют отдельные предметы (машины, кирпичи, люди и т.д.). В первоначальной постановке задачи о ранце рассматривалась загрузка бомбардировщика бомбовым запасом.

2. Задача о назначениях. Имеется n различных самолетов, которые требуется распределить между n авиалиниями. Известно, что на j -й авиалинии i -й самолет будет приносить доход c_{ij} . Требуется так распределить самолеты по линиям, чтобы максимизировать суммарный доход.

Положим

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-й самолет направлен на } j\text{-ю авиалинию;} \\ 0, & \text{в противном случае.} \end{cases}$$

Тогда математическая модель задачи может быть представлена в следующем виде:

найти

$$\max \left\{ f(\bar{x}) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \right\},$$

при ограничениях:

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

$$x_{ij} = \begin{cases} 1, & i = \overline{1, n}, \quad j = \overline{1, n}. \\ 0, & \end{cases}$$

Ограничения этой задачи отражают требования того, что каждый самолет назначается только на одну авиалинию и на каждую авиалинию назначается один самолет.

Очевидно, данная задача является задачей с булевыми переменными.

3. Задача о коммивояжере. Пусть имеются города, занумерованные числами $0, 1, 2, \dots, n$. Выехав из города 0 , коммивояжер должен объехать все остальные города, побывав в каждом из них по одному разу, и вернуться в исходный город. Известны расстояния c_{ij} между городами i и j ($i = 0, \dots, n, j = 0, \dots, n$).

Требуется найти самый короткий маршрут.

Введем следующие переменные:

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер из города } i \text{ переезжает в город } j, \quad i \neq j; \\ 0, & \text{в противном случае.} \end{cases}$$

Таким образом, получаем следующую задачу:

найти

$$\min \left\{ f(\bar{x}) = \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \right\}$$

при ограничениях:

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

(в город j коммивояжер приезжает только один раз);

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

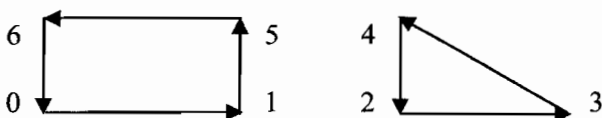
(из города i коммивояжер выезжает только один раз).

Для обеспечения *связности* маршрута введем дополнительные переменные u_i, u_j , которые удовлетворяют следующему неравенству:

$$u_i - u_j + nx_{ij} \leq n-1; \quad i, j = \overline{1, n}; \quad x_{ij} = (0 \cup 1).$$

Вспомогательные переменные u_i, u_j , участвующие только в ограничениях, совершенно необходимы.

Например, маршрут, изображенный на приведенном ниже рисунке, удовлетворяет первым двум ограничениям, но не подходит под условия задачи о коммивояжере.



1.7.2. О решении задач линейного целочисленного программирования

На первых порах требование целочисленности переменных явилось существенным препятствием для исследования и решения таких задач. Самым естественным путем было попытаться использовать обычные методы линейного программирования, несколько видоизменив их.

Казалось бы, что естественный путь решения целочисленной задачи состоит в решении соответствующей линейной задачи с последующим округлением компонент ее оптимального вектора до ближайших целых чисел. На самом деле такой путь в большинстве случаев не только уводит от оптимума, но даже приводит иногда к недопустимому решению задачи.

Пример 1.17. Найти

$$\max \{f(\bar{x}) = x_1 + x_2\}$$

при ограничениях

$$\begin{cases} 2x_1 + 11x_2 \leq 38, \\ 4x_1 - 5x_2 \leq 5, \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0, x_1, x_2 - \text{целые.}$$

Решим задачу геометрически (рис. 1.7).

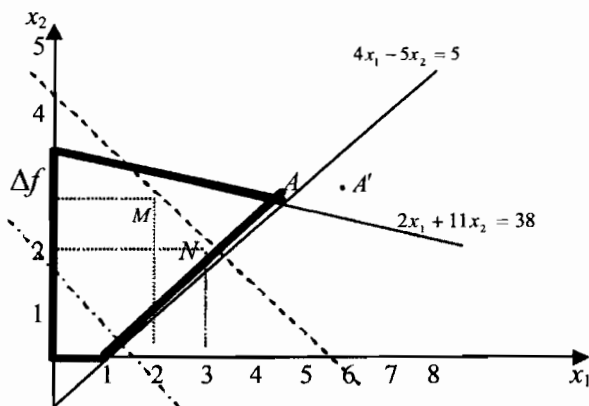


Рис. 1.7. Геометрическое решение задачи

На рис. 1.7 максимум $f(\bar{x}) = 7,17$ достигается в вершине $A(4,54; 2,63)$. Поскольку нас интересуют лишь точки с целочисленными координатами, следовательно, точка A не является допустимым решением задачи. Если округлить значения координат точки A , то получим точку $A'(5; 3)$, которая вообще не принадлежит области поиска. Можно показать, что целочисленный оптимум достигается в точках $N(3; 2)$ и $M(2; 3)$ и равен 5. Обе точки находятся внутри области поиска.

Итак, для задач с требованием целочисленности необходимо иметь особые методы решения, и, кроме того, оказывается, что оптимальное решение задач целочисленного программирования не обязательно принадлежит границе допустимого многогранника, что было характерно для обычных задач линейного программирования.

Тем не менее, некоторые из целочисленных задач могут быть непосредственно сведены к задачам линейного программирования.

Так, обратимся к задаче о назначениях и заменим условие $x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$

условием $x_{ij} \geq 0, i, j = \overline{1, n}$. После такого преобразования задача о назначениях принимает вид транспортной задачи. Поскольку числа

в правых частях ограничений – целые, то по известному свойству транспортной задачи все опорные планы такой задачи целочисленны. Следовательно, ее можно решить методом потенциалов.

Условия же неотрицательности и остальные ограничения $\left(\sum x_{ij} = 1 \right)$ обеспечат, чтобы каждая компонента этого решения была равна 0, или 1.

Однако целочисленную задачу линейного программирования более общего вида, такого, как задача о ранце, уже нельзя свести так просто к обычной задаче линейного программирования. Поэтому теория развития этих задач развивалась в двух направлениях.

Одно из них, предназначенное для решения общей целочисленной задачи, носит название «метод отсечения» и реализует идею сопоставления целочисленной задачи некоторой обычной задаче линейного программирования.

Вторая группа методов носит название «метод ветвей и границ». Здесь также используются иногда свойства соответствующих задач линейного программирования. Кроме того, для специфических задач разрабатываются принципиально новые комбинаторные принципы. Например, для задач о коммивояжере есть особый метод из класса «ветвей и границ», а для задачи о назначениях – очень простой метод, получивший название «венгерский алгоритм». Последний существенно проще процедуры сведения к транспортной задаче.

1.7.3. Метод отсечения. Первый алгоритм Гомори

Сущность этого метода состоит в том, что если после решения линейной задачи (без учета условия целочисленности) симплекс-методом не получен целочисленный ответ, то строят дополнительное линейное ограничение, уменьшающее допустимую область. Дополнительное ограничение, называемое отсечением, отсекает такую часть области, в которой не содержится допустимых решений целочисленной задачи, но есть найденное оптимальное решение нецелочисленной задачи. Таким образом,

отсечение не позволяет потерять оптимальное решение целочисленной задачи. Далее процесс повторяется.

Рассмотрим процесс формирования алгоритма отсечения по Гомори. Постановка задачи в канонической форме имеет вид:

найти

$$\max \left\{ f(\bar{x}) = \sum_{j=1}^n c_j x_j + c_0 \right\} \quad (1.11)$$

при ограничениях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m}; \quad (1.12)$$

$$x_{ij} \geq 0 \text{ — целы, } i = \overline{1, n}. \quad (1.13)$$

Искомое отсечение представляет собой линейное уравнение и строится по i -й строке симплекс-таблицы. Очевидно, i -я строка соответствует некоторой дробной базисной переменной. Отсечение имеет вид

$$x_{n+1} = -\{\beta_i\} + \sum_{j \in \omega} \{\alpha_{ij}\} x_j, \quad (1.14)$$

где x_{n+1} — дополнительная переменная, на которую также накладывается условие целочисленности; $\{\alpha_{ij}\}$, $\{\beta_i\}$ — дробные части от соответствующих чисел; ω — множество индексов свободных переменных.

Как известно, дробная часть положительного числа d определяется по формуле:

$$\{d\} = d - [d],$$

где $[d]$ — ближайшее целое число, не превосходящее d (для отрицательных чисел последнее соотношение может не выполняться).

Например, для числа 5,8 будем иметь: $\{d\} = 0,8$, для числа $-5,9$, соответственно, получим, $\{d\} = 0,1$.

Очевидно, что $0 \leq \{d\} < 1$ и $d = \{d\} + [d]$.

Докажем, что построенное отсечение (1.14) исключает текущее оптимальное решение, но сохраняет все допустимые значения переменных в целочисленной задаче. Такое отсечение является *правильным*.

Введем понятие правильного отсечения более строго.

Определение 1.3. Пусть \bar{x}^* – оптимальное решение задачи (1.11), (1.12), не являющееся целочисленным. Неравенство

$$\sum_{j=1}^n \mu_j x_j \leq \mu_0$$

называется *правильным отсечением*, если оно удовлетворяет требованиям:

- 1) для вектора \bar{x}^* неравенство не выполняется;
- 2) если \bar{x} – целочисленное допустимое решение задачи (т.е. удовлетворяет условиям (1.12)), то для \bar{x} неравенство выполняется.

Теорема. Пусть \bar{x}^* – оптимальное решение задачи (1.11), (1.12). Если \bar{x}_i^* – нецелое число, то неравенство

$$\sum_{j \in \omega} \{\alpha_{ij}\} x_j \geq \{\beta_i\} \quad (1.15)$$

является *правильным отсечением*.

(Уравнение (1.14) получается из неравенства (1.15) при добавлении ослабляющей переменной \bar{x}_{n+1} , что необходимо для приведения задачи к каноническому виду при решении симплекс-методом.)

Доказательство.

а) Пусть \bar{x}^* – базисное решение. Тогда

$$\begin{cases} x_i^* = \beta_i - \text{нецелое, следовательно, } \{\beta_i\} > 0; \\ x_j^* = 0, j \in \omega. \end{cases} \quad (1.16)$$

Таким образом, получим,

$$\sum_{j \in \omega} \{\alpha_{ij}\} x_j = 0,$$

т.е. из неравенства (1.15) следует, что $\{\beta_i\} \leq 0$. Это противоречит условию (1.16).

б) Пусть \bar{x} – целочисленное, допустимое решение задачи. Следовательно, оно удовлетворяет системе ограничений. Рассмотрим i -ю строку системы ограничений, а именно:

$$x_i = \beta_i - \sum_{j \in \omega} \alpha_{ij} x_j,$$

что эквивалентно

$$x_i = [\beta_i] + \{\beta_i\} - \sum_{j \in \omega} ([\alpha_{ij}] + \{\alpha_{ij}\}) x_j.$$

Запишем это уравнение по-другому:

$$x_i - [\beta_i] + \sum_{j \in \omega} [\alpha_{ij}] x_j = \{\beta_i\} - \sum_{j \in \omega} \{\alpha_{ij}\} x_j.$$

Итак, левая часть уравнения – целое число, следовательно, правая часть уравнения (обозначим ее через z) – тоже целое число:

$$z = \{\beta_i\} - \sum_{j \in \omega} \{\alpha_{ij}\} x_j \text{ – целое.}$$

Причем $z < 1$, так как $\{\beta_i\} < 1$, а $\{\alpha_{ij}\} \geq 0$ и $x_j \geq 0$.

Отсюда следует, что $-z > -1$, но так как $-z$ – целое число, то $-z \geq 0$, или, с учетом предыдущего выражения,

$$-\{\beta_i\} + \sum_{j \in \omega} \{\alpha_{ij}\} x_j \geq 0, \text{ т.е. } \sum_{j \in \omega} \{\alpha_{ij}\} x_j \geq \{\beta_i\}.$$

Таким образом, доказано, что любое целочисленное допустимое решение задачи удовлетворяет неравенству (1.15).

Алгоритм метода отсечения.

1. Найти оптимальное решение задачи линейного программирования (1.11), (1.12) без учета условия целочисленности.

Если все базисные переменные полученного решения – целые, то целочисленная задача решена. В противном случае перейти к шагу 2.

2. Выбрать (в оптимальном решении) дробную базисную переменную с максимальной дробной частью x_i и построить дополнительное линейное ограничение по i -й строке последней симплекс-таблицы

$$x_{n+1} = -\{\beta_i\} + \sum_{j \in \omega} \{\alpha_{ij}\} x_j.$$

3. Добавить к исходной задаче (1.11) – (1.13) новое ограничение и перейти к шагу 1. При этом исходной симплекс-таблицей для новой задачи служит последняя симплекс-таблица предыдущей задачи, в которую добавлены строка и столбец, соответствующие новому ограничению.

Можно показать, что в результате применения такого алгоритма через конечное число шагов либо будет найдено целочисленное решение, либо будет обнаружена неразрешимость задачи.

Замечания.

1. После введения отсечения в число ограничений задача становится задачей с недопустимым базисом, так как появляется отрицательный свободный член $-\{\beta_i\}$, поэтому решение задачи осуществляется двойственным симплекс-методом.

2. По мере ввода дополнительных ограничений размерность решаемой задачи возрастает. Ограничить рост размерности можно следующим образом: если в базисе осталась переменная, введенная как дополнительная на предыдущем шаге, то надо вычеркнуть соответствующие ей строку и столбец.

1.7.4. Метод ветвей и границ

Метод ветвей и границ заключается в следующем: множество допустимых решений делят на два непересекающихся подмножества, в каждом из которых вычисляют оптимальные значения целевой функции. Из двух подмножеств выбирают то, для которого значение целевой функции лучше. Выбранное подмножество снова делят на две части. В результате процесс поиска экстремума разветвляется. При этом возможен возврат к отбрасываемым ветвям.

На рис. 1.8 приведен пример поиска оптимального решения методом ветвей и границ.

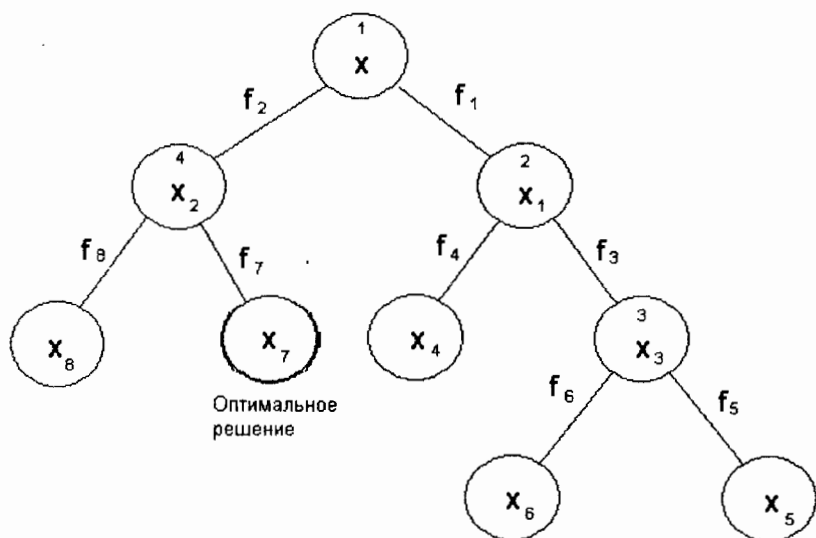


Рис. 1.8. Принцип формирования дерева метода ветвей и границ

Для данного примера, допустимое решение f_1 лучше f_2 , однако решения f_6 и f_5 , которые получены из решения f_1 , оказались хуже чем f_2 . В результате оптимальным решением является f_7 .

Как правило, метод ветвей и границ применяется к задачам следующего вида:
найти

$$\max \left\{ f(\bar{x}) = c_0 + \sum_{j=1}^n c_j x_j \right\}$$

при ограничениях:

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

$$M_j \leq x_j \leq N_j, \quad j = \overline{1, n}; \quad x_j \geq 0 - \text{целые}.$$

Числа M_j, N_j – целые числа. Возможна ситуация, когда $M_j = 0$, а $N_j = \infty$. В этом случае последнее условие просто эквивалентно условию неотрицательности \bar{x} .

Идея метода такова: сначала решается обычная задача линейного программирования. Если решение не целое, т.е. имеется дробная базисная переменная, например x_i^* , тогда из ограничения $M_i \leq x_i \leq N_i$ для x_i формируют два ограничения:

$$[x_i^*] + 1 \leq x_i \leq N_i \quad \text{и} \quad M_i \leq x_i \leq [x_i^*].$$

Здесь $[x_i^*]$ – целая часть переменной x_i^* .

Это можно проиллюстрировать рис. 1.9.

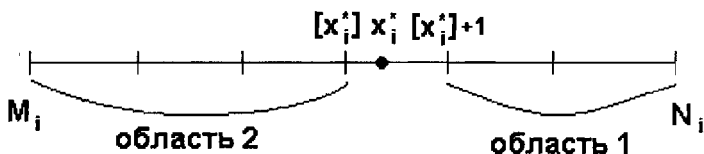


Рис. 1.9. Формирование областей поиска решения

Итак, в результате получаются две задачи линейного программирования. Процесс деления последовательно продолжается.

Алгоритм метода ветвей и границ.

0-я итерация. Первоначальный список задач состоит из исходной задачи без условия целочисленности. За начальную оценку целевой функции $\hat{f}^1(\bar{x})$ берут либо очевидное минимальное значение целевой функции при допустимых значениях переменных, либо $\hat{f}^1(x) = -\infty$ в трудных случаях.

k-я итерация. 1. Выбрать из основного списка и решить очередную задачу с помощью симплекс алгоритма. Если список пуст, то последняя оценка целевой функции и соответствующие ей значения переменных представляют собой искомое целочисленное решение задачи.

2. Если выбранная задача не имеет решения или если полученное оптимальное значение целевой функции меньше текущей оценки $\hat{f}^k(\bar{x})$, то оставить в качестве оценки целевой функции $\hat{f}^k(\bar{x})$, т.е. положить $\hat{f}^{k+1}(\bar{x}_{k+1}) = \hat{f}^k(\bar{x})$. Перейти к шагу 1.

В противном случае, т.е. если выбранная задача имеет решение и оптимальное значение целевой функции больше оценки $\hat{f}^k(\bar{x})$, перейти к шагу 3.

3. Если полученное оптимальное решение задачи – целочисленное, то зафиксировать его, принять в качестве новой оценки $\hat{f}^{k+1}(\bar{x})$ полученное оптимальное значение целевой функции и вернуться к шагу 1.

В противном случае, т.е. если решение задачи не является целочисленным, перейти к шагу 4.

4. Выбрать любую дробную переменную x_i^* из полученного оптимального решения и дополнить основной список двумя новыми задачами, отличающимися от решенной только i -м ограничением.

В первой задаче i -е ограничение имеет вид:

$$M_i \leq x_i \leq \lfloor x_i^* \rfloor,$$

а во второй

$$\lfloor x_i^* \rfloor + 1 \leq x_i \leq N_i.$$

5. Сохранить имеющуюся оценку целевой функции для следующей итерации, т.е. положить $\hat{f}^{k+1}(\bar{x}) = \hat{f}^k(\bar{x})$, и перейти к шагу 1.

1.7.5. Метод зондирования решений

Данным методом решают задачу целочисленного программирования с булевыми переменными:

найти

$$\max \left\{ f(\bar{x}) = \sum_{j=1}^n c_j x_j \right\}$$

при ограничениях:

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

$$x_j = \begin{cases} 0, \\ 1, \end{cases} \quad j = \overline{1, n}.$$

Решением задачи является набор переменных $\{x_1, x_2, \dots, x_n\}$, каждая из которых принимает значение 0 или 1. Всего существует 2^n таких наборов. Для получения решения поставленной задачи линейного программирования, в сущности, необходимо перебрать все эти наборы. Непосредственный перебор приводит к длительным вычислениям. Метод зондирования позволяет эти вычисления сократить. Метод относится к классу «ветвей и границ».

Совокупность ℓ переменных, где $1 \leq \ell \leq n$, принимающих значения 0 или 1, называется *пробным зондированием*. Переменные, не входящие в пробное решение, называются *свободными*. Любой конкретный набор свободных переменных образует *дополнение пробного решения*. Если имеется n переменных, ℓ из которых входят в пробное решение, то можно построить $2^{n-\ell}$ дополнений пробного решения.

Пусть $n = 8$ и пробное решение имеет вид $\{x_3 = 1, x_6 = 0\}$, тогда число свободных переменных равно $n - \ell = 8 - 2 = 6$. Произвольным дополнением пробного решения будет, например, такой набор свободных переменных

$$\{x_1 = 1, x_2 = 0, x_4 = 0, x_5 = 1, x_7 = 1\}.$$

Всего же можно составить $2^6 = 64$ дополнений пробного решения.

Пусть каким-то образом выбрано пробное решение. *Зондированием* называется проверка всех его дополнений на допустимость и оптимальность. Для сокращения объема вычислений при построении алгоритма зондирования используют две теоремы, позволяющие сделать заключение о невозможности найти дополнение, которое улучшает текущую оценку целевой функции,

или позволяет включить в пробное решение какую-нибудь свободную переменную.

Пример 1.18. Пусть требуется найти

$$\max \{f(\bar{x}) = x_1 + 8x_2 + 4x_3 + 9x_4 - x_5 + 3x_6\}.$$

Предположим, что оценка целевой функции на k -й итерации $\hat{f}^k(\bar{x}) = 15$.

Рассмотрим пробное решение $\{x_2 = 0, x_3 = 1, x_4 = 0\}$, оценка целевой функции, следовательно, будет

$$\hat{f}^{k+1}(\bar{x}) = x_1 + 4 - x_5 + 3x_6.$$

Максимальное значение целевой функции, которое при этом возможно (при максимальных значениях свободных переменных $x_1 = 1, x_5 = 0, x_6 = 1$) — это $f(\bar{x}) = 8$. А текущая оценка $\hat{f}^k(\bar{x}) = 15$. Следовательно, нельзя подобрать дополнения пробного решения, увеличивающего значение целевой функции.

Теорема дополнения. Дополнения, улучшающего пробное решение, не существует, если выполняется хотя бы одно из неравенств:

$$\sum_{\substack{\text{по свободным} \\ \text{переменным}}} \min(a_{ij}, 0) > b_i - \sum_{\substack{\text{по пробным} \\ \text{переменным}}} a_{ij} x_j, \quad i = \overline{1, m+1};$$

где

$$a_{m+1,j} = -c_j, \quad b_{m+1} = -\hat{f}^k(\bar{x}) - 1.$$

Теорема расширения. Свободную переменную x_k можно включить в пробное решение, если хотя бы при одном i выполняется неравенство

$$\sum_{\substack{\text{по свободным} \\ \text{переменным}}} \min(a_{ij}, 0) + |a_{ik}| > b_i - \sum_{\substack{\text{по пробным} \\ \text{переменным}}} a_{ij} x_j, \quad i = \overline{1, m+1}.$$

Причем

$$\begin{aligned} x_k &= 0, & \text{если } a_{ik} &\geq 0; \\ x_k &= 1, & \text{если } a_{ik} &< 0. \end{aligned}$$

Алгоритм метода зондирования решений.

0-я итерация. 1. Взять в качестве оценки целевой функции $\hat{f}^1(\bar{x})$ нуль или сумму всех отрицательных коэффициентов целевой функции (в трудных случаях $\hat{f}^1(\bar{x}) = -\infty$).

2. Сформировать первое пробное решение, состоящее из одной переменной. Переменную x_ℓ , входящую в первое пробное решение, выбрать произвольно или, чтобы улучшить сходимость, выбрать переменную, имеющую максимальный коэффициент в целевой функции. Занести в основной список две задачи. В одной из них $x_\ell = 0$, в другой – $x_\ell = 1$.

k-я итерация. 1. Прекратить вычисления, если основной список пуст. В противном случае выбрать задачу из основного списка.

2. С помощью теоремы дополнения установить, существует ли допустимое дополнение пробного решения, у которого значение целевой функции превосходит текущую оценку $\hat{f}^k(\bar{x})$. Если такого дополнения не существует оставить оценку $\hat{f}^{k+1}(\bar{x}) = \hat{f}^k(\bar{x})$ и вернуться к шагу 1.

В противном случае перейти к шагу 3.

3. С помощью теоремы расширения найти свободные переменные, которые следует включить в пробное решение.

4. Если расширенное пробное решение является полным, т.е. содержит все n переменных, то (если значение целевой функции превышает оценку) зафиксировать его, принять оценку $\hat{f}^{k+1}(\bar{x})$ равной соответствующему данному решению значению целевой функции и вернуться к шагу 1.

В противном случае перейти к шагу 5.

5. Выбрать любую свободную переменную x_k , не входящую в пробное решение. Внести в основной список две задачи. В одной из них $x_k = 0$, в другой $x_k = 1$.

6. Если в полученных задачах пробные решения не являются полными, то взять значение $\hat{f}^{k+1}(\bar{x}) = \hat{f}^k(\bar{x})$ и вернуться к шагу 3.

Если пробные решения обеих задач полные, то проверить каждое из них на допустимость. Для допустимых решений посчитать значения целевой функции. Сравнить их с оценкой

$\hat{f}^k(\bar{x})$. Выбрать и зафиксировать решение, у которого $f(\bar{x}) > \hat{f}^k(\bar{x})$, положить $\hat{f}^{k+1}(\bar{x}) = f(\bar{x})$. Если таких решений нет, то положить $\hat{f}^{k+1}(\bar{x}) = \hat{f}^k(\bar{x})$, а полученные задачи отбросить.

Перейти к шагу 1.

1.7.6. Примеры решения задач целочисленного программирования

Пример 1.19. Метод отсечения.

Найти

$$\max \{f(\bar{x}) = 21x_1 + 11x_2\}$$

при ограничениях:

$$7x_1 + 4x_2 \leq 13,$$

$$x_1, x_2 \geq 0 - \text{целые}.$$

В канонической форме задача выглядит так:

найти

$$\max \{f(\bar{x}) = 21x_1 + 11x_2\}$$

при ограничениях:

$$7x_1 + 4x_2 + x_3 = 13;$$

$$x_1, x_2, x_3 \geq 0 - \text{целые}.$$

1. Решение задачи симплекс-методом.

Составим симплекс-таблицу (табл. 1.38).

Таблица 1.38

Базис	Свободный член	x_1	x_2	x_3
x_3	13	7	4	1
$f(\bar{x})$	0	-21	-11	0

Преобразуя табл. 1.38 в соответствии с симплекс-методом, получим табл. 1.39.

Базис	Свободный член	x_1	x_2	x_3
x_1	13/7	1	4/7	1/7
$f(\bar{x})$	39	0	1	3

Исходя из последней симплекс-таблицы, оптимальным решением будет: $x_1 = \frac{13}{7}$, $x_2 = x_3 = 0$, $f(\bar{x}) = 39$. Очевидно, x_1 – дробное число.

2. Введем отсечение по строке для x_1

$$x_4 = -\frac{6}{7} + \frac{4}{7}x_2 + \frac{1}{7}x_3.$$

Построим следующую симплекс-таблицу (табл. 1.40).

Таблица 1.40

Базис	Свободный член	x_1	x_2	x_3	x_4
x_1	13/7	1	4/7	1/7	0
x_4	-6/7	0	4/7	-1/7	1
$f(\bar{x})$	39	0	1	3	0

Преобразовав табл. 1.40 в соответствии с симплекс-методом, получим табл. 1.41.

Таблица 1.41

Базис	Свободный член	x_1	x_2	x_3	x_4
x_1	1	1	0	0	1
x_2	3/2	0	1	1/4	-7/4
$f(\bar{x})$	75/2	0	0	11/4	7/4

Из табл. 1.41 следует, что оптимальное решение:

$$x_1 = 1, \quad x_2 = 3/2, \quad x_4 = x_3 = 0, \quad f(x) = 75/2$$

3. Введем отсечение по строке для x_2 , в результате получим:

$$x_5 = -1/2 + 1/4x_3 + 1/4x_4;$$

$$\{-7/4\} = -7/4 - (-8/4) = 1/4.$$

Используя симплекс-метод, проведем ряд последовательных преобразований табл. 1.41. Получим табл. 1.42 – 1.44.

Таблица 1.42

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_1	1	1	0	0	1	0
x_2	3/2	0	1	1/4	-7/4	0
x_5	-1/2	0	0	-1/4	-1/4	1
$f(\bar{x})$	75/2	0	0	11/4	7/4	0

Таблица 1.43

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_1	-1	1	0	-1	0	4
x_2	5	0	1	2	0	-7
x_4	2	0	0	1	1	-4
$f(\bar{x})$	34	0	0	1	0	7

Таблица 1.44

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5
x_3	1	1	0	1	0	-4
x_2	3	2	1	0	0	1
x_4	1	1	0	0	1	0
$f(\bar{x})$	33	1	0	0	0	11

Таким образом, получаем следующее оптимальное решение исходной задачи:

$$x_1^* = 0, \quad x_2^* = 3, \quad f(\bar{x}^*) = 33.$$

Пример 1.20. Метод ветвей и границ.

Найти

$$\max \{f(\bar{x}) = 5x_1 - x_2\} \quad (1)$$

при ограничениях:

$$\begin{cases} 3x_1 - 8x_2 \leq 3; & (2) \\ -4x_1 - 8x_2 \leq -12; & (3) \\ 7x_1 + 6x_2 \leq 42; & (4) \\ 0 \leq x_1 \leq 9; \\ 0 \leq x_2 \leq 9; \\ x_1, x_2 - \text{целые.} \end{cases}$$

В качестве начальной оценки целевой функции возьмем $\hat{f}^1(\bar{x}) = -\infty$, так как найти явно минимальное значение затруднительно.

1-я итерация. Очевидно, рассмотренная ранее задача 1.10 совпадает с исходной задачей без условия целочисленности.

Задача 1.11. Учитывая исходную систему ограничений, имеем:

$$\begin{cases} x_3 = 3 - (3x_1 - 8x_2); \\ x_4 = -12 - (-4x_1 - 8x_2); \\ x_5 = 42 - (7x_1 + 6x_2); \\ x_6 = 9 - x_1; \\ x_7 = 9 - x_2. \end{cases}$$

На основе симплекс-метода построим табл. 1.45.

Таблица 1.45

Базис	Свободный член	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_3	3	3	8	1	0	0	0	0
x_4	-12	-4	-8	0	1	0	0	0
x_5	42	7	6	0	0	1	0	0
x_6	9	1	0	0	0	0	1	0
x_7	9	0	1	0	0	0	0	1
$f(\bar{x})$	0	-5	1	0	0	0	0	0

Так как коэффициенты в строке целевой функции не все положительны, то двойственный алгоритм симплекс-метода не пригоден. Для решения задачи воспользуемся методом искусственного базиса. При этом система ограничений приводится к виду:

$$\begin{cases} y_1 = 3 - (3x_1 - 8x_2 + x_3); \\ y_2 = 12 - (4x_1 + 8x_2 - x_4); \\ y_3 = 42 - (7x_1 + 6x_2 + x_5); \\ y_4 = 9 - (x_1 + x_6); \\ y_5 = 9 - (x_2 + x_7). \end{cases}$$

Целевая функция имеет вид

$$\begin{aligned} \Phi(\bar{y}) &= y_1 + y_2 + y_3 + y_4 + y_5 = \\ &= 75 - (16x_1 + 6x_2 + x_3 - x_4 + x_5 + x_6 + x_7). \end{aligned}$$

Оптимальным решением задачи 1 будет:

$$x_1 = 4,78; \quad x_2 = 1,42; \quad f(\bar{x}) = 22,5.$$

Полученное решение не является целочисленным, значение $f(\bar{x}) > f^1(\bar{x}) = -\infty$, следовательно, необходимо принять новую оценку целевой функции, равной прежней, т.е. $\hat{f}^2(\bar{x}) = -\infty$ и образовать две новые задачи, взяв дробную переменную $x_1 = 4,78$.

Задача 1.12. Соотношения (1), (2), (3), (4) представляют собой систему ограничений, при этом

$$\begin{cases} 0 \leq x_1 \leq 4; \\ 0 \leq x_2 \leq 9. \end{cases}$$

Задача 1.13. Соотношения (1), (2), (3), (4) представляют собой систему ограничений, при этом

$$\begin{cases} 5 \leq x_1 \leq 9; \\ 0 \leq x_2 \leq 9. \end{cases}$$

На рис. 1.10 графически изображены допустимые множества, соответствующие исходной задаче и задачам 1.12 и 1.13.

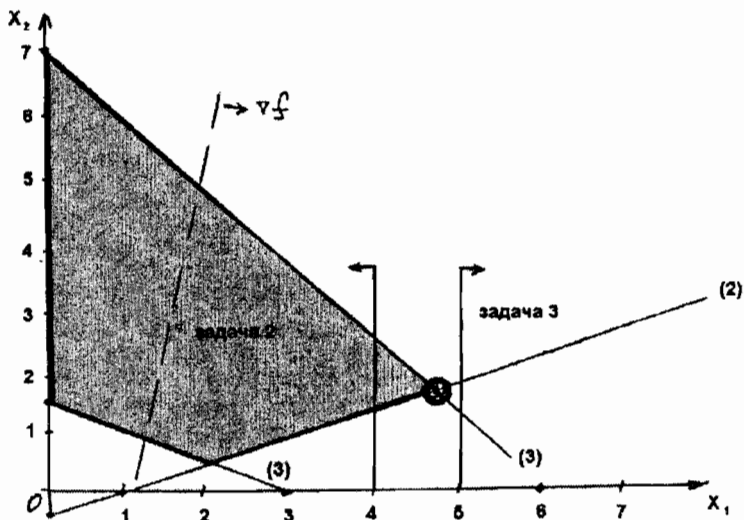


Рис. 1.10. Графическое изображение допустимых множеств задач 1, 2, 3

2-я итерация. Задача 1.13 – решения нет ($\hat{f}^3(\bar{x}) = -\infty$).

3-я итерация. Оптимальное решение задачи 1.12:

$$x_1 = 4; \quad x_2 = 1,125; \quad f(\bar{x}) = 18,875.$$

Как видим, решение не является целочисленным. Причем значение $f(\bar{x}) > \hat{f}^3(\bar{x})$, следовательно, $\hat{f}^4(\bar{x}) = -\infty$ и образуются две задачи, при этом используется дробная переменная $x_1 = 1,125$.

Задача 1.14. Соотношения (1), (2), (3), (4) – система ограничений, при этом

$$\begin{cases} 0 \leq x_1 \leq 4; \\ 0 \leq x_2 \leq 1. \end{cases}$$

Задача 1.15. Соотношения (1), (2), (3), (4) – система ограничений, при этом

$$\begin{cases} 0 \leq x_1 \leq 4; \\ 2 \leq x_2 \leq 9. \end{cases}$$

Изобразим графически допустимые множества задач 1.14 и 1.15 (рис. 1.11).

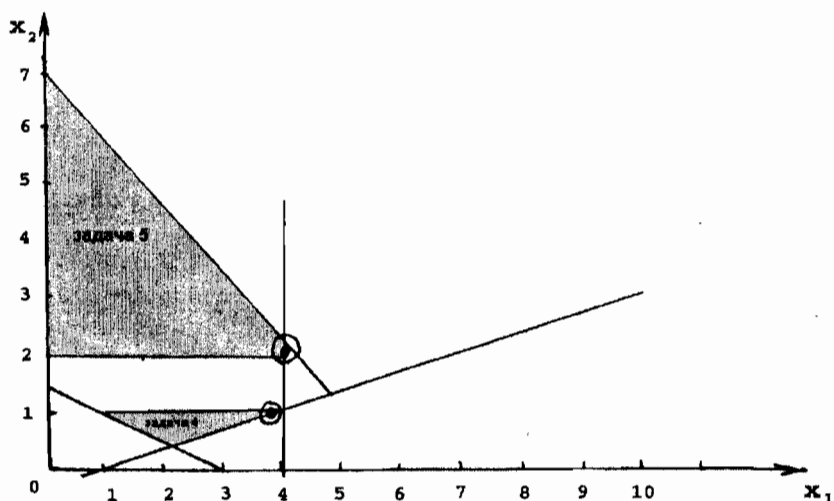


Рис. 1.11. Графическое изображение допустимых множеств задач 1.14 и 1.15

4-я итерация. Оптимальное решение задачи 1.14:

$$x_1 = 3,66, \quad x_2 = 1, \quad f(\bar{x}) = 17,33.$$

Решение не целочисленное, значение $f(\bar{x}) > \hat{f}^4(\bar{x})$, следовательно, $\hat{f}^5(\bar{x}) = -\infty$, при этом образуются две задачи.

Задача 1.16. Соотношения (1), (2), (3), (4) — система ограничений, при этом

$$\begin{cases} 0 \leq x_1 \leq 3; \\ 0 \leq x_2 \leq 1. \end{cases}$$

Задача 1.17. Соотношения (1), (2), (3), (4) — система ограничений, при этом

$$\begin{cases} x_1 = 4; \\ 0 \leq x_2 \leq 1. \end{cases}$$

Очевидно, оптимальным будет решение, соответствующее решению задач 1.15, 1.16 или 1.17.

5-я итерация. *Оптимальное решение задачи 1.15:*

$$x_1 = 4, \quad x_2 = 2, \quad f(\bar{x}) = 18.$$

Это целочисленное решение, следовательно, в качестве оценки целевой функции возьмем $\hat{f}^6(x) = 18$.

6-я итерация. *Оптимальное решение задачи 1.16:*

$$x_1 = 3, \quad x_2 = 0,75, \quad f(\bar{x}) = 14,75.$$

Очевидно, $f(\bar{x}) < \hat{f}^6(\bar{x})$, поэтому переходим в основной список задач, где осталась одна нерешенная задача 1.17.

7-я итерация. *Задача 1.17 – не имеет решения.*

8-я итерация. Итак, основной список пуст, следовательно, *оптимальное целочисленное решение:*

$$x_1 = 4, \quad x_2 = 2, \quad f(\bar{x}) = 18.$$

Решение задачи методом ветвей и границ хорошо иллюстрировать с помощью дерева решения (рис. 1.12).

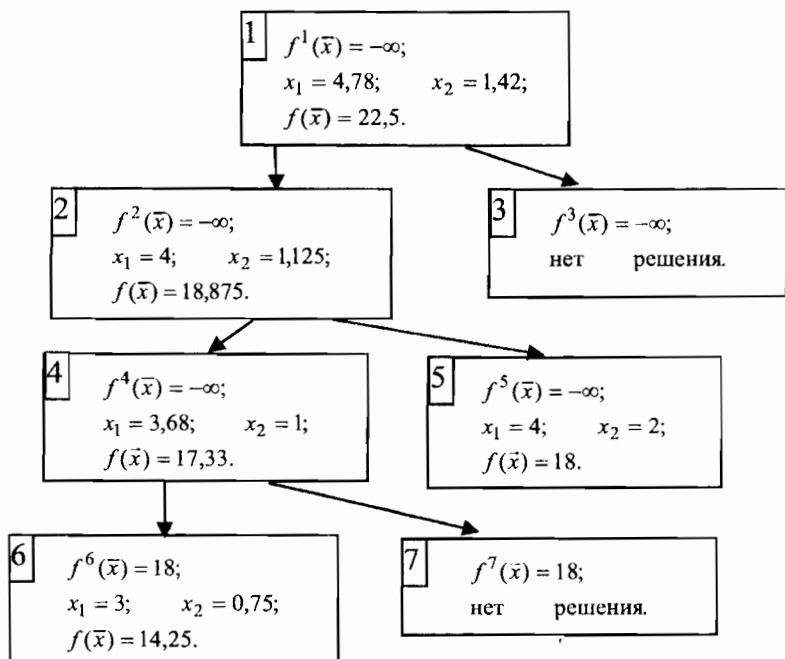


Рис. 1.12. Дерево решения задачи методом ветвей и границ

Глава 2

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ В ПРИКЛАДНЫХ ЗАДАЧАХ ОПТИМИЗАЦИИ

2.1. Применение линейного программирования в теоретико-игровых методах исследования сложных систем

2.1.1. Теоретические основы матричных игр

Матричной игрой будем называть антагонистическую игру, в которой каждый игрок имеет конечное множество стратегий.

Как известно [1], игра двух лиц задается прямоугольной матрицей:

$$A = \|a_{ij}\|, \quad i = \overline{1, m}, \quad j = \overline{1, n},$$

где a_{ij} есть значение выигрыша игрока 1, если он выбрал свою i -ю стратегию, а игрок 2 выбрал свою j -ю стратегию. Данная игра называется игрой размерности $(m \times n)$, а матрица A – *платежной матрицей*.

Если игрок 1 выбирает номер строки i , а второй – j , то в результате выбора игроками независимых стратегий игрок 2 платит игроку 1 выигрыш a_{ij} . Следовательно, игрок 1 может гарантировать себе выигрыш не менее значения: $v_1 = \max_i \min_j a_{ij}$ – *нижняя цена игры*, а второй гарантировать себе проигрыш не более величины $v_2 = \min_j \max_i a_{ij}$ – *верхняя цена игры*. В общем случае $v_1 \leq v \leq v_2$.

Если $\min_j \max_i a_{ij} = \max_i \min_j a_{ij} = a_{i^* j^*} = v$, то в такой игре минимаксные стратегии i^* и j^* являются оптимальными, так как

решения о принятых стратегиях игроков получены независимо. В этом случае решение игры является ситуацией равновесия, а стратегии i^* и j^* называются чистыми стратегиями.

Если $\min_j \max_i a_{ij} > \max_i \min_j a_{ij}$, то в игре нет ситуации равновесия в чистых стратегиях. Для разрешения данной ситуации вводятся смешанные стратегии игроков.

Смешанной стратегией игрока 1 называется вектор $X = (x_1, \dots, x_m)^T$, удовлетворяющий условиям:

$$\sum_{i=1}^m x_i = 1, x_i \geq 0, \quad i = \overline{1, n},$$

где число x_i – вероятность, с которой игрок 1 выбирает свою i -ю чистую стратегию.

Величина

$$H(X, Y) = \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_i y_j = X^T A Y \quad (2.1)$$

представляет собой математическое ожидание выигрыша 1-го игрока в ситуации (X, Y) .

Ситуация (X^*, Y^*) называется *ситуацией равновесия* в смешанном расширении матричной игры, если для любых X и Y выполняются неравенства

$$H(X, Y^*) \leq H(X^*, Y^*) \leq H(X^*, Y). \quad (2.2)$$

В матричной игре может быть несколько ситуаций равновесия.

Можно доказать следующее свойство:

пусть $X^* \in S_m$, $Y^* \in S_n$, v – действительное число, тогда для того чтобы X^*, Y^* были оптимальными стратегиями, а v – ценой игры, необходимо и достаточно, чтобы выполнялись соотношения:

$$\begin{cases} H(i, Y^*) = \sum_{j=1}^n a_{ij} y_j^* \leq v, & i = \overline{1, m}; \\ H(X^*, j) = \sum_{i=1}^m a_{ij} x_i^* \geq v & j = \overline{1, n}. \end{cases} \quad (2.3)$$

Под решением матричной игры будем понимать нахождение векторов X и Y , а также значения цены игры v .

Итак, пусть дана платежная матрица

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m1} & \dots & a_{mn} \end{pmatrix}.$$

Положим, что решения в чистых стратегиях нет.

Будем искать решение игры $(m \times n)$ в смешанных стратегиях в виде:

$$\begin{aligned} X &= (x_1, \dots, x_m)^T, & x_i &\geq 0, & i &= \overline{1, m}; \\ Y &= (y_1, \dots, y_n)^T, & y_j &\geq 0, & j &= \overline{1, n} \end{aligned} \quad (2.4)$$

при ограничениях:

$$\sum_{i=1}^m x_i = 1, \quad \sum_{j=1}^n y_j = 1.$$

Тогда если X^*, Y^* – оптимальные смешанные стратегии первой и второй стороны, то, в соответствии с соотношениями (2.3), выполняются неравенства:

$$\begin{cases} \sum_{j=1}^n a_{ij} y_j^* \leq v, & i = \overline{1, m}; \\ \sum_{i=1}^m a_{ij} x_i^* \geq v, & j = \overline{1, n}, \end{cases} \quad (2.5)$$

где v – цена игры.

2.1.2. Сведение матричной игры к задаче линейного программирования

Как правило, любая матричная игра может быть сведена к паре двойственных задач линейного программирования. Чтобы преобразование было корректным, необходимо выполнение условия $v > 0$. Это условие обязательно выполняется, если все элементы платежной матрицы неотрицательны. Добиться неотрицательности можно путем прибавления достаточно большой константы ко всем элементам матрицы, что не меняет множества оптимальных стратегий игроков, так как приводит к игре стратегически эквивалентной исходной [1].

В теории игр доказывается следующая теорема.

Теорема 2.1. Пусть множество V' состоит из тех и только тех чисел v' , для которых существует такая стратегия Y игрока 2, что справедливы неравенства

$$H(i, Y) \leq v', \quad i = \overline{1, m}, \quad (2.6)$$

тогда значение цены матричной игры v равно наименьшему из чисел множества V' , а вектор Y^* , для которого эти неравенства справедливы при $v' = v$, является оптимальной стратегией игрока 2.

Таким образом, чтобы найти v и Y^* , надо определить минимальное значение v' , удовлетворяющее неравенствам (2.6), которые удобнее записать так:

$$\sum_{j=1}^n a_{ij} y_j \leq v', \quad i = \overline{1, m}. \quad (2.7)$$

Разделим обе части неравенства (2.7) на v' и положим $u_j = y_j / v'$, тогда оно примет вид

$$\sum_{j=1}^n a_{ij} u_j \leq 1, \quad i = \overline{1, m}. \quad (2.8)$$

С учетом того, что

$$\sum_{j=1}^n u_j = \frac{1}{v'} \sum_{j=1}^n y_j = \frac{1}{v'}, \quad (2.9)$$

задача минимизации v' сводится к задаче максимизации $\frac{1}{v'}$ или, что то же самое, максимизации функции $f(\bar{u})$, т.е. получили задачу линейного программирования, которую можно сформулировать в виде:

найти

$$\max \left\{ f(\bar{u}) = \sum_{j=1}^n u_j \right\} \quad (2.10)$$

при ограничениях:

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

Пусть $\bar{u}^* = (u_1^*, \dots, u_n^*)$ – оптимальное решение полученной задачи линейного программирования, тогда искомая цена игры определяется соотношением:

$$v = \frac{1}{f(\bar{u}^*)}. \quad (2.11)$$

Исходя из этого, компоненты оптимальной стратегии второго игрока можно представить в виде

$$y_j^* = v u_j^*, \quad j = \overline{1, n}. \quad (2.12)$$

Необходимость упомянутого ранее условия неотрицательности ($v > 0$) для корректности проведенных преобразований очевидна.

Приведенную ранее теорему 2.1 можно аналогичным образом применить и к стратегиям первого игрока. Из нее следует, что для определения цены игры v и оптимальной стратегии X^* необходимо найти максимальное значение числа v'' , удовлетворяющее неравенствам:

$$\sum_{i=1}^m a_{ij} x_i \geq v'', \quad j = \overline{1, n}.$$

В результате, полагая $t_i = x_i / v$, приходим к следующей задаче линейного программирования:

найти

$$\min \left\{ \varphi(\bar{t}) = \sum_{i=1}^m t_i \right\} \quad (2.13)$$

при ограничениях:

$$\begin{cases} \sum_{i=1}^m a_{ij} t_i \geq 1, & j = \overline{1, n}; \\ t_i \geq 0, & i = \overline{1, m}. \end{cases}$$

Тогда если $\bar{t}^* = (t_1^*, \dots, t_m^*)$ – оптимальное решение этой задачи, то цена игры определяется соотношением, аналогичным выражению (2.11):

$$v = \frac{1}{\varphi(\bar{t}^*)}, \quad (2.14)$$

и, соответственно, компоненты оптимальной стратегии первого игрока задаются в виде

$$x_i^* = v t_i^*, \quad i = \overline{1, m}. \quad (2.15)$$

Как видно, вторая задача (2.13) является двойственной к первой (2.10), поэтому справедливо равенство:

$$f(\bar{u}^*) = \varphi(\bar{t}^*). \quad (2.16)$$

Как отмечено в разд. 1.5, решение двойственных задач линейного программирования осуществляется одновременно, путем однократного применения симплекс-метода.

Таким образом, сведение любой матричной игры к задаче линейного программирования приводит к необходимости, с одной стороны, максимизации функции $f(\bar{u})$:

$$\max \left\{ f(\bar{u}) = \sum_{j=1}^n u_j \right\}$$

при ограничениях:

$$\begin{cases} \sum_{j=1}^n a_{ij} u_j \leq 1 & i = \overline{1, m}; \\ u_j \geq 0, & j = \overline{1, n}, \end{cases}$$

где $u_j = y_j / v'$.

С другой стороны, минимизации функции $\varphi(\bar{t})$:

$$\min \left\{ \varphi(\bar{t}) = \sum_{i=1}^m t_i \right\}$$

при ограничениях:

$$\begin{cases} \sum_{i=1}^m a_{ij} t_i \geq 1, & j = \overline{1, n}; \\ t_i \geq 0, & i = \overline{1, m}, \end{cases}$$

где $t_i = x_i / v''$.

Легко видеть, что вторая задача является двойственной к первой, более того, и первая и вторая задача прекрасно сводятся к канонической форме записи задачи линейного программирования, которая успешно решается методами, изложенными в гл. 1.

2.2. Использование методов линейного программирования при решении задач перебазирования средств истребительной авиации

2.2.1. Математическая постановка задачи перебазирования

При ведении оборонительных операций правильная организация сосредоточения сил и средств истребительной авиации (ИА) позволяет существенно повысить эффективность обороны охраняемой территории [6].

Задача оптимального распределения средств истребительной авиации может быть формализована и сведена к задаче целочисленного линейного программирования. Рассмотрим математическую постановку задачи.

Пусть $T_1, T_2, \dots, T_j, \dots, T_n$ – множество целей, которые составляют гипотетический налет воздушных объектов (ВО) противника.

Введем понятие «аэродром-тип», которое характеризует расположение аэродрома и тип истребителей, дислоцированных на нем. Имеются также основные аэродромы-типы $A_1, A_2, \dots, A_i, \dots, A_m$, находящиеся в распоряжении тех подразделений, которым поставлена задача уничтожить или ослабить налет противника.

Предположим, что имеется некоторый резерв времени между моментом фиксации факта массированного налета противника и моментом вылета самолетов с основных аэродромов-типов для воздействия по гипотетическому налету целей. За это располагаемое время возможно усилить оборону объектов (т.е. обеспечить более эффективное воздействие по целям), если перебазировать истребители-перехватчики с резервных аэродромов-типов $B_1, B_2, \dots, B_k, \dots, B_l$ на некоторые основные аэродромы-типы.

Обобщив вышесказанное, сформируем следующие множества.

$A = \{A_1, A_2, \dots, A_i, \dots, A_m\}$ – множество основных аэродромов-типов.

Каждый основной аэродром-тип A_i ($i = 1, \dots, m$) характеризуется следующими параметрами:

координатами расположения аэродрома-типа;

типом дислоцированных на нем ЛА;

μ_i – количеством дислоцированных ЛА;

α_i – вместимостью аэродрома (количеством самолетов, которые может принять аэродром, исходя из имеющихся свободных мест, запасов топлива и имеющихся ракет для данного типа истребителя);

Q_i – подмножеством номеров резервных аэродромов-типов, самолеты с которых могут перебазироваться за время, меньшее располагаемого, в точку расположения основного аэродрома-типа.

$B = \{B_1, B_2, \dots, B_k, \dots, B_l\}$ – множество резервных аэродромов-типов.

Каждый резервный аэродром-тип B_k ($k = 1, \dots, l$) характеризуется:

координатами расположения резервного аэродрома-типа;

типом дислоцированных летательных аппаратов (ЛА);

β_k – количеством дислоцированных ЛА;

P_k – подмножеством номеров основных аэродромов-типов, в точки расположения которых возможно перебазирование за время, не более установленного (определяется из штурманских расчетов).

$T = \{T_1, T_2, \dots, T_j, \dots, T_n\}$ – множество целей.

Каждая цель T_j ($j = 1, \dots, n$) характеризуется:

минимальным временем уничтожения j -й цели i -м аэродромом-типом τ_{ij} ($i = 1, \dots, m$) (см. разд. 4.5);

величиной $c_{ij} = 1/\tau_{ij}$ ($i = 1, \dots, m$).

На рис. 2.1 представлена схема, поясняющая принцип осуществления перегруппирования средств истребительной авиации. Так, например, на основной аэродром-тип A_1 за располагаемое время успевают перебазироваться истребители с резервных аэродромов-типов B_1, B_2 , а значит, $Q_1 = \{1; 2\}$, аналогично определяются множества $Q_2 = \{1; 2\}; \dots, Q_i = \{2; k\}; \dots; Q_m = \{k; l\}$. С резервного аэродрома-типа B_1 за располагаемое время успевают перебазироваться истребители на основные аэродромы-типы A_1, A_2 , а значит, $P_1 = \{1; 2\}$, аналогично определяются множества $P_2 = \{1; 2; i\}; \dots, P_k = \{i; m\}; \dots; P_l = \{m\}$.

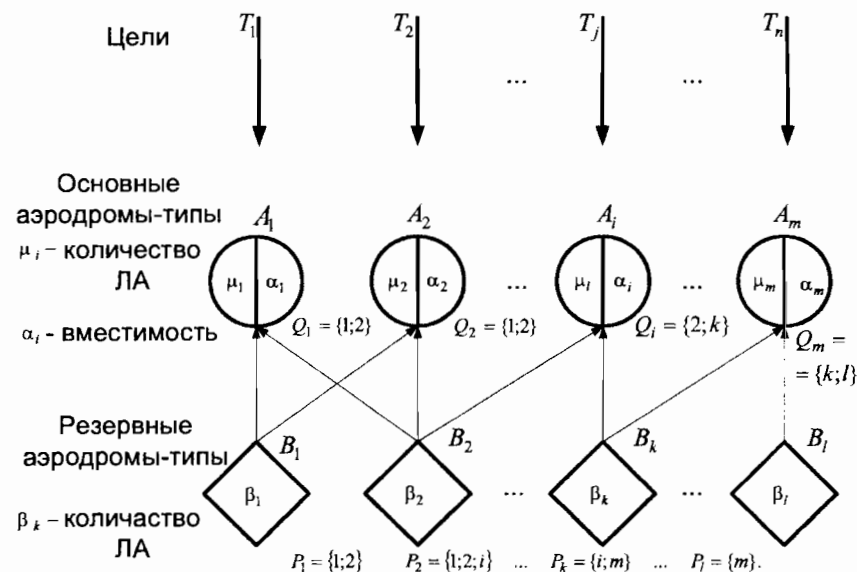


Рис. 2.1. Схема перегруппирования средств ИА

В результате решения задачи необходимо сформировать схему перераспределения истребителей с резервных аэродромов-типов на основные аэродромы-типы. При этом общее время перебазирования должно удовлетворять условию минимума суммарного углубления целей на охраняемую территорию.

Введем обозначения:

1) x_{ij} – количество ЛА с i -го основного аэродрома-типа, назначенных на j -ю цель;

2) y_{kij} – количество ЛА, перебазированных с k -го резервного аэродрома-типа в точку с координатами i -го основного аэродрома-типа и назначенных на j -ю цель;

3) $c_{ij} = 1/\tau_{ij}$; τ_{ij} – минимальное время уничтожения j -й цели перехватчиком с i -го аэродрома-типа (если перехват невозможен, то $\tau_{ij} = \infty$, а $c_{ij} = 0$), подробное описание решения данной задачи приведено в работе [9];

4) $c_{ijk} = 1/\tau_{ijk}$; τ_{ijk} – минимальное время уничтожения j -й цели перехватчиком с i -го аэродрома-типа, который был перебазирован с k -го резервного аэродрома-типа.

Минимальное суммарное углубление целей обеспечивает критерий:

$$F(x_{ij}, y_{kij}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{k=1}^l \sum_{i \in P_k} \sum_{j=1}^n c_{ijk} y_{kij}, \quad (2.17)$$

который необходимо максимизировать.

Кроме того, на искомые параметры наложен ряд ограничений:

с любого i -го основного аэродрома-типа не может быть назначено истребителей больше, чем μ_i :

$$\sum_{j=1}^n x_{ij} \leq \mu_i, \quad i = 1, \dots, m; \quad (2.18)$$

на любую j -ю цель должно быть назначено не более одного истребителя:

$$\sum_{i=1}^m x_{ij} + \sum_{k=1}^l \sum_{i \in P_k} y_{kij} \leq 1, \quad j = 1, \dots, n, \quad (2.19)$$

на любой i -й основной аэродром-тип не может перебазироваться самолетов больше вместимости α_i :

$$\sum_{k \in Q_i} \sum_{j=1}^n y_{kij} \leq \alpha_i, \quad i = 1, \dots, m; \quad (2.20)$$

с любого k -го резервного аэродрома-типа на все допустимые основные аэродромы-типы не может перебазироваться число истребителей, превышающее β_k :

$$\sum_{i \in P_k} \sum_{j=1}^n y_{kij} \leq \beta_k, \quad k = 1, \dots, l; \quad (2.21)$$

кроме того, все неизвестные переменные являются целыми и неотрицательными:

$$x_{ij}, y_{kij} \in N, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad k = 1, \dots, l. \quad (2.22)$$

Таким образом, задача имеет вид:
найти

$$\max \left\{ F(x_{ij}, y_{kij}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{k=1}^l \sum_{i \in P_k} \sum_{j=1}^n c_{ijk} y_{kij} \right\} \quad (2.23)$$

при ограничениях:

$$\left\{ \begin{array}{ll} \sum_{j=1}^n x_{ij} \leq \mu_i, & i = 1, \dots, m; \\ \sum_{i=1}^m x_{ij} + \sum_{k=1}^l \sum_{i \in P_k} y_{kij} \leq 1, & j = 1, \dots, n; \\ \sum_{k \in Q_i} \sum_{j=1}^n y_{kij} \leq \alpha_i, & i = 1, \dots, m; \\ \sum_{i \in P_k} \sum_{j=1}^n y_{kij} \leq \beta_k, & k = 1, \dots, l; \\ x_{ij}, y_{kij} \in N, & i = 1, \dots, m, \quad j = 1, \dots, n, \quad k = 1, \dots, l; \end{array} \right. \quad (2.24)$$

Очевидно, поставленная задача является задачей целочисленного программирования и может быть решена одним из описанных методов. Однако здесь предлагается использовать обычный симплекс-метод, что существенно упрощает решение данной задачи.

2.2.2. Преобразование задачи целочисленного программирования к задаче линейного программирования

Отметим, что в задаче (2.23) – (2.24) переменные x_{ij} , y_{kij} – целые и неотрицательные, т.е. задача (2.23) – (2.24) – задача целочисленного программирования. Однако рассматриваемая задача относится к частному классу задач целочисленного программирования, в которых ограничение целочисленности оказывается несущественным. Здесь необходимо привести некоторые определения и теоремы [7]:

Определение 2.1. Матрица называется вполне унимодулярной, если определитель любой ее невырожденной квадратной подматрицы равен по модулю 1.

Следствие. Элементами унимодулярной матрицы может быть только 0, 1, –1.

Теорема 2.2. Если матрица ограничений разрешимой задачи линейного программирования с целыми коэффициентами вполне унимодулярна, то у нее существует целочисленное решение.

Теорема 2.3. Матрица A вполне унимодулярна тогда и только тогда, когда для любого целочисленного вектора \bar{b} все вершины многогранника

$$\begin{cases} A\bar{x} \leq \bar{b}; \\ \bar{x} \geq 0 \end{cases}$$

являются целочисленными.

Полные доказательства теорем 2.2, 2.3 приводятся в [7], [8].

Очевидно, что матрица системы ограничений рассматриваемой задачи вполне унимодулярна, так как состоит только из элементов, равных 0 или 1. Тогда, согласно теореме 2.3, координаты всех вершин симплекса в этом случае целые. Поэтому в качестве метода решения рассматриваемой задачи целесообразно выбрать именно симплекс-метод, как наиболее простой и надежный.

Как известно, стандартная задача линейного программирования формулируется следующим образом:

найти

$$\max \left\{ J(z_r) = \sum_{r=1}^R \gamma_r z_r \right\}, \quad (2.25)$$

при ограничениях:

$$\sum_{r=1}^R b_{rq} z_r \leq a_q, \quad q=1, \dots, Q, \quad Q < R, \quad z_r \geq 0. \quad (2.26)$$

Очевидно, в нашем случае коэффициенты b_{rq} равны либо 0, либо 1. Причем $R = nm + nml$, $Q = 2m + n + l$.

Установим взаимно-однозначное соответствия x_{ij} и y_{kij} в z_r и наоборот, что совершенно эквивалентно преобразованию индексов i, j (для x_{ij}) и i, j, k (для y_{kij}) исходной задачи в индекс r стандартной задачи линейного программирования.

Для нахождения индексов i, j (для x_{ij}) воспользуемся правилом преобразования в десятичное число двухразрядного кода, первый разряд которого содержит значения от 1 до n , а второй от 1 до m . Тогда

$$(r-1) = (j-1) + (i-1)n, \quad j=1, \dots, m; \quad i=1, \dots, n. \quad (2.27)$$

Очевидно, в зависимости от значений i и j индекс r будет изменяться от 1 до nm .

Обратное соотношение будет иметь вид:

$$\begin{cases} (i-1) = \left\lfloor \frac{r-1}{n} \right\rfloor; \\ (j-1) = (r-1) - \left\lfloor \frac{r-1}{n} \right\rfloor n; \end{cases} \quad r=1, \dots, nm \quad (2.28)$$

Для нахождения индексов k, i, j (для y_{kij}) воспользуемся правилом преобразования в десятичное число трехразрядного кода, первый разряд которого принимает значения от 1 до l , второй от 1 до m , третий от 1 до n . Так как первые nm значений индекса r уже использованы, то для вычисления r будем использовать формулу:

$$\begin{aligned} (k(r-1) - 1) + (j-1)l + (i-1)ln + mn; \\ k=1, \dots, l; \quad i=1, \dots, m; \quad j=1, \dots, n. \end{aligned} \quad (2.29)$$

Согласно формуле (2.29) индекс r изменяется от $(nm + 1)$ до $(nm + nml)$. Обратное отображение индекса r в индексы i, j, k (для y_{kij}) будет иметь вид:

$$\begin{cases} (i-1) = \left[\frac{r-1}{nl} \right]; \\ (j-1) = \left[\frac{(r-1) - \left[\frac{r-1}{nl} \right] nl}{l} \right]; \\ (k-1) = (r-1) - \left[\frac{(r-1) - \left[\frac{r-1}{nl} \right] nl}{l} \right] l; \end{cases} \quad r = nm + 1, \dots, nm + nml. \quad (2.30)$$

Теперь можем записать формулу соответствия коэффициентов γ_r в критерии (2.25) и коэффициентов c_{ij} и c_{kij} в критерии (2.23):

$$\gamma_r = \begin{cases} c_{ij}, & i, j \text{ вычисляются по формулам (2.28),} \\ & \text{для } 1 \leq r \leq nm; \\ c_{kij}, & i, j, k \text{ вычисляются по формулам (2.30),} \\ & \text{для } (nm + 1) \leq r \leq (nm + nml). \end{cases} \quad (2.31)$$

Аналогично получаем формулу соответствия свободных членов a_q в ограничениях (2.26) и свободных членов μ_i, α_i, β_k в ограничениях (2.24):

$$a_q = \begin{cases} \mu_i, & i = q, \quad 1 \leq q \leq m; \\ 1, & m + 1 \leq q \leq m + n; \\ \alpha_i, & i = q - m - n, \quad m + n + 1 \leq q \leq 2m + n; \\ \beta_k, & k = q - 2m + n, \quad 2m + n + 1 \leq q \leq 2m + n + l. \end{cases} \quad (2.32)$$

Как уже отмечалось, коэффициенты b_{rq} в ограничениях (2.26) имеют значения либо 0, либо 1. Учитывая это, формулу для определения b_{rq} в зависимости от q можно представить в виде:

1) для $1 \leq q \leq m$

$$b_{rq} = \begin{cases} 1, & \text{если } [(i=q) \cap (r \leq nm)], \quad i = \left[\frac{r-1}{n} \right] + 1; \\ 0, & \text{в остальных случаях;} \end{cases} \quad (2.33)$$

2) для $m+1 \leq q \leq n+m$

$$b_{rq} = \begin{cases} 1, & \text{если } [(i=q-m) \cap (r \leq nm)], \quad i \text{ вычисляется по формулам (2.28);} \\ 1, & \text{если } [(i=q-m) \cap (r \geq nm+1)], \quad i \text{ вычисляется по формулам (2.30);} \\ 0, & \text{в остальных случаях;} \end{cases} \quad (2.34)$$

3) для $n+m+1 \leq q \leq 2m+n$

$$b_{rq} = \begin{cases} \begin{cases} 1, & \text{если } k \in Q_i; \\ 0, & \text{если } k \notin Q_i; \end{cases} & \text{если } [(r \geq nm+1) \cap (i=q-m-n)], \\ i, \quad j, \quad k & \text{вычисляется по формулам (2.28);} \\ 0, & \text{в остальных случаях;} \end{cases} \quad (2.35)$$

4) для $2m+n+1 \leq q \leq 2m+n+l$

$$b_{rq} = \begin{cases} \begin{cases} 1, & \text{если } i \in P_k; \\ 0, & \text{если } i \notin P_k; \end{cases} & \text{если } [(r \geq nm+1) \cap (i=q-2m-n)], \\ i, \quad j, \quad k & \text{вычисляются по формулам (2.30);} \\ 0, & \text{в остальных случаях.} \end{cases} \quad (2.36)$$

Таким образом, получаем стандартную задачу линейного программирования:

найти

$$\max \left\{ F(z_r) = \sum_{r=1}^{mn+mnl} \gamma_r z_r \right\} \quad (2.37)$$

при ограничениях:

$$\sum_{r=1}^{mn+mnl} b_{rq} z_r \leq a_q, \quad 1 \leq q \leq 2m + n + l. \quad (2.38)$$

Причем, как уже отмечалось, b_{rq} равны либо 0, либо 1 и определяются по формулам (2.33) – (2.36), а γ_r и a_q по формулам (2.31) и (2.32) соответственно.

Для решения этой задачи используется симплекс-метод, который, как уже упоминалось, при целочисленных значениях a_q и b_{rq} , равных 0 или 1, обеспечивает целочисленное решение z^* . После определения оптимального вектора z^* легко найти соответствующие значения x_{ij}^* и y_{kij}^* , где индексы i, j и k вычисляются по формулам (2.29) и (2.30) для всех значений r из отрезка $[1, mn + mn l]$.

В заключении данного раздела необходимо отметить, что решение поставленной выше задачи базируется на решении двух подзадач:

определение минимального времени перехвата j -й цели с i -го аэродрома-типа τ_{ij} ;

определение подмножеств Q_i ($i = 1, \dots, m$) и P_k ($k = 1, \dots, l$).

Методы решения этих задач относятся к нелинейному программированию.

2.2.3. Тестовый пример задачи оптимального перегруппирования средств истребительной авиации

Для иллюстрации работоспособности описанного выше подхода к задаче оптимального перегруппирования средств ИА рассмотрен гипотетический налет из 126 воздушных объектов противника. В табл. 2.1 представлены характеристики основных, а в табл. 2.2 резервных аэродромов-типов в плане их технической оснащенности.

Таблица 2.1

Название аэродрома	Тип ЛА	Количество ЛА	Вместимость
Аэродром 1.1	Тип 1	8	0
Аэродром 1.1	Тип 3	8	56
Аэродром 1.2	Тип 3	12	60
Аэродром 1.3	Тип 2	6	66

Таблица 2.2

Название аэродрома	Тип ЛА	Количество ЛА
Аэродром 2.1	Тип 2	36
Аэродром 2.2	Тип 3	24
Аэродром 2.2	Тип 1	24

Резерв времени между моментом фиксации налета противника и моментом вылета истребителей с основных аэродромов-типов составляет 2 ч 00 мин. За это время любой истребитель с любого резервного аэродрома имеет возможность перебазироваться на любой основной аэродром.

При решении задачи целераспределения по имеющемуся гипотетическому налету целей, принимая во внимание только основные аэродромы, получим, что только на 34 цели (из 126 целей, осуществляющих налет) будут назначены средства воздействия ИА.

Если принять в рассмотрение как основные, так и резервные аэродромы, то в результате получим оптимальное перегруппирование средств ИА (табл. 2.3) и уже на 118 (из 126) целей будут назначены средства воздействия ИА.

Таблица 2.3

№	Резервный аэродром	Тип ЛА	Количество ЛА	Основной аэродром
1	Аэродром 2.1	Тип 2	36	Аэродром 1.3
2	Аэродром 2.2	Тип 1	24	Аэродром 1.2
3	Аэродром 2.2	Тип 3	24	Аэродром 1.2

Необходимо отметить, что возможны и другие варианты перегруппирования, при которых число потенциально уничтоженных целей будет таким же, но, именно, этот способ расчета гарантирует минимальное суммарное углубление целей.

2.3. Оптимальное распределение запасов реактивности при работе системы ядерных реакторов в переменном суточном графике нагрузки

2.3.1. Физическая постановка задачи

В условиях широкого ввода ядерно-энергетических мощностей и растущего разуплотнения графиков нагрузок энергосистем использование атомных электростанций (АЭС) в базовой части графика нагрузок будет представлять все большие трудности. Поэтому часть электростанций вынуждена будет работать в соответствии с суточными и сезонными колебаниями потребности в электроэнергии.

При этом решение проблемы маневренности энергоблоков АЭС возможно только после проведения комплекса научно-исследовательских и опытно-конструкторских работ. Основные из них следующие:

- 1) снятие ограничений на число циклов пуск-остановка для всего оборудования АЭС;
- 2) улучшение схем пусков и остановок АЭС и снижение потерь тепла при расхолаживании блоков за счет рационального использования остаточного энерговыделения ядерного топлива и теплоаккумулирующей способности графитовой кладки;
- 3) решение проблемы выбора оптимального типа топлива, защитных оболочек для него и конструкций тепловыделяющих элементов с учетом работы реакторов при переменных нагрузках;
- 4) приведение в соответствие с требованиями эксплуатации систем управления и защиты реакторов и обеспечение реакторов необходимым запасом реактивности.

Первые три проблемы носят, в основном, теплотехнический и прочностной характер, в то время как последняя связана с физикой ядерного реактора и обусловлена нестационарным отравлением реактора ксеноном.

Обеспечение реактора оперативным запасом реактивности, позволяющим компенсировать нестационарное отравление ксеноном, приводит к снижению энерговыработки реактора. Потерю энерговыработки реактора, работающего в переменном графике нагрузки, по сравнению с энерговыработкой при работе на

номинальной мощности можно связать с резервируемым запасом реактивности соотношением

$$\Delta Q = \frac{\Delta \rho}{a}, \quad (2.39)$$

где a – темп выгорания, $1/(\text{кВт} \cdot \text{сут})$; $\Delta \rho$ – запас реактивности, отн. ед.; $\Delta Q = Q_m - Q$ – разность между энерговыработкой при работе в базисном режиме (Q_m) и режиме переменных нагрузок (Q), $\text{кВт} \cdot \text{сут}$.

Величина резервируемого запаса реактивности, а следовательно, и потеря энерговыработки реактора зависят от требуемой длительности работы реактора на пониженной мощности; W от степени снижения мощности реактора $\varepsilon = \frac{W}{W_n}$, где W_n – номинальная мощность реактора; от величины плотности потока нейтронов при работе на номинальной мощности и других характеристик реактора.

В рамках точечной модели запас реактивности, обеспечивающий работу реактора на пониженной мощности εW_n в течение произвольного времени после снижения мощности, равен

$$\Delta \rho = x_m - x_p, \quad (2.40)$$

где x_m и x_p – максимальная и равновесная концентрации ксенона,

нормированные на $\frac{\nu_f \Sigma_f}{\sigma_x}$; Σ_f – макроскопическое сечение

деления активной зоны реактора; ν_f – среднее число вторичных нейтронов на акт деления; σ_x – сечение поглощения ксенона, см^2 .

На АЭС, как правило, устанавливается несколько энергоблоков. В общем случае энергоблоки могут отличаться электрической мощностью, темпом выгорания, другими характеристиками и работать независимо друг от друга. Условием, связывающим реакторы, входящие в состав АЭС, является выработка заданного количества электроэнергии атомной электростанцией.

При ограниченном оперативном запасе реактивности, например, для реакторов с непрерывной перегрузкой или в конце кампании для реакторов с дискретной перегрузкой возникает задача оптимизации

распределения запасов реактивности в системе реакторов с целью минимизации суммарной потери энерговыработки при удовлетворении заданному графику суточного снижения мощности.

Математически задача формулируется следующим образом:
найти

$$\min_{\Delta\rho_1 \dots \Delta\rho_N} \left\{ \sum_{i=1}^N \frac{\Delta\rho_i}{a_i} \right\}$$

при ограничениях:

$$\begin{aligned} \sum_{i=1}^N \delta_i \epsilon_i (\Delta\rho_i) &= \alpha; \\ 0 \leq \Delta\rho_i &\leq \Delta\rho_{im}, i = 1, \dots, N, \end{aligned} \quad (2.41)$$

где N — число реакторов на станции; $\Delta\rho_i$ — оперативный запас реактивности i -го реактора; ϵ_i — степень снижения мощности i -го реактора; α — заданная степень снижения мощности АЭС, причем

$$\alpha = \frac{\sum_{i=1}^N W_i}{\sum_{i=1}^N W_{in}};$$

δ_i — доля электрической мощности i -го реактора,

$$\delta_i = \frac{W_{in}}{\sum_{i=1}^N W_{in}}.$$

Заметим, что оптимизация возможна только при $0 < \alpha < 1$, так как при $\alpha = 0$ и $\alpha = 1$ значения оперативных запасов реактивности определены:

$$\begin{aligned} \Delta\rho_i &= \Delta\rho_{im}, \\ \Delta\rho_i &= 0. \end{aligned}$$

Поэтому качественно понятно, что эффект от оптимизации будет сильнее всего сказываться в средней части диапазона изменения α . Вид зависимости $\epsilon(\Delta\rho)$ возможной степени сниже-

ния мощности реактора от запаса реактивности, вообще говоря, определяется режимом изменения мощности реактора.

Наиболее простым режимом для анализа является снижение мощности реактора с максимально возможной скоростью до определенного уровня и поддержание реактора на данном уровне до момента выхода АЭС на номинальную мощность. Характер зависимости $\varepsilon(\Delta\rho)$ при различных плотностях потоков нейтронов этом случае показан на рис. 2.2.

Как видно из рисунка, при малых плотностях потока нейтронов зависимость $\varepsilon(\Delta\rho)$ представляет собой практически линейную функцию. С увеличением плотности потока нейтронов нелинейность в характере зависимости $\varepsilon(\Delta\rho)$ увеличивается.

В данном разделе рассмотрено решение задачи для системы реакторов с линейной зависимостью возможной степени снижения мощности от запаса реактивности.

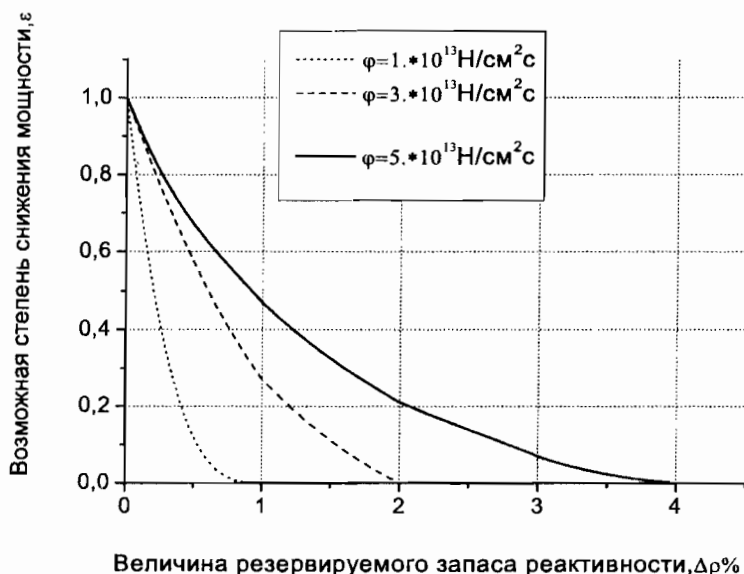


Рис. 2.2. Зависимость возможной степени снижения мощности от величины резервируемого запаса реактивности

2.3.2. Оптимальное распределение запасов реактивности в системе реакторов с линейной зависимостью степени снижения мощности от запаса реактивности

В этом случае математическая постановка задачи такова:
найти

$$\min_{\Delta \rho_i} \left\{ \sum_{i=1}^N \frac{\Delta \rho_i}{a_i} \right\}$$

при ограничениях:

$$\begin{aligned} \sum_{i=1}^N \delta_i \varepsilon_i (\Delta \rho_i) &= \alpha; \\ 0 \leq \Delta \rho_i &\leq \Delta \rho_{im}, \quad i = 1, \dots, N, \end{aligned} \quad (2.42)$$

где $\varepsilon_i = 1 - K_i \Delta \rho_i$.

Как видно из соотношения (2.42), оптимизационная задача относится к классу задач линейного программирования и может быть решена симплекс-методам (см. разд. 1.2). Однако вследствие того, что задача содержит всего одно уравнение связи между переменными, она может быть решена для произвольного числа реакторов, входящих в систему, достаточно просто без применения ЭВМ.

Решение задачи. Графическая интерпретация. Введем для удобства новые переменные

$$z_i = \frac{\Delta \rho_i}{\Delta \rho_{im}}, \quad i = 1, \dots, N,$$

имеющие смысл относительных запасов реактивности. Примем во внимание, что в широком диапазоне изменения плотностей потоков нейтронов $\left(\varphi > 6 \div 7 \cdot 10^{12} \frac{\text{нейтр.}}{\text{см}^2 \cdot \text{с}} \right)$ зависимость $\Delta \rho_m(\varphi)$ линейна,

т.е. $\Delta \rho_m = b \cdot \varphi$, где $b = \text{const}$.

Запишем оптимизационную задачу (2.42) для новых переменных:

найти

$$\min_{z_1 \dots z_N} \sum_{i=1}^N \frac{\varphi_i}{a_i} z_i$$

при ограничениях:

$$\begin{aligned}\sum_{i=1}^N \delta_i (1 - z_i) &= \alpha; \\ 0 \leq z_i &\leq 1 \quad i = 1, \dots, N.\end{aligned}\tag{2.43}$$

Учитывая, что $\sum_{i=1}^N \delta_i = 1$, и обозначив

$$\begin{aligned}y_i &= \delta_i z_i, \\ \mu_i &= \frac{\varphi_i}{\delta_i a_i},\end{aligned}$$

приведем оптимизационную задачу (2.43) к стандартному виду задач линейного программирования:

найти

$$\min_{z_1 \dots z_N} \sum_{i=1}^N \mu_i y_i$$

при ограничениях:

$$\begin{aligned}\sum_{i=1}^N \delta_i (1 - z_i) &= \alpha; \\ 0 \leq z_i &\leq 1 \quad i = 1, \dots, N.\end{aligned}\tag{2.44}$$

Перенумеруем реакторы, входящие в систему, в порядке возрастания коэффициентов μ_i .

Пусть $1 - \alpha \leq \delta_1$, тогда ясно, что сумма $S = \sum_{i=1}^N \mu_i y_i$ минимальна,

если

$$y_1 = 1 - \alpha, \quad y_2 = \dots = y_i = \dots = y_N = 0.$$

Действительно, если распределение $\{y_i\}$ имеет любой другой вид, например

$$y_1 = \gamma_1, \quad y_2 = \gamma_2, \quad \dots, \quad y_N = 1 - \alpha - \sum_{i=1}^{N-1} \gamma_i, \quad \text{причем} \quad \gamma_1 = 1 - \alpha,$$

то минимизируемая сумма $\sum_{i=1}^N \mu_i y_i$ имеет величину

$$S' = \mu_1 \gamma_1 + \mu_2 \gamma_2 + \dots + \mu_N \left(1 - \alpha - \sum_{i=1}^{N-1} \gamma_i \right),$$

при оптимальном же распределении

$$S' = \mu_1 \gamma_1 + \mu_1 \gamma_2 + \dots + \mu_1 \left(1 - \alpha - \sum_{i=1}^{N-1} \gamma_i \right).$$

Так как $\mu_1 \leq \mu_2 \leq \dots \leq \mu_N$, то очевидно, что $S < S'$.

Рассуждая аналогичным образом для случаев

$$\delta_1 < 1 - \alpha \leq \delta_1 + \delta_2, \quad \delta_1 + \delta_2 < 1 - \alpha \leq \delta_1 + \delta_2 + \delta_3, \quad \dots,$$

$$\sum_{i=1}^{N-1} \delta_i < 1 - \alpha \leq \sum_{i=1}^N \delta_i = 1,$$

можно получить следующее решение задачи (2.44):

$$y_1 = 1 - \alpha, \quad y_2 = \dots = y_i = \dots = y_N = 0 \quad \text{при} \quad 0 < 1 - \alpha \leq \delta_1;$$

$$y_1 = \delta_1, \quad y_2 = 1 - \alpha - \delta_1, \quad y_3 = \dots = y_N = 0 \quad \text{при} \quad \delta_1 < 1 - \alpha \leq \delta_1 + \delta_2;$$

$$y_1 = \delta_1, \quad y_2 = \delta_2, \quad y_{N-1} = \delta_{N-1}, \quad y_N = 1 - \alpha - \sum_{i=1}^{N-1} \delta_i$$

$$\text{при} \quad \sum_{i=1}^{N-1} \delta_i < 1 - \alpha \leq \sum_{i=1}^N \delta_i = 1.$$

Возвращаясь к прежним переменным $z_i = y_i / \delta_i$, получим решение исходной оптимизационной задачи (2.43):

$$z_1 = \frac{1 - \alpha}{\delta_1}, \quad z_2 = \dots = z_i = \dots = z_N = 0 \quad \text{при} \quad 1 - \delta_1 \leq \alpha < 1;$$

$$z_1 = 1, \quad z_2 = \frac{1 - \alpha - \delta_1}{\delta_2}, \quad z_3 = \dots = z_N = 0 \quad \text{при} \quad 1 - \delta_1 - \delta_2 \leq \alpha < 1;$$

$$z_1 = z_2 = \dots = z_{N-1} = 1, \quad z_N = \frac{1 - \alpha - \sum_{i=1}^{N-1} \delta_i}{\delta_N} \quad \text{при} \quad 0 < \alpha < 1 - \sum_{i=1}^{N-1} \delta_i. \quad (2.45)$$

Оптимальное распределение относительных запасов реактивности при заданной степени снижения мощности системы

α является координатами точки в N -мерном пространстве изменения переменных z_1, \dots, z_N . Совокупность точек для $0 < \alpha < 1$ представляет собой оптимальную траекторию распределения запасов реактивности. Оптимальные траектории находятся на границе области изменения переменных.

В качестве примера рассмотрим систему, состоящую из двух реакторов. Область изменения переменных представляет собой квадрат *ODEL* (рис. 2.3).



Рис. 2.3. Траектории оптимальных распределений относительных запасов реактивности в системе двух реакторов с низким потоком (линейная зависимость $\epsilon(\Delta\rho)$)

Степень отличия реактора будем характеризовать величиной

$$F = \frac{\mu_1}{\mu_2} = \frac{\delta_1 a_1}{\varphi_1} \bigg/ \frac{\delta_2 a_2}{\varphi_2},$$

в дальнейшем называемой параметром системы.

Если параметр системы больше единицы, то точка, координаты которой являются решением оптимизационной задачи, лежит на стороне OD при $1 - \delta_1 \leq \alpha < 1$ и на стороне DE при $0 < \alpha \leq 1 - \delta_1$. Оптимальной траекторией запасов реактивности является ломаная ODE . Если параметр системы меньше единицы, то точка с оптимальными координатами лежит на стороне OL при $1 - \delta_2 \leq \alpha < 1$ и на стороне LE при $0 < \alpha \leq 1 - \delta_2$. Оптимальной траекторией является ломаная OLE .

При параметре системы, равном единице, целевая функция от распределения относительных запасов реактивности не зависит и координаты любой точки, принадлежащей квадрату $ODEL$, являются решением задачи.

Физическая интерпретация оптимального распределения запаса реактивности. Запас реактивности, в первую очередь, необходимо резервировать в реакторе с большой величиной комплекса $\delta a/\phi$. Если заданное снижение мощности станции можно обеспечить только за счет реактора с максимально величиной $\delta a/\phi$, то в остальных реакторах запас реактивности на работу в переменном режиме не резервируется. В противном случае, в реакторе с максимальным значением комплекса $\delta a/\phi$ резервируется максимальный запас реактивности, а в следующем по величине комплекса $\delta a/\phi$ реакторе — запас реактивности, необходимый для обеспечения заданного снижения мощности АЭС и т.д. На рис. 2.4 показаны траектории оптимальных степеней снижения мощностей для системы двух низкопоточных реакторов, соответствующие оптимальному распределению запасов реактивности.

Как видно из рис. 2.4, при параметре системы $F > 1$ первый реактор работает в полупиковом режиме, вплоть до полной остановки (если $\alpha < 1 - \delta_1$), второй реактор работает в базисном режиме. При $F < 1$ характер изменения степеней снижения мощности реакторов противоположен.



Рис. 2.4. Траектории оптимальных степеней снижения мощности в системе двух реакторов с низким потоком (линейная зависимость $\varepsilon(\Delta p)$)

Максимально возможный эффект оптимизации. Об эффективности оптимизации будем судить по величине:

$$\Delta S(\alpha) = \frac{S_{\max} - S_{\min}}{\sum_{i=1}^N \frac{\varphi_i}{a_i}},$$

где S_{\max} и S_{\min} – максимальная и минимальная потеря энерговыработки АЭС при степени снижения мощности α ;

$\sum_{i=1}^N \frac{\varphi_i}{a_i}$ – потеря энерговыработки АЭС при резервировании запаса

реактивности на полную остановку реакторов.

По своему физическому смыслу величина ΔS характеризует максимально возможный проигрыш в энерговыработке системы реакторов от пренебрежения оптимизацией и дает возможность судить о целесообразности оптимизации данной системы. На рис. 2.5, 2.6 показана зависимость максимально возможного эффекта оптимизации от степени снижения мощности АЭС для

систем из двух низкопоточных реакторов. Различные системы реакторов характеризуются величиной параметра $F = \frac{\delta_1 a_1}{\varphi_1} / \frac{\delta_2 a_2}{\varphi_2}$.

На рисунках отражено, соответственно, два случая: доли мощности реакторов одинаковы ($\delta_1 = \delta_2 = 0,5$), доли мощности реакторов отличаются в два раза ($\frac{\delta_1}{\delta_2} = 2$).

Таким образом, в системе реакторов с линейной зависимостью степени снижения мощности реактора от запаса реактивности оптимальное распределение запасов реактивности находится на границе области изменения переменных, т.е. оптимальный режим работы реакторов формируется из полупиковых и базисных режимов работы отдельных реакторов. Эффект оптимизации тем больше, чем большее значение имеет параметр системы.

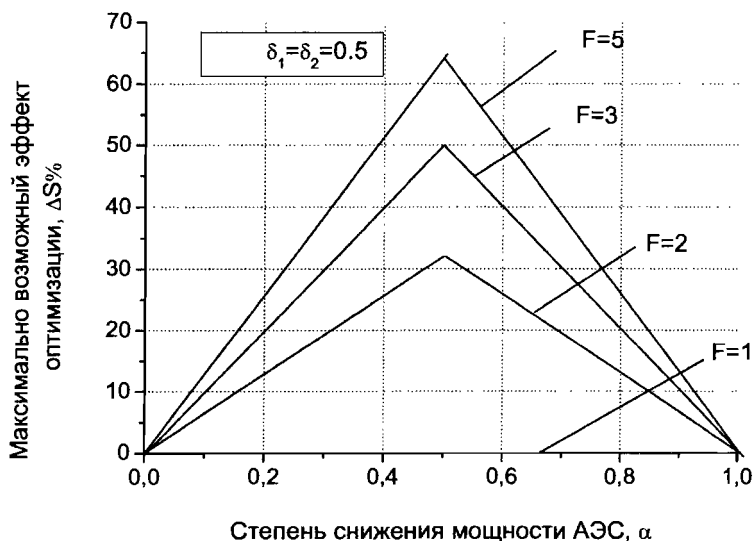


Рис. 2.5. Зависимость величины максимално возможного эффекта оптимизации от степени снижения мощности системы при различных величинах параметра F

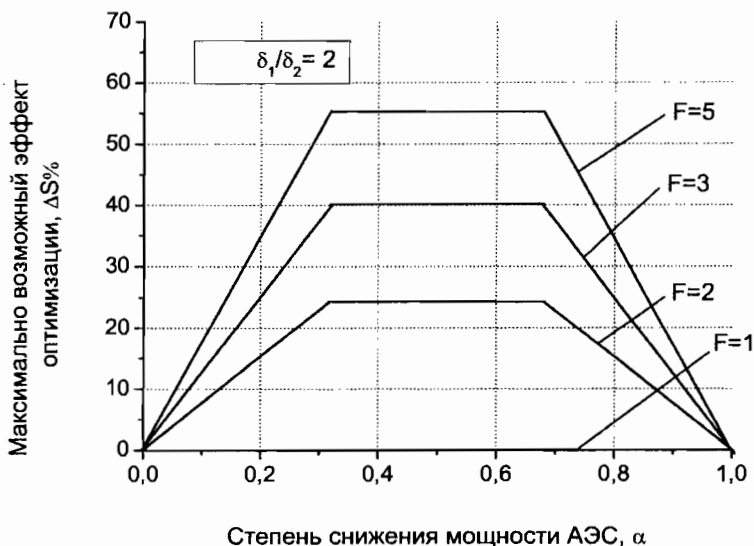


Рис. 2.6. Зависимость величины максимально возможного эффекта оптимизации от степени снижения мощности системы при различных величинах параметра F

2.4. Оптимизация размещения радиолокационных станций

2.4.1. Модели радиолокационных станций и основные этапы решения задачи

Задачу оптимального размещения радиолокационных станций (РЛС) [12] можно сформулировать в следующем виде: при заданной трубке траекторий целей найти такое множество точек размещения РЛС в заданной области, чтобы их количество было минимальным, а обеспечиваемая эффективность пеленга – максимальной.

Трубка траекторий может быть разбита на элементарные, причем в пределах каждой элементарной трубки все траектории считаются совпадающими с точки зрения пеленга целей. О каждой элементарной трубке будем судить по единственной принадлежащей ей пробной траектории.

Тогда эффективность пеленга может быть оценена как отношение количества пеленгуемых пробных целей к общему количеству целей. Задача сводится к отысканию минимума с ограничениями в бу-

левых переменных (соответствующих возможным точкам размещения и принимающих значение 1, если в соответствующей точке присутствует РЛС, и 0 в противном). Целевая функция при этом может выражать непосредственно количество РЛС; ограничения возникают из потребности обеспечить пеленг всех целей, траектории которых эффективно пересекают область действия РЛС. Сформулированная задача относится к классу задач линейного программирования.

Блок-схема алгоритма решения задачи приведена на рис. 2.7.

В данном разделе используются следующие модели РЛС (или области действия) [13, 14].

1. *Интерференционная модель РЛС.* Такая модель учитывает отражение электромагнитных волн от поверхности Земли (без учета рельефа). Область действия такой РЛС описывается непрерывной функцией $r = r(\vartheta)$, имеющей вид

$$r(\vartheta) = R_{\max} f(\vartheta) \sqrt{1 + \rho - 2\rho \cos\left(\vartheta + \frac{2\pi\delta}{\lambda}\right)}, \quad (2.46)$$

где R_{\max} — максимальная дальность действия без учета ослабления; $f(\vartheta)$ — диаграмма направленности антенны; ρ — эффективный коэффициент отражения; λ — рабочая длина волны РЛС.

Такая область действия невыпукла и имеет «лепестковую» структуру (рис. 2.8).

2. *Цилиндрическая модель.* Область действия такой РЛС — выпуклая, представляющая собой прямой круговой цилиндр, центр основания которого находится в точке размещения РЛС (рис. 2.9).

3. *Кусочно-линейная модель.* Область действия представляет собой поверхность вращения, каждое сечение которой представляется кусочно-линейной функцией (рис. 2.10).



Рис. 2.7. Блок-схема алгоритма оптимального размещения РЛС

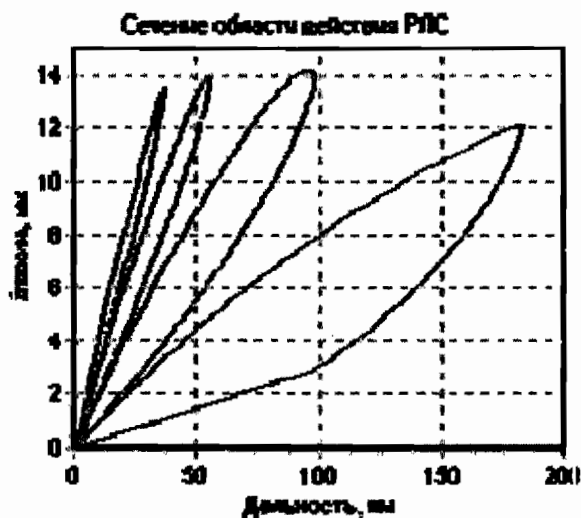


Рис. 2.8. Интерференционная модель области действия РЛС

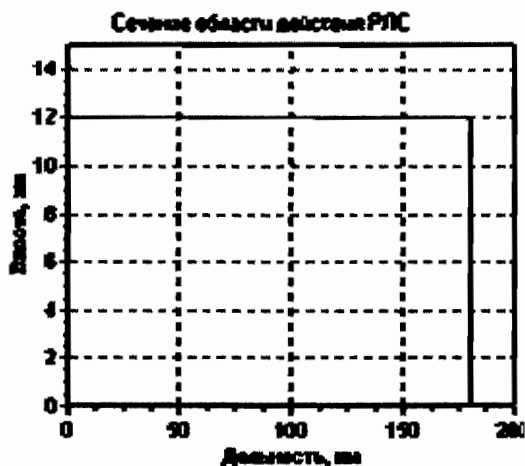


Рис. 2.9. Цилиндрическая модель области действия РЛС

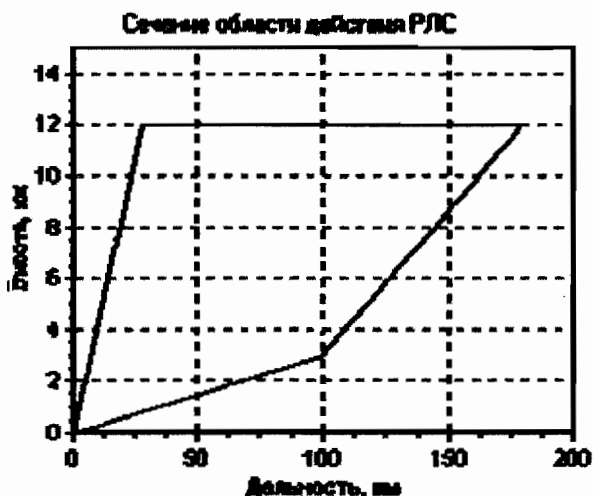


Рис. 2.10. Кусочно-линейная модель области действия РЛС

Очевидно, цилиндрическая модель является простейшей, а интерференционная — наиболее точной.

Для использования такой модели в алгоритме определения пересечений траектории с областью действия РЛС требуется способ вычисления дальности действия станции $r(\vartheta)$ в направлении ϑ .

2.4.2. Используемые системы координат

Большинство вычислений выполнено в геоцентрической системе координат.

Геоцентрическая система координат построена следующим образом (рис. 2.11). Начало координат находится в центре эллипсоида. Ось Ox направлена по вектору угловой скорости вращения Земли; ось Oy проходит через гринвичский меридиан, ось Oz дополняет систему координат до правой. Сферические координаты определяются двумя углами: долготой и широтой точки.

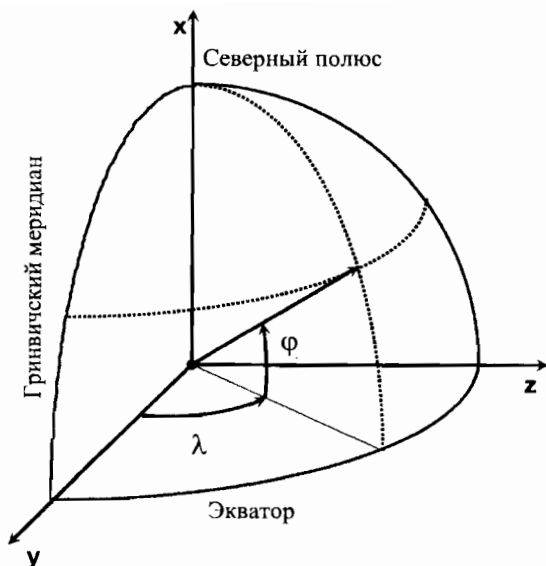


Рис. 2.11. Неинерциальная геоцентрическая система координат

Долгота λ – угол между осью Oy и плоскостью меридиана, проходящего через заданную точку, отсчитывается от гринвичского меридиана и принимает значения в пределах $\lambda \in [-180^\circ; 180^\circ]$.

Геоцентрическая широта φ_c – угол между радиусом-вектором и плоскостью экватора. Отсчитывается от плоскости экватора и принимает значения $\varphi_c \in [-90^\circ; 90^\circ]$.

Соотношение между декартовыми и сферическими координатами при этом устанавливается следующим образом:

$$\begin{cases} x = r \sin \varphi_c; \\ y = r \cos \varphi_c \cos \lambda; \\ z = r \cos \varphi_c \sin \lambda. \end{cases} \quad (2.47)$$

Помимо геоцентрической системы в примере используются другие системы координат, связанные с траекториями целей или с точками размещения РЛС. Все эти системы координат преобразуются к геоцентрическим.

Из всех моделей Земли, представленных поверхностями второго порядка, наиболее общей является модель на основании трехосного эллипсоида. К трехосному эллипсоиду, очевидно, могут быть сведены и простейшая сферическая модель, и модель в виде эллипсоида вращения (например, эллипсоид Красовского).

В настоящей работе принята следующая трехосная модель с учетом следующих соглашений: центр эллипсоида помещен в центр экватора (начало геоцентрических координат); оси эллипсоида совпадают с осями геоцентрической системы координат.

В этом случае геоцентрическая система координат будет для эллипсоида канонической, и его уравнение запишется в виде

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \quad (2.48)$$

где a , b и c – полуоси эллипсоида.

Это уравнение может быть представлено в параметрическом виде, причем роль параметров будут выполнять геоцентрическая широта и долгота соответствующей точки:

$$\begin{cases} x = a \sin \varphi_c; \\ y = b \cos \varphi_c \cos \lambda; \\ z = c \cos \varphi_c \sin \lambda. \end{cases} \quad (2.49)$$

(это соотношение устанавливается, например, путем сжатия эллипсоида вдоль оси Ox в a раз, вдоль оси Oy в b раз, вдоль оси Oz в c раз; образом эллипсоида при таком преобразовании будет единичная сфера).

Длина радиуса-вектора точки, таким образом, есть

$$r = \sqrt{a^2 \sin^2 \varphi_c + (b \cos \lambda + c \sin \lambda)^2 \sin^2 \varphi_c}. \quad (2.50)$$

Последнее выражение также представляет собой уравнение поверхности в сферических координатах.

Углы, образованные радиусом-вектором с осями координат, могут быть найдены через соответствующие направляющие косинусы:

$$\begin{cases} \cos \angle(i, r) = \frac{a \sin \varphi_c}{\sqrt{a^2 \sin^2 \varphi_c + (b \cos \lambda + c \sin \lambda)^2 \sin^2 \varphi_c}}; \\ \cos \angle(j, r) = \frac{b \cos \varphi_c \cos \lambda}{\sqrt{a^2 \sin^2 \varphi_c + (b \cos \lambda + c \sin \lambda)^2 \sin^2 \varphi_c}}; \\ \cos \angle(k, r) = \frac{c \cos \varphi_c \sin \lambda}{\sqrt{a^2 \sin^2 \varphi_c + (b \cos \lambda + c \sin \lambda)^2 \sin^2 \varphi_c}}. \end{cases} \quad (2.51)$$

Уравнение касательной плоскости, проходящей через заданную точку (x_0, y_0, z_0) на поверхности эллипсоида, имеет вид

$$\frac{x_0}{a^2} x + \frac{y_0}{b^2} y + \frac{z_0}{c^2} z = 1. \quad (2.52)$$

Соответственно, вектор нормали в точке (x_0, y_0, z_0) есть

$$\mathbf{N} = \left[\frac{x_0}{a^2} \quad \frac{y_0}{b^2} \quad \frac{z_0}{c^2} \right]^T, \quad (2.53)$$

или в сферических координатах

$$\mathbf{N} = \left[\frac{\sin \varphi_0}{a} \quad \frac{\cos \varphi_0 \cos \lambda_0}{b} \quad \frac{\cos \varphi_0 \sin \lambda_0}{c} \right]^T. \quad (2.54)$$

Широко распространены следующие частные случаи рассмотренной модели.

1. *Сферическая модель.* Представляет собой сферу, ограничивающую объем, равный объему Земли. Центр совпадает с центром окружности экватора. Радиус сферы $R = 6\,371\,110$ м.

2. *Эллипсоид Красовского.* Представляет собой эллипсоид вращения, центр которого совпадает с центром земного экватора. Большая полуось (радиус экватора) $a = 6\,378\,245$ м. Сжатие меридионального эллипса $\alpha = (a - b)/a = 1/298.3$, эксцентриситет e

вычисляется из соотношения $\varepsilon^2 = 1 - k^2$, где $k = b/a = 1 - \alpha$ — коэффициент сжатия.

Пусть на прямоугольнике введена следующая декартова система координат: начало координат помещено в центр прямоугольника, ось Ox' направлена вдоль большей стороны прямоугольника, ось Oz' направлена перпендикулярно плоскости прямоугольника, ось Oy' дополняет тройку до правой.

Пусть рассматриваемый прямоугольник касается поверхности эллипсоида своим центром. Пусть азимут середины меньшей стороны прямоугольника из этой точки равен α , длина и ширина прямоугольника — L и H соответственно, а радиус-вектор точки касания имеет вид

$$r = [Rx \sin \varphi_0 \quad Ry \cos \varphi_0 \cos \lambda_0 \quad Rz \cos \varphi_0 \sin \lambda_0]^T. \quad (2.55)$$

Преобразование координат из локальной системы $(Ox'y'z')$ в геоцентрическую определяется соотношением

$$\vec{r}_{\text{гц}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin \lambda_0 & \cos \lambda_0 \\ 0 & -\cos \lambda_0 & \sin \lambda_0 \end{bmatrix} \begin{bmatrix} \cos \varphi_0 & 0 & \sin \varphi_0 \\ 0 & 1 & 0 \\ -\sin \varphi_0 & 0 & \cos \varphi_0 \end{bmatrix} \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \vec{r}' + \begin{bmatrix} Rx \sin \varphi_0 \\ Ry \cos \varphi_0 \cos \lambda_0 \\ Rz \cos \varphi_0 \sin \lambda_0 \end{bmatrix}. \quad (2.56)$$

Проецирование точек осуществляется следующим образом: вычисляется долгота и широта, находится соответствующая им точка поверхности эллипсоида.

В плоском случае траектории атмосферных летательных аппаратов моделировались прямыми, параллельными друг другу и поверхности плоской Земли (горизонтальные траектории). Трубка траекторий в этом случае была прямоугольным параллелепипедом.

Очевидно, к криволинейной поверхности такая модель траекторий неприменима. Рассмотрим аналог прямолинейных горизонтальных траекторий в случае эллипсоидальной поверхности.

Прямая на плоскости является геодезической линией, т.е. отрезок прямой имеет наименьшую длину из всех кривых, соединяющих пару точек. Геодезической линией на сфере будет большая окружность. Трехосный эллипсоид является образом сферы при трех последовательных растяжениях, поэтому геодезической линией на эллипсоиде будет большой эллипс (его центр совпадает с центром эллипсоида).

Будем считать, что «горизонтальные» траектории должны в проекции на поверхность эллипсоида давать большие эллипсы. Тогда собственно траектория высоты H будет получаться из собственной проекции смещением каждой точки этой проекции на расстояние H в направлении вектора нормали.

Рассмотрим построение «горизонтальной» траектории, проходящей над двумя точками на поверхности эллипсоида. Пусть проекция траектории на поверхность эллипсоида начинается в точке \vec{r}_1 и заканчивается в точке \vec{r}_2 . Отобразим эллипсоид на единичную сферу. Матрица преобразования имеет вид

$$H = \begin{bmatrix} R_x & 0 & 0 \\ 0 & R_y & 0 \\ 0 & 0 & R_z \end{bmatrix}^{-1}. \quad (2.57)$$

Образами точек r_1 и r_2 будут точки $r'_1 = H \cdot r_1$ и $r'_2 = H \cdot r_2$.

Рассмотрим декартову систему координат, направление оси Ox которой совпадает с направлением r'_1 , ось Oz перпендикулярна плоскости, в которой лежат векторы r_1 , r_2 , ось Oy дополняет систему до правой.

Орты осей в этом случае выражаются следующим образом (с учетом того, что r'_1 и r'_2 — единичные векторы):

$$i = r'_1, \quad k = \frac{[r'_1, r'_2]}{|[r'_1, r'_2]|}, \quad j = \frac{[r'_1, k]}{|[r'_1, k]|}. \quad (2.58)$$

Проекция траектории при выполнении преобразования H является окружностью единичного радиуса, причем ее параметрическое уравнение в построенной выше системе координат

$$\vec{r}(t) = \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \\ 0 \end{bmatrix}, \quad (2.59)$$

где ω – угол между векторами \vec{r}_1' и \vec{r}_2' , а параметр t принимает значения $[0; 1]$, при этом $r(0) = r_1'$, т.е. соответствует начальной точке траектории, а $r(1) = r_2'$, т.е. соответствует конечной точке траектории.

Матрица перехода от рассмотренных координат к геоцентрическим имеет вид

$$R = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}, \quad (2.60)$$

где i_x, i_y, i_z – проекции орта оси Ox на оси геоцентрической системы координат и т.д.

Следовательно, проекция траектории, проходящей через заданные точки на эллипсоиде, в параметрической форме в геоцентрических координатах запишется как

$$r_{\text{гц}}^{\text{np}}(t) = Rr(t). \quad (2.61)$$

Восстановление траектории по ее проекции осуществляется добавлением к радиусу-вектору точки проекции единичного вектора нормали в текущей точке, умноженного на высоту траектории:

$$r_{\text{гц}}(t) = r_{\text{гц}}^{\text{np}}(t) + hN(r_{\text{гц}}^{\text{np}}(t)). \quad (2.62)$$

2.4.3. Метод решения задачи оптимизации и результаты расчетов

Точки пересечения траектории с поверхностью области действия РЛС находятся из уравнения

$$r_{\text{тр}}(t) = r_{\text{РЛС}}(\alpha, \beta), \quad (2.63)$$

где $r = r_{\text{тр}}(t)$ – параметрическое уравнение траектории в геоцентрических координатах, $t \in [0; 1]$ – параметр; $r = r_{\text{РЛС}}(\alpha, \beta)$ – параметрическое уравнение поверхности, ограничивающей область действия РЛС, в тех же координатах (α и β – параметры).

Метод решения таких уравнений состоит в следующем.

1. Вычисляется расстояние $\rho(t)$ от точки траектории до точки размещения РЛС как функция параметра t .
2. В точке размещения РЛС находится угловая высота точки траектории над горизонтом $\vartheta(t)$ как функция t .
3. Строится функция

$$f(t) = \rho(t) - R_{\text{РЛС}}(\vartheta), \quad (2.64)$$

где $R_{\text{РЛС}}(\vartheta)$ – дальность действия РЛС в направлении ϑ .

4. В точках пересечения траектории постросная функция меняет знак. Входу цели в область действия соответствует смена знака с «+» на «-», выходу – обратная смена знака.

5. Вводится сетка

$$\left\{ t_k : t_k = \frac{1}{\Delta t} k \right\}. \quad (2.65)$$

6. Между узлами сетки, в которых функция $f(t)$ имеет разный знак, значение параметра, соответствующее пересечению, уточняется методом дихотомии или любым другим, предназначенным для нахождения нулей функции на отрезке.

Упорядоченный перебор узлов сетки в направлении движения цели по траектории обеспечивает получение пар точек пересечения, соответствующих входу и выходу.

Пусть в геоцентрических координатах траектория описывается уравнением $r = r_{\text{тр}}(t)$, радиус-вектор точки размещения РЛС в тех же координатах – $r_{\text{РЛС}}$. Тогда вектор, соединяющий точку размещения и точку траектории, есть $r_{\text{тр}}(t) - r_{\text{РЛС}}$, и расстояние $\rho(t) = |r_{\text{тр}}(t) - r_{\text{РЛС}}|$.

Пусть нормаль к поверхности, на которой расположена РЛС, в точке $r_{\text{РЛС}}$ есть $N(r_{\text{РЛС}})$. Тогда угол между направлением на

точку и нормалью может быть вычислен через скалярное произведение векторов:

$$90^\circ - \vartheta = \arccos \frac{(r_{\text{тр}}(t) - r_{\text{РЛС}}) \cdot N(r_{\text{РЛС}})}{|r_{\text{тр}}(t) - r_{\text{РЛС}}| \cdot |N(r_{\text{РЛС}})|}. \quad (2.66)$$

Соответственно, угловая высота цели

$$\vartheta(t) = 90^\circ - \arccos \frac{(r_{\text{тр}}(t) - r_{\text{РЛС}}) \cdot N(r_{\text{РЛС}})}{|r_{\text{тр}}(t) - r_{\text{РЛС}}| \cdot |N(r_{\text{РЛС}})|}. \quad (2.67)$$

Решение задачи линейного программирования проводится методом зондирования решений (см. разд. 1.7.5). Для плоской модели Земли вычисление точек пересечения проведено аналитически (сведено к нахождению пересечения прямой (траектории) и окружности (горизонтального сечения области действия РЛС)), что сделано для уменьшения времени работы алгоритма с плоской моделью.

При численных расчетах сравнивались эффективности пеленга для различных моделей РЛС и различных моделей поверхности Земли.

Были проведены следующие серии расчетов.

1. Сравнение различных моделей РЛС на плоскости.
2. Сравнение различных моделей РЛС на эллипсоиде Красовского.
3. Сравнение плоской модели Земли и эллипсоида Красовского при использовании интерференционной модели РЛС.
4. Сравнение плоской модели Земли и эллипсоида Красовского при использовании кусочно-линейной модели.

Во всех экспериментах параметры трубки траекторий и области размещения были приняты одинаковыми: минимальная высота цели – 5 км; максимальная высота цели – 10 км; максимальная скорость цели в числах Маха – 4М; фронт налета покрывает всю область размещения РЛС; угол налета – 0° ; шаг разбиения трубки траекторий по высоте – 3 км; шаг разбиения трубки траекторий по ширине – 20 км; размеры региона – 800×200 км; расстояние между возможными точками размещения – 100 км. На эллипсоиде Красовского центр региона помещен в точку с геоцентрическими координатами $\varphi = 45^\circ$, $\lambda = 45^\circ$; регион ориентирован таким образом, что его длинная сторона образует с меридианом угол 0° .

Основные результаты расчетов приведены в табл. 2.4 – 2.9. Жирными точками показано оптимальное размещение РЛС. В каждой таблице две модели РЛС, по которым в данном случае оптимизация не проводилась, использованы в тех же точках поверхности Земли, которые являются оптимальными для размещения РЛС с выбранной для оптимизации моделью.

В табл. 2.4 – 2.9 приведены диаграммы размещения результатов оптимизации для моделей РЛС различного типа:

цилиндрической модели РЛС на плоскости (табл.2.4);

цилиндрической модели РЛС на эллипсоиде Красовского (табл. 2.5);

КЛФ-модели РЛС на плоскости (табл 2.6);

кусочно-линейной модели РЛС на эллипсоиде Красовского (табл. 2.7);

интерференционной модели РЛС на плоскости (табл.2.8);

интерференционной модели РЛС на эллипсоиде Красовского (табл 2.9).

Таблица 2.4

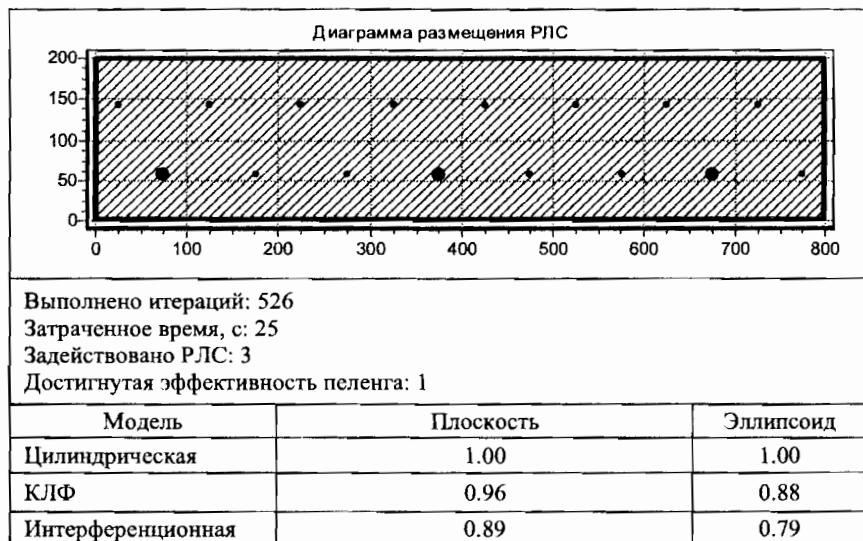


Таблица 2.5

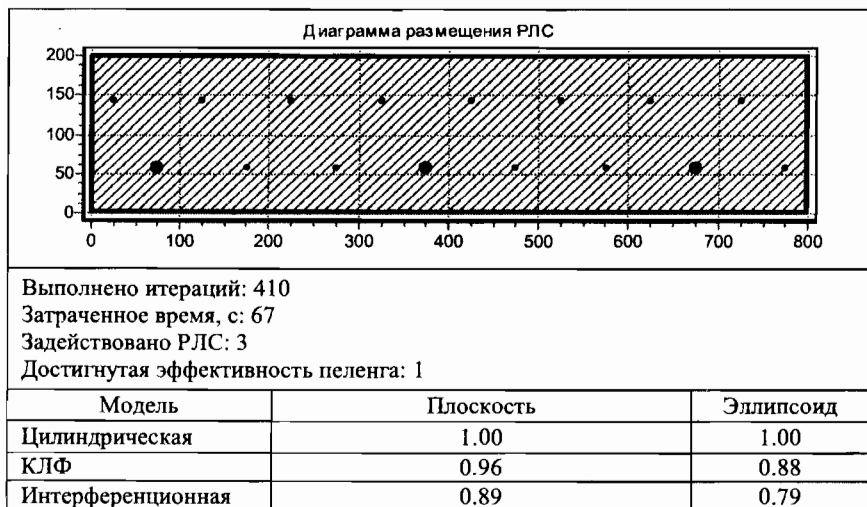


Таблица 2.6

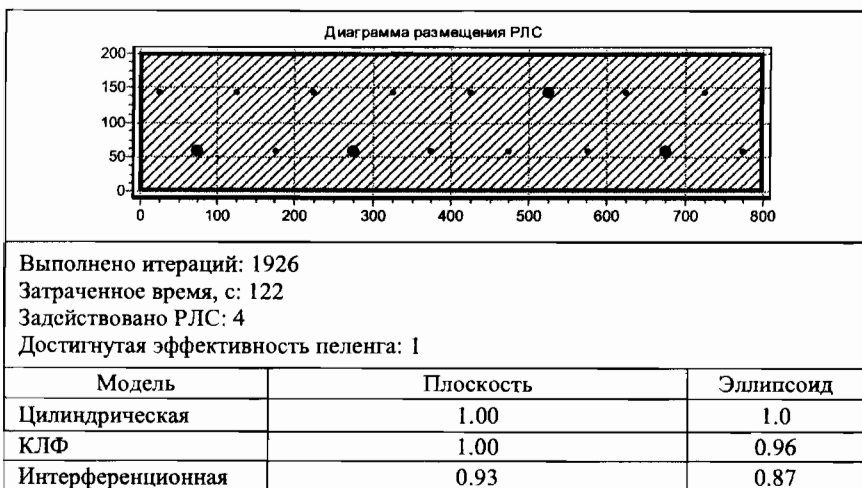
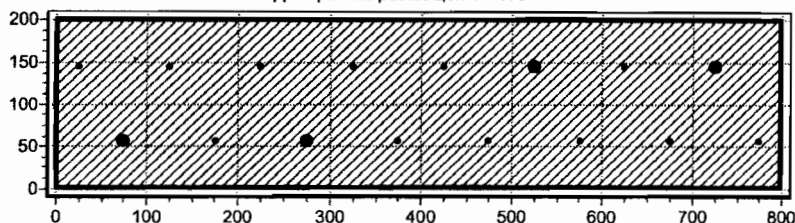


Таблица 2.7

Диаграмма размещения РЛС



Выполнено итераций: 1926

Затраченное время, с: 122

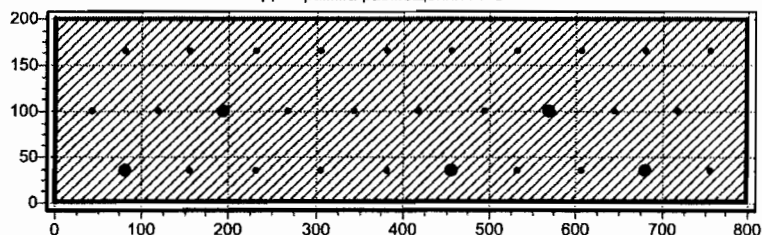
Задействовано РЛС: 4

Достигнутая эффективность пеленга: 1

Модель	Плоскость	Эллипсоид
Цилиндрическая	1.00	1.00
КЛФ	1.00	1.00
Интерференционная	1.00	0.74

Таблица 2.8

Диаграмма размещения РЛС



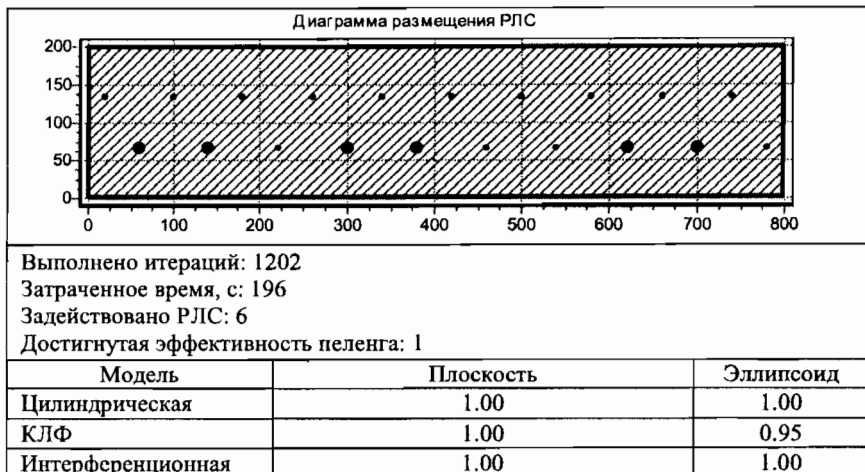
Выполнено итераций: 1305

Затраченное время, с: 146

Задействовано РЛС: 5

Достигнутая эффективность пеленга: 1

Модель	Плоскость	Эллипсоид
Цилиндрическая	1.00	1.00
КЛФ	1.00	1.00
Интерференционная	1.00	0.93



Полученные результаты позволяют сделать вывод, что оптимальное количество РЛС становится наибольшим при невыпуклой (интерференционной) модели РЛС для эллипсоида Красовского. При этом в силу того, что стандартная кусочно-линейная модель области действия РЛС не является в полном смысле огибающей для интерференционной модели, результаты, оптимальные для интерференционной модели, могут оказаться неоптимальными для кусочно-линейной модели.

2.5. Оптимальная кластеризация как задача линейного программирования

2.5.1. Математическая постановка задачи кластерного анализа

Термин «кластерный анализ» предложен Р. Трайном в 1939 г. (cluster – гроздь). Часто в литературе используются как синонимы следующие термины: автоматическая классификация, таксономия, распознавание образов без обучения, самообучение и т.д. Кластер и класс – суть синонимы. Цепью кластерного анализа является автоматическая (или автоматизированная) классификация (кластеризация) объектов по признаку схожести [15].

Схожесть объектов определяется близостью их векторов дескрипторов (параметров описания) в некотором пространстве параметров.

Примером успешного неавтоматизированного кластерного анализа является классификация растений и животных в биологии и зоологии, таблица Д.И. Менделеева в химии, разделение стран на развитые и развивающиеся в экономике и т.д.

Объекты, подлежащие классификации, называются операционными таксономическими единицами (ОТЕ).

Признаковое пространство (векторов-дескрипторов) называется Π -пространством. Обычно $\dim(\Pi) < K$, где K — количество всех возможных признаков ОТЕ, так как между отдельными признаками возможна корреляция (если Π — ЛВП) и в более общем случае — зависимость (часто некоррелированность).

Если для принадлежности некоторых ОТЕ к одному классу необходимо совпадение всех (или подгруппы) признаков, то такой кластер называется монотетическим или таксоном.

Если в этом нет необходимости, и лишь требуется принадлежность всех, агрегируемых в кластер, ОТЕ к заданной окрестности точки Π -пространства, то такие кластеры называют политетическими.

Если вместо всего множества независимых признаков (количество таких признаков = $\dim(\Pi)$) применяется их часть, тем самым неявно вводится взвешивание признаков (отброшенным приписывается нулевой вес, а оставленным вес). При математической постановке задачи кластерного анализа взвешивание признаков (не только с весами 0, 1) является достаточно распространенным приемом.

Пусть имеется P независимых признаков (дескрипторов) для n ОТЕ. Тогда значение i признака для j объекта будем обозначать x_{ij} ($i=1, \dots, p$; $j=1, \dots, n$). Вектор $x_j = (x_{1j}, \dots, x_{pj})$ и будет вектором признаков для j объекта. Матрица, составленная из векторов-столбцов x_j^T , иногда называется в кластерном анализе матрицей данных.

Если значения признаков не меняются в течение времени кластеризации, то p -мерное Π -пространство есть конечное множество, состоящее из n элементов. В этом случае его как ЛВП над полем P рассматривать нельзя.

Если Π -пространство считать метрическим, то метрики $\rho_l = \rho_s(x_l, x_s)$, являющиеся расстояниями между ОТЕ с номерами l, s ($l, s=1, \dots, n$), могут составить так называемую матрицу расстояний p размерности $n \times n$, симметричную, с нулевой диагональю.

В табл. 2.10 приведены некоторые примеры метрических пространств.

Существуют расстояния, не являющиеся метриками в строгом смысле, например расстояние Джеффриса – Мачуситы:

$$M = \left(\sum_{i=1}^p \left(\sqrt{x_{il}} - \sqrt{x_{is}} \right)^2 \right)^{1/2},$$

или «коэффициент дивергенции»

$$CD = \left[\frac{1}{p} \sum_{i=1}^p \left(\frac{x_{il} - x_{is}}{x_{il} + x_{is}} \right)^2 \right]^{1/2}.$$

Таблица 2.10

Название пространства	Формула для $\rho(x_l, x_s)$
R_2^p	$\rho = \left[\sum_{i=1}^p (x_{il} - x_{is})^d \right]^{1/d}, \quad d \in \mathbb{N}$
R_1^p	$\rho = \sum_{i=1}^p x_{il} - x_{is} $
R_∞^p	$\rho = \sup_{i=1, \dots, p} \{ x_{il} - x_{is} \}$
Пространство Махаланобиса	$(\bar{x}_l - \bar{x}_s) W^{-1} (\bar{x}_l - \bar{x}_s)^T, \quad W - \text{заданная матрица}$

Математическая постановка задачи кластерного анализа выглядит следующим образом: пусть $m \leq n$. Надо на основании данных о П-пространстве разбить множество из n ОТЕ на m подмножеств (кластеров) K_1, \dots, K_m так, чтобы ни одна ОТЕ не принадлежала двум или более кластерам. Обычно $m < n$.

Для того чтобы подчеркнуть, что объекты в кластеры объединяются по признаку сходства, иногда используется понятие функции сходства.

Функцией сходства называется такая неотрицательная вещественная функция $s(x_l, x_s)$, что:

- 1) $0 \leq s(\bar{x}_l, \bar{x}_s) < 1, \quad \bar{x}_l \neq \bar{x}_s$;
- 2) $s(\bar{x}_l, \bar{x}_l) = 1$;
- 3) $s(\bar{x}_l, \bar{x}_s) = s(\bar{x}_s, \bar{x}_l)$; $S(\bar{x}_l, \bar{x}_s) = S_{l,s}$.

Примером кластеризации может служить разбиение n стран по уровню развития, опираясь на три признака ($p = 3$): x_1 – ВВП на душу населения, x_2 – личное потребление на душу населения, x_3 – электроэнергия на душу населения.

Решением задачи кластерного анализа является разбиение, соответствующее множеству из m кластеров, которое формируется на основании некоторого критерия, который может быть минимизируемой целевой функцией, где термин «функция» понимается в самом общем смысле.

Рассмотрим в качестве примера целевой функции (критерия) максимальное внутрикластерное расстояние до центра кластера. Если пространство евклидово, то, считая центром кластера его центр «масс» (при том, что «массы» всех ОТЕ одинаковы) при $p = 1$, получим, что ОТЕ принадлежат одному кластеру, если

$$(x_{il} - x_{\text{цм}})^2 \leq r - \text{фиксированное число, } l = 1, \dots, n, \quad (2.68)$$

$$\text{где } x_{\text{цм}} = \frac{\sum_{l=1}^{\tilde{n}} x_{il}}{\tilde{n}}.$$

Если $m < n$, то необходимо допустить, что есть кластер K_v ($v \in \{1, \dots, m\}$), которому принадлежит более чем одна ОТЕ. Поэтому рассматривается множество всех различных пар ОТЕ, вычисляются их $x_{\text{цм}}$, а затем проверяется условие (2.68) по парам.

Если оно ни для одной пары не выполняется, то задача кластеризации для $m < n$ не имеет решения как для пар, так и для троек и т.д. В случае нахождения такой пары (пар) переходим к рассмотрению троек и т.д., пока не определятся самые крупные кластеры, т.е. такие, которые содержат сразу n ОТЕ или $n - 1$, или $n - 2$ и т.д. В данном примере рассмотрено так называемое «естественное» задание кластеров (так как «центры» выбирались, исходя из свойств самих ОТЕ). Если бы «центры» выбирались априорно, то распределение по кластерам было бы другим.

Значения функции сходства составляют симметричную матрицу сходства

$$S = \begin{pmatrix} 1 & s_{12} & \dots & s_{1n} \\ s_{21} & 1 & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \dots & 1 \end{pmatrix},$$

величину $s_{l,s}$ иногда называют коэффициентом сходства.

Величина

$$S_p = \frac{1}{2} \sum_{l=1}^n \sum_{s=1}^n \rho(x_l, x_s)$$

называется общим рассеянием данной функции расстояния.

Величина $\bar{S}_p = S_p / N_p$, где $N_p = \frac{n^2 - n}{2}$, называется средним рассеянием.

Величина S_p представляет собой сумму n^2 расстояний (элементов матрицы расстояний D), из которых n равны нулю, а $\frac{n^2 - n}{2}$, вообще говоря, различны и не равны. Поэтому \bar{S}_p есть среднее арифметическое ненулевых (вообще говоря) расстояний между неповторяющимися парами ОТЕ.

Матрица S размерности $p \times p$ вида

$$S_x = \sum_{l=1}^n (x_l - \bar{x})(x_l - \bar{x})^T,$$

где $\bar{x} = \frac{1}{n} \sum_{l=1}^n \bar{x}_l$, называется матрицей рассеяния множества ОТЕ.

След матрицы S_x :

$$\text{tr } S_x = \sum_{l=1}^n \sum_{i=1}^p (x_{il} - \bar{x}_i)^2$$

называется статистическим рассеянием множества ОТЕ. Легко показать, что

$$\text{tr } S_x = \frac{1}{n} \sum_{l=1}^n \sum_{\substack{s=1 \\ l < s}}^n \rho^2(\bar{x}_l, \bar{x}_s).$$

Для введения критерия кластеризации (целевой функции) необходимо установить признак сходства внутри кластеров и признак различия между кластерами.

Пусть $\{I_1, \dots, I_{n_1}\}$ и $\{I_{n_1+1}, \dots, I_n\}$ принадлежат различным кластерам K_1 и K_2 соответственно. Тогда

$$\rho_1(K_1, K_2) = \min_{\substack{l=1, \dots, n_1 \\ s=n_1+1, \dots, n}} \rho(\bar{x}_l, \bar{x}_s)$$

называется минимальным локальным расстоянием между кластерами K_1 и K_2 в данном метрическом Π -пространстве, а функция

$$\rho_2 = \min_{\substack{l=1, \dots, n_1 \\ s=n_1+1, \dots, n}} \rho(x_l, x_s)$$

называется максимальным локальным расстоянием;

$$\rho_3 = \sum_{l=1}^{n_1} \sum_{s=n_1+1}^n \rho(x_l, x_s) / n_1(n - n_1)$$

называется средним расстоянием;

$$\rho_4 = \frac{n_1(n - n_1)}{n} (\bar{\bar{x}}' - \bar{\bar{x}}'')^T (\bar{\bar{x}}' - \bar{\bar{x}}''), \quad (2.69)$$

где

$$\bar{\bar{x}}' = \sum_{l=1}^{n_1} x_l / n_1; \quad \bar{\bar{x}}'' = \sum_{l=n_1+1}^n x_l / (n - n_1),$$

называется статистическим расстоянием между кластерами K_1 и K_2 .

Формулу (2.69) можно пояснить так: пусть кластер $K = K_1 \cup K_2$. Тогда для K

$$S = \sum_{l=1}^{n_1} (x_p - \bar{x})(x_p - \bar{x})^T + \sum_{s=n_1+1}^n (x_l - \bar{x})(x_l - \bar{x})^T, \quad (2.70)$$

где

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n \bar{x}_i = \frac{n_1 \bar{x}' + (n - n_1) \bar{x}''}{n}, \quad (2.71)$$

тогда из (2.70) можем записать

$$n_1 \cdot (\bar{x}' - \bar{x})(\bar{x}' - \bar{x})^T = \frac{n_1(n - n_1)^2}{n^2} (\bar{x}' - \bar{x}'')(\bar{x}' - \bar{x}'')^T,$$

$$n_2 \cdot (\bar{x}'' - \bar{x})(\bar{x}'' - \bar{x})^T = \frac{n_1^2(n - n_1)}{n^2} (\bar{x}' - \bar{x}'')(\bar{x}' - \bar{x}'')^T. \quad (2.72)$$

Складывая (2.71) и (2.72), получим

$$\frac{n_1(n - n_1)}{n} (\bar{x}' - \bar{x}'')(\bar{x}' - \bar{x}'')^T = M, \quad (2.73)$$

где M – матрица межгруппового рассеяния, тогда

$$S_x = S_{x'} + S_{x''} + M, \quad (2.74)$$

$$\rho_4 = \text{tr } M. \quad (2.75)$$

Пусть дано n объектов, разделенных по m кластерам. Тогда матрица A размерности $m \times n$ называется матрицей толерантности, если $a_{ij} = 1$, при том, что объект j принадлежит кластеру i , $a_{ij} = 0$ – в противном случае.

2.5.2. Математические критерии оптимальной кластеризации

Проблема определения процедуры разбиения анализируемых данных на кластеры остается открытой и после выбора меры сходства объектов. Критерий кластеризации может либо воспроизводить некие эвристические соображения, либо основываться на минимизации (или максимизации) какого-нибудь показателя качества.

При эвристическом подходе решающую роль играют интуиция и опыт. Он предусматривает задание набора правил, которые

обеспечивают использование выбранной меры сходства для отнесения объектов к одному из кластеров. Евклидово расстояние хорошо приспособлено для подобного подхода, что связано с естественностью его интерпретации как меры близости. Поскольку, однако, близость двух объектов является относительной мерой их подобия, обычно приходится вводить порог, чтобы установить приемлемые степени сходства для процесса отыскания кластеров.

Подход к кластеризации, предусматривающий использование показателя качества, связан с разработкой процедур, которые обеспечат минимизацию или максимизацию выбранного показателя качества. Одним из наиболее популярных показателей является сумма квадратов ошибки

$$J = \sum_{j=1}^{N_c} \sum_{x \in S_j} \|x - m_j\|^2, \quad (2.76)$$

где N_c – число кластеров; S_j – множество объектов, относящихся к кластеру j ;

$$m_j = \frac{1}{N_j} \sum_{x \in S_j} x - \quad (2.77)$$

вектор выборочных средних значений для множества S_j ; N_j характеризует количество объектов, входящих во множество S_j . Показатель качества определяет общую сумму квадратов отклонений характеристик всех объектов, входящих в некоторый кластер, от соответствующих средних значений по кластеру.

Другой широко применяемый критерий – критерий полной суммы внутрикластерных расстояний

$$J = \sum_{i=1}^N \sum_{k=1}^N \sum_{j=1}^N \rho_{kj} a_{ik} a_{ij}. \quad (2.78)$$

Здесь ρ_{ij} – евклидово расстояние между i -м и j -м объектами.

На матрицу толерантности накладывается ограничение «каждый объект принадлежит одному и только одному кластеру»

$$\sum_{i=1}^N a_{ij} = 1, \text{ где } j = 1, \dots, N. \quad (2.79)$$

Существует много показателей качества помимо рассмотренных: среднее квадратов расстояний между объектами в кластере; среднее квадратов расстояний между объектами, входящими в разные кластеры; показатели, основанные на понятии матрицы рассеяния; минимум и максимум дисперсии и т.д.

Недостатком таких критериев качества является их нелинейность и, как следствие, возрастание сложности оптимизации, как алгоритмической, так и вычислительной. Поэтому надо стремиться использовать линейные критерии в сочетании с выбором линейных же ограничений. В качестве примера можно рассмотреть критерий вида [2]

$$J = \sum_{i=1}^N \sum_{j=1}^N \rho_{ij} a_{ij} + \sum_{i=1}^N h \cdot a_{ii} \rightarrow \min \quad (2.80)$$

при ограничениях

$$\begin{aligned} \sum_{i=1}^N a_{ij} &= 1, \text{ где } j = 1, \dots, N; \\ \sum_{j=1, j \neq i}^N a_{ij} - N \cdot a_{ii} &\leq 0, \text{ где } i = 1, \dots, N. \end{aligned} \quad (2.81)$$

Здесь N – число объектов, подлежащих кластеризации; a_{ij} – элементы матрицы толерантности; ρ_{ij} – евклидово расстояние между i -м и j -м объектами; h – неотрицательный весовой коэффициент. Первое ограничение гарантирует, что каждый объект должен входить в один и только один кластер. Второе ограничение предотвращает возникновение кластеров без головного объекта. Для кластеров, содержащих головной объект, первое слагаемое критерия определяет сумму внутрикластерных расстояний.

Нередко применяются алгоритмы отыскания кластеров, основанные на совместном использовании эвристического подхода и показателя качества. Подобной комбинацией является алгоритм ИСОМАД (ISODATA).

В настоящее время разработано много методов решения задачи кластеризации. К точным методам относятся методы, явно оптимизирующие некоторый критерий качества (целевую функцию), например это методы целочисленного программирования. Приближенные методы имеют в большей или меньшей степени эвристическое происхождение, однако могут неявно также оптимизировать некоторый критерий качества. В качестве примера приближенных методов можно назвать семейство алгоритмов иерархической кластеризации и нейросетевые методы.

2.5.3. Методы оптимальной кластеризации

Как известно, решение задачи кластеризации с математической точки зрения принято представлять в виде матрицы толерантности $A = (a_{ij})$, которая по определению унимодулярна. Критерий качества кластеризации зависит от элементов этой матрицы a_{ij} , т.е. a_{ij} – переменные, по которым проводится оптимизация. Отсюда следует, что задача оптимизации такого критерия качества является задачей целочисленного программирования с булевыми переменными.

Целевая функция в общем случае является нелинейной, что сильно затрудняет (прежде всего, с точки зрения скорости) ее оптимизацию. Как правило, в задаче кластеризации стараются выбрать линейный критерий, который может быть оптимизирован методами линейного программирования. Одним из наиболее известных является целочисленный симплекс-метод (ЦСМ).

Обычный симплекс-метод не позволяет явно учесть ограничение целочисленности переменных и в общем случае дает нецелочисленное решение, поэтому Гомори (разд. 1.7.3) предложил метод отсечений, позволяющий за конечное число шагов привести полученное обычным симплекс-методом нецелочисленное решение к целочисленному. Если после решения задачи симплекс-методом без учета целочисленности не получен целочисленный результат, то строят дополнительные линейные ограничения, уменьшающие допустимую область. Дополнительное ограничение называется правильным отсечением, если отсекает такую часть допустимой области, в которой содержится оптимальное решение

нецелочисленной задачи, но нет допустимых решений целочисленной.

На практике реализация этого метода сопряжена со значительными трудностями. ЦСМ основан на методе отсечений Гомори, при использовании которого на основе дробных частей элементов нецелочисленной симплекс-таблицы строятся правильные отсечения. В то же время представление дробных чисел в современных компьютерах осуществляется в основном типами данных с плавающей запятой, которые имеют ограниченную по числу значащих цифр точность. Это приводит к тому, что последние цифры таких чисел являются незначащими и могут иметь произвольные значения, что, в свою очередь, приводит к невозможности достоверно сравнивать такие числа между собой. Одним из выходов является сравнение с некоторой допустимой погрешностью $\varepsilon \ll 1$, однако со временем при длительных преобразованиях эти погрешности накапливаются, и в некоторый момент превышают некоторые элементы симплекс-таблицы. Поэтому данный подход можно признать тупиковым.

Заметим, что исходная симплекс-таблица при использовании ограничений вида (2.81) является целочисленной, но не унимодулярной, что является причиной появления дробных чисел. В этом случае, если представлять элементы симплекс-таблицы не числами с плавающей запятой, а рациональными числами в виде пары двух целых чисел — числителя и знаменателя, то можно добиться абсолютной точности. Однако практика показала, что даже этот весьма емкий по меркам современных персональных компьютеров тип данных быстро (менее чем за 1 с расчетов) переполняется. Выходом является реализация с помощью ассемблера целочисленного типа большей разрядности.

Аддитивный алгоритм Балаша, как и целочисленный симплекс-метод (ЦСМ), позволяет максимизировать линейную целевую функцию (ЦФ) при линейных же ограничениях, но, в отличие от ЦСМ, работает только с булевыми переменными. Для решения задач такого вида Э. Балаш предложил алгоритм частичного (неявного) перебора [14], который получил название аддитивного, поскольку в силу условия (2.72) для вычисления по алгоритму используются лишь операции сложения (вычитания). Данный алгоритм позволяет либо найти оптимальное решение, либо

установить отсутствие такового, не проводя перебора всех допустимых значений переменных.

Приведем краткое описание этого метода. В обобщенном виде задача оптимальной кластеризации как задача линейного программирования имеет вид (1.1). Очевидно, что задача (2.80) является ее частным случаем – достаточно переобозначить и перенумеровать коэффициенты и переменные для приведения двойной суммы к одинарной.

Пусть z – ранее достигнутое значение ЦФ. Тогда зондируемое решение не имеет допустимого дополнения, улучшающего значение ЦФ, если

$$\sum_{j \in S} \min(a_{ij}, 0) > b_i - \sum_{j \in \Omega} a_{ij} x_j \text{ при } \forall i = 1, \dots, m, \quad (2.82)$$

где левая часть неравенства – сумма отрицательных коэффициентов при свободных переменных в i строке ограничений; S – совокупность номеров свободных переменных; Ω – совокупность номеров частичных решений.

Если для некоторой свободной переменной x_k выполняется

$$\sum_{j \in S} \min(a_{ij}, 0) + |a_{ik}| > b_i - \sum_{j \in \Omega} a_{ij} x_j \text{ для } \forall i, \text{ то} \quad (2.83)$$

$$x_k = \begin{cases} 0, & \text{если } a_{ik} > 0, \\ 1, & \text{если } a_{ik} < 0. \end{cases} \quad (2.84)$$

Этапы аддитивного алгоритма таковы.

1. Если основной список задач пуст, то переходим к п. 5. Если нет, то переходим к п. 2.

2. Проверка условия (2.82). Если оно выполняется, то переходим к новой задаче из списка основных задач (п. 1). Если нет, то вычисляются (2.83), (2.84) и переходим к п. 3.

3. Если допустимое дополнение частичного решения и зондируемое решение в совокупности состоят из n переменных, то это решение запоминается, вычисляется новое значение ЦФ и переходим к п. 1. Если нет, то переходим к п. 4.

4. Выбирается любая оставшаяся свободная переменная и в список основных задач вводятся две задачи: в одной эта переменная – нулевая, в другой – единичная. Переходим к п. 1.

5. Вывод результата: либо решения нет, либо максимум ЦФ и список из оптимальных значений n переменных.

В отличие от ЦСМ, данный метод не порождает сложностей в программной реализации. Однако весомым недостатком этого метода, как и метода зондирования решений, является длительное время их работы. Это вызвано тем, что используемая форма матрицы толерантности допускает до $N!$ (N – число ОТЕ, подлежащих кластеризации) возможных представлений одного и того же разбиения по кластерам за счет перестановок строк этой матрицы. Решаемая этими методами задача имеет почти в $N!$ раз большую размерность, чем на самом деле. Это и является причиной быстрого роста времени кластеризации при незначительном увеличении мощности множества ОТЕ.

Для решения этой проблемы можно предложить использовать альтернативную матрицу толерантности вида

$$\tilde{A} = (\tilde{a}_{ij}),$$

где

$$\tilde{a}_{ij} = \begin{cases} 1, & x_i, x_j \in \text{одному кластеру,} \\ 0, & \text{иначе,} \end{cases} \quad (2.85)$$

$i, j = 1, 2, \dots, N$.

Заметим, что матрица \tilde{A} – симметричная, это позволяет рассматривать только ее верхний правый (или нижний левый) треугольник. В этом случае каждое разбиение по кластерам представляется только одним возможным видом матрицы толерантности, и решаемая методами упорядоченного перебора задача оптимизации приобретает адекватную размерность.

Метод зондирования решений является разновидностью аддитивного алгоритма Балаша и часто не выделяется отдельно. Его отличие заключается в добавлении к системе ограничений дополнительного линейного ограничения, зависящего от достигнутой на данном шаге оценки ЦФ. Это ограничение позволяет значительно (в несколько раз) сократить множество перебираемых значений переменных. Однако это не решает проблему с факториальным ростом размерности задачи, характерную для аддитивного алгоритма.

Метод градиентного спуска (МГС) – итерационный метод нахождения минимума функции, в общем случае, многих переменных.

Пусть \bar{x} – точка в N -мерном пространстве, $f(\bar{x})$ – ЦФ, значение которой нужно минимизировать, тогда общая формула градиентного метода имеет вид

$$x_{k+1} = x_k - \lambda \nabla f(x_k), \quad (2.86)$$

где x_k – значение точки минимума $f(x)$, достигнутое на k -м шаге метода; $\nabla f(x_k)$ – значение градиента функции $f(x)$ в точке x_k ; λ – положительный шаг. Для запуска метода необходимо задать начальное приближение x_0 . В зависимости от способа выбора шага λ существуют различные градиентные методы.

Применение данного метода к задаче кластеризации имеет следующие особенности:

1) оптимизируемая целевая функция имеет в качестве аргумента булеву матрицу – матрицу толерантности;

2) шаг λ может принимать одно из трех возможных значений: $\{-1; 0; 1\}$;

3) градиентный метод не позволяет явно учесть ограничения (например, вида (2.79));

4) даже при учете ограничения (2.79) оптимизация традиционных оценочных критериев кластеризации, таких как полная сумма внутрикластерных расстояний (2.78) или сумма квадратов отклонений от центров кластеров (2.76), приводит к вырожденному решению, где каждый объект входит в монокластер.

Градиентный метод был использован для кластеризации при минимизации значения критерия:

1) полной суммы внутрикластерных расстояний (2.78) при ограничении (2.79);

2) псевдополной суммы внутрикластерных расстояний (2.80) при ограничениях (2.81).

В обоих случаях алгоритм градиентного спуска был модифицирован таким образом, чтобы на каждом шаге метода, включая начальный, матрица толерантности удовлетворяла условию (2.79).

Практическое применение метода показало, что, как правило, для улучшения значения критерия необходимо сделать шаг только по одному направлению (по которому производная от критерия максимальна), шаг по еще одному направлению без пересчета значений производных критерия приводил к ухудшению значения критерия.

Оптимизация критерия полной суммы внутрикластерных расстояний (2.78) с ограничением (2.79) приводит к полностью или частично вырожденному решению, в зависимости от начального приближений.

При оптимизации критерия псевдополной суммы внутрикластерных расстояний (2.80) на каждой итерации производилась проверка на выполнение второго ограничения системы (2.81), и запоминались лучшие решения, удовлетворяющие этой системе. Однако значения, которые принимает матрица толерантности в процессе минимизации, в подавляющем большинстве случаев не удовлетворяют второму ограничению системы (2.81), и кластеризация не производится.

Наиболее перспективным выходом из описанных трудностей является использование других оптимизируемых критериев, не требующих введения в процесс градиентной оптимизации искусственного контроля за ограничениями.

2.5.4. Приближенные методы кластеризации

В приближенных методах кластеризации явно оптимизируемая ЦФ отсутствует. Как и в точных методах, результат кластеризации представляется в виде матрицы толерантности, однако ее элементы могут не принимать непосредственного участия в процессе кластеризации, как это имеет место, например, при использовании нейросетей (НС).

Метод иерархической кластеризации заключается в объединении объектов в достаточно большие кластеры, используя заданную меру сходства или расстояние между объектами. Ход такой кластеризации можно представить в виде иерархического дерева (дендограммы) [18] (рис. 2.12).

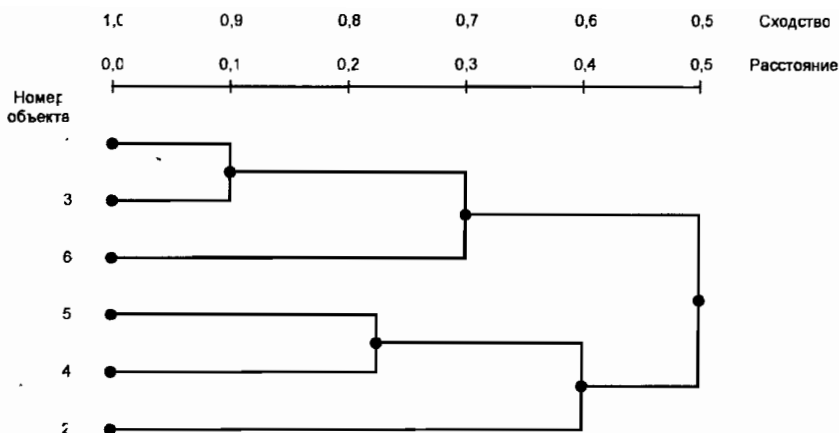


Рис. 2.12. Пример дендограммы

Выполнение метода начинается с того, что каждый объект в выборке считается монокластером – кластером, состоящим из одного объекта. Далее последовательно «ослабляется» критерий о том, какие объекты являются уникальными, а какие нет. Другими словами, понижается порог, относящийся к решению об объединении двух или более объектов в один кластер. В результате связывается вместе все большее и большее число объектов и агрегируется (объединяется) все больше и больше кластеров, состоящих из все сильнее различающихся элементов. Окончательно, на последнем шаге все объекты объединяются вместе.

На первом шаге, когда каждый объект представляет собой отдельный кластер, расстояния между этими объектами определяются выбранной мерой. Однако когда связываются вместе несколько объектов, возникает вопрос об определении расстояния между кластерами. Необходимо правило объединения или связи для двух кластеров. Можно связать два кластера вместе, когда любые два объекта в двух кластерах ближе друг к другу, чем соответствующее расстояние связи: используется «правило ближайшего соседа» для определения расстояния между кластерами; этот метод называется методом одиночной связи. Это правило строит «волокнуистые» кластеры, т.е. кластеры, «сцепленные вместе» только отдельными элементами, случайно

оказавшимися ближе остальных друг к другу. Как альтернативу можно использовать соседей в кластерах, которые находятся дальше всех остальных пар объектов друг от друга. Этот метод называется методом полной связи. Существует также множество других методов объединения кластеров, основанных на различных подходах к объединению кластеров. Наиболее часто встречающиеся варианты иерархической кластеризации рассмотрены ниже.

Одиночная связь (метод ближайшего соседа). Как было описано, в этом методе расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах. Это правило должно, в известном смысле, «нанизывать» объекты вместе для формирования кластеров, и результирующие кластеры имеют тенденцию быть представленными длинными «цепочками».

Полная связь (метод наиболее удаленных соседей). В этом методе расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах («наиболее удаленными соседями»). Этот метод обычно работает очень хорошо, когда объекты происходят на самом деле из реально различных «групп». Если же кластеры имеют в некотором роде удлиненную форму или их естественный тип является «цепочечным», то этот метод непригоден.

Невзвешенное попарное среднее. В этом методе расстояние между двумя различными кластерами вычисляется как среднее расстояние между всеми парами объектов в них. Метод эффективен, когда объекты в действительности формируют различные «группы», однако он работает одинаково хорошо и в случаях протяженных («цепочного» типа) кластеров. Снит и Сокэл [15] вводят аббревиатуру UPGMA для ссылки на этот метод, как на метод невзвешенного попарного арифметического среднего – unweighted pair-group method using arithmetic averages.

Взвешенное попарное среднее. Метод идентичен методу невзвешенного попарного среднего, за исключением того, что при вычислениях размер соответствующих кластеров (т.е. число объектов, содержащихся в них) используется в качестве весового коэффициента. Поэтому предлагаемый метод должен быть использован (скорее даже, чем предыдущий), когда

предполагаются неравные размеры кластеров. Снит и Сокэл вводят аббревиатуру WPGMA для ссылки на этот метод, как на метод взвешенного попарного арифметического среднего – weighted pair-group method using arithmetic averages.

Невзвешенный центроидный метод. В этом методе расстояние между двумя кластерами определяется как расстояние между их центрами тяжести. Снит и Сокэл используют аббревиатуру UPGMC для ссылки на этот метод, как на метод невзвешенного попарного центроидного усреднения – unweighted pair-group method using the centroid average.

Взвешенный центроидный метод (медиана). Этот метод идентичен предыдущему, за исключением того, что при вычислениях используются веса для учета разницы между размерами кластеров (т.е. числами объектов в них). Поэтому, если имеются (или подозреваются) значительные отличия в размерах кластеров, этот метод оказывается предпочтительнее предыдущего. Снит и Сокэл использовали аббревиатуру WPGMC для ссылок на него, как на метод взвешенного попарного центроидного усреднения – weighted pair-group method using the centroid average.

Метод Варда. Этот метод отличается от всех других методов, поскольку он использует методы дисперсионного анализа для оценки расстояний между кластерами. Метод минимизирует сумму квадратов (SS) для любых двух (гипотетических) кластеров, которые могут быть сформированы на каждом шаге. Метод является очень эффективным, однако он стремится создавать кластеры малого размера.

К новым приближенным методам кластеризации можно отнести и нейросетевые подходы [19], [20]. Рассмотрим их возможности на примере достаточно часто применяемой нейросети Кохонена. Нейросеть (НС) Кохонена относится к нейросетям без учителя. Это означает, что подстройка весов нейронов (обучение) такой сети производится исключительно на основе поступающих на ее вход данных. В традиционной нейросети без учителя используется так называемое соревновательное обучение, при этом выходы сети максимально скоррелированы: при любом значении входа активность всех нейронов, кроме нейрона-победителя, одинакова и равна нулю. Такой режим функционирования сети называется «победитель забирает все».

Нейрон-победитель (с индексом i^*), свой для каждого входного вектора, будет служить прототипом этого вектора. Поэтому победитель выбирается так, что его вектор весов \mathbf{w}_{i^*} , определенный в том же d -мерном пространстве, находится ближе к данному входному вектору \mathbf{x} , чем у всех остальных нейронов: $|\mathbf{w}_{i^*} - \mathbf{x}| \leq |\mathbf{w}_i - \mathbf{x}|$ для всех i . Если, как это обычно и делается, применять правила обучения нейронов, обеспечивающие одинаковую нормировку всех весов, например $|\mathbf{w}_i| = 1$, то победителем окажется нейрон, дающий наибольший отклик на данный входной стимул: $\mathbf{w}_{i^*} \cdot \mathbf{x} \geq \mathbf{w}_i \cdot \mathbf{x} \quad \forall i$. Выход такого нейрона усиливается до единичного, а остальных – подавляется до нуля.

Количество нейронов в соревновательном слое определяет максимальное разнообразие выходов и выбирается в соответствии с требуемой степенью детализации входной информации. Обученная сеть может затем классифицировать входы: нейрон-победитель определяет, к какому классу относится данный входной вектор.

По предъявлении очередного примера корректируются лишь веса нейрона-победителя

$$\Delta \mathbf{w}_{i^*}^\tau = \eta (\mathbf{x}^\tau - \mathbf{w}_{i^*}). \quad (2.87)$$

Здесь η – скорость обучения; \mathbf{w}_{i^*} – вектор весов нейрона-победителя; \mathbf{x}^τ – предъявленный пример.

Описанный базовый алгоритм обучения на практике обычно несколько модифицируют, так как он, например, допускает существование так называемых мертвых нейронов, которые никогда не выигрывают, и, следовательно, бесполезны. Самый простой способ избежать их появления – выбирать в качестве начальных значений весов случайно выбранные в обучающей выборке входные вектора. Такой способ хорош еще и тем, что при достаточно большом числе прототипов он способствует равной «нагрузке» всех нейронов-прототипов. Это соответствует максимизации энтропии выходов в случае соревновательного слоя. В идеале каждый из нейронов соревновательного слоя должен одинаково часто становиться победителем, чтобы априори невозможно было бы предсказать, какой

из них победит при случайном выборе входного вектора из обучающей выборки.

Наиболее быструю сходимость обеспечивает пакетный (batch) режим обучения, когда веса изменяются лишь после предъявления всех примеров. В этом случае можно сделать приращения не малыми, помещая вес нейрона на следующем шаге сразу в центр тяжести всех входных векторов, относящихся к его ячейке. Такой алгоритм сходится за $O(1)$ итераций.

Записав правило соревновательного обучения в градиентном виде: $\langle \Delta \mathbf{w} \rangle = -\eta \frac{\partial E}{\partial \mathbf{w}}$, легко убедиться, что оно минимизирует квадратичное отклонение входных векторов от их прототипов – весов нейронов-победителей:

$$E = \frac{1}{2} \sum_{\alpha} \left| \mathbf{x}^{\alpha} - \mathbf{w}_{*}^{\alpha} \right|^2. \quad (2.88)$$

Иными словами, сеть осуществляет кластеризацию данных: находит такие усредненные прототипы, которые минимизируют ошибку округления данных. Недостаток такого варианта кластеризации очевиден – «навязывание» количества кластеров, равного числу нейронов.

Один из вариантов модификации базового правила обучения соревновательного слоя состоит в том, чтобы обучать не только нейрон-победитель, но и его «соседей», хотя и с меньшей скоростью. Такой подход – «подтягивание» ближайших к победителю нейронов – применяется в топографических картах Кохонена. Небольшой модификацией соревновательного обучения можно добиться того, что положение нейрона в выходном слое будет коррелировать с положением прототипов в многомерном пространстве входов сети: близким нейронам будут соответствовать близкие значения входов. Тем самым, появляется возможность строить топографические карты, чрезвычайно полезные для визуализации многомерной информации. Обычно для этого используют соревновательные слои в виде двумерных сеток. Такой подход сочетает квантование данных с отображением, понижающим размерность. Причем это достигается с помощью всего лишь одного слоя нейронов, что существенно облегчает обучение.

Нейроны выходного слоя упорядочиваются, образуя одно- или двумерные решетки, т.е. теперь положение нейронов в такой решетке маркируется векторным индексом \mathbf{i} . Такое упорядочение естественным образом вводит расстояние между нейронами $|\mathbf{i} - \mathbf{j}|$ в слое. Модифицированное Кохоненом правило соревновательного обучения учитывает расстояние нейронов от нейрона-победителя:

$$\Delta \mathbf{w}_i^r = \eta \Lambda(|\mathbf{i} - \mathbf{i}^*|) (\mathbf{x}^r - \mathbf{w}_i). \quad (2.89)$$

Функция соседства $\Lambda(|\mathbf{i} - \mathbf{i}^*|)$ равна единице для нейрона-победителя с индексом \mathbf{i}^* и постепенно спадает с расстоянием, например по закону $\Lambda(a) = \exp(-a^2/\sigma^2)$. Как темп обучения η , так и радиус взаимодействия нейронов σ постепенно уменьшаются в процессе обучения, так что на конечной стадии обучения мы возвращаемся к базовому правилу адаптации весов только нейронов-победителей.

После обучения нейросети Кохонена вектора весов нейронов единственного слоя представляют собой ни что иное, как центры кластеров. Принадлежность объекта кластеризуемого множества к определенному кластеру определяется близостью вектора объекта к вектору весов нейрона.

Тестовые множества могут различаться одним или несколькими из следующих параметров: размерность пространства дескрипторов (П-пространства) d ; общее количество кластеризуемых объектов N .

Основные характеристики компьютера, на котором проводилось тестирование, приведены в табл. 2.11.

Таблица 2.11

Название параметра	Значение параметра
Процессор	AMD Athlon 1200 МГц
Объем и тип оперативной памяти	512 МБ DDR-266
Операционная система	Windows 2000 Professional SP3

Параметры тестовых множеств представлены в табл. 2.12.

Таблица 2.12

Название параметра	Значение параметра
Диапазон значений параметра	[0; 10]
Распределение значений в диапазоне	Равномерное

Настройки для методов кластеризации UPGMC и НС Кохонена приведены в табл. 2.13.

Отметим, что для каждой серии экспериментов число кластеров необходимо задавать явно (НС Кохонена) или неявно (UPGMC). Для НС Кохонена число нейронов в слое является ограничением сверху числа кластеров и одновременно наиболее вероятным (по данным экспериментов) числом выделяемых кластеров. В некоторых случаях нейросеть Кохонена с N нейронами в единственном слое выделяет $N - 1$ кластер, что можно отметить как положительное свойство НС учитывать реальную структуру данных.

Таблица 2.13

Название параметра	Значение параметра
Активационная функция нейронов НС Кохонена	Линейная: $f(x) = x$
Число нейронов в слое НС Кохонена	Равно заданному числу кластеров
Начальные веса слоя НС Кохонена	Случайно выбранные объекты кластеризуемого множества
Точность обучения НС Кохонена	0,01
Начальный радиус взаимодействия нейронов	1,0
Коэффициент изменения радиуса взаимодействия	0,9
Начальная скорость обучения	1,0
Коэффициент изменения скорости обучения	1,0
Пакетный режим обучения	Да
Критическое расстояние UPGMC	Для получения заданного числа кластеров
Частота обновления информации, 1/итер.	100

На основе известного тестового множества подбирается критическое расстояние UPGMC для получения заданного числа кластеров. Затем найденное расстояние используется для проведения эксперимента.

Для НС Кохонена проводилось по три эксперимента с различными начальными приближениями, и в качестве результата бралось среднее арифметическое. Используются следующие обозначения для критериев оценки качества кластеризации: полная сумма

внутрикластерных расстояний – ПСВКР; сумма квадратов отклонений от центра кластера – СКО.

2.5.5. Зависимость времени и качества кластеризации от количества объектов, кластеров и размерности признакового пространства

Положим число кластеров $m = 5$, размерность признакового пространства $d = 3$. Результаты испытаний приведены в табл. 2.14 и на рис. 2.13, 2.14.

Таблица 2.14

Параметр	Количество кластеризуемых объектов			
	10	50	75	100
Время UPGMC, мин:с.мс	00:00.019	00:04.325	00:41.000	03:42.000
Время НС Кохонена, мин:с.мс	00:00.060	00:00.477	00:00.811	00:01.191
Полная сумма внутрикластерных расстояний UPGMC	24	1134	2867	9673
Полная сумма внутрикластерных расстояний НС Кохонена	28	903	1989	4412
Сумма квадратов отклонений UPGMC	30	440	784	1427
Сумма квадратов отклонений НС Кохонена	45	335	508	900

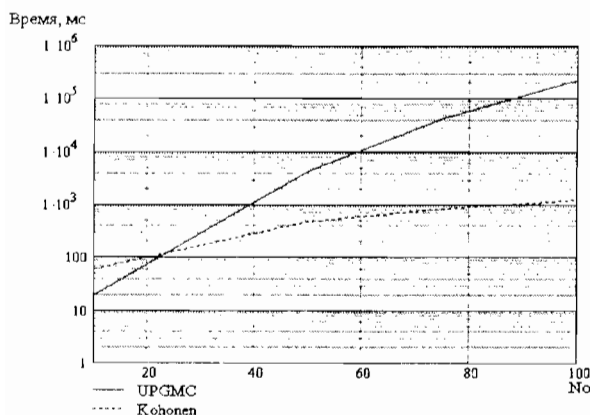


Рис. 2.13. Зависимость времени кластеризации от количества объектов

Как видно из рис. 2.13, время кластеризации для обоих алгоритмов растет экспоненциально, однако для алгоритма НС Кохонена коэффициент роста значительно ниже, в результате уже примерно при 25 объектах в кластеризуемом множестве метод UPGMC начинает проигрывать по времени. Во всех случаях нейросеть Кохонена дает лучшее значение по критерию полной суммы внутрикластерного расстояния (рис. 2.14) и по критерию среднеквадратичной ошибки (рис. 2.15).

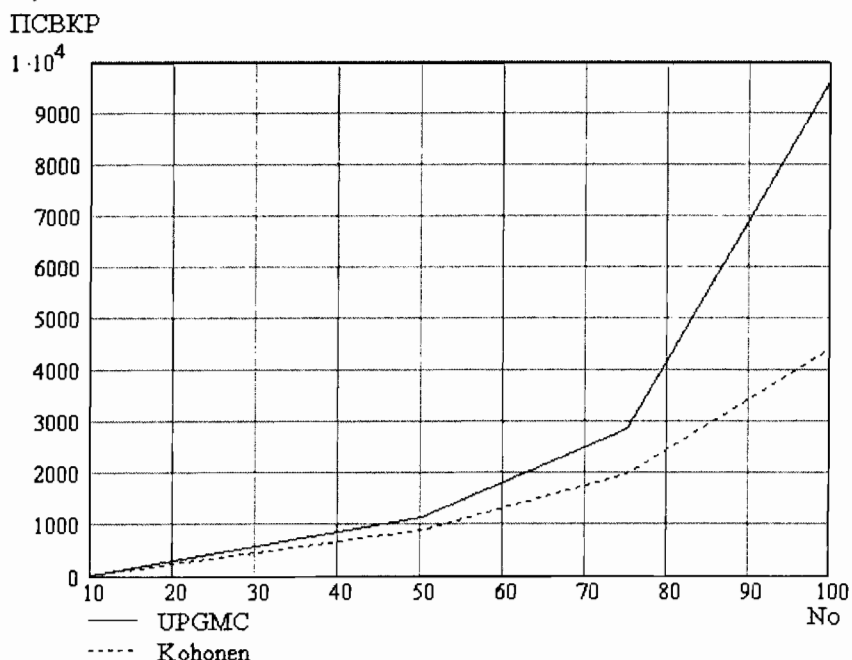


Рис. 2.14. Зависимость значения критерия качества (ПСВКР) от количества объектов

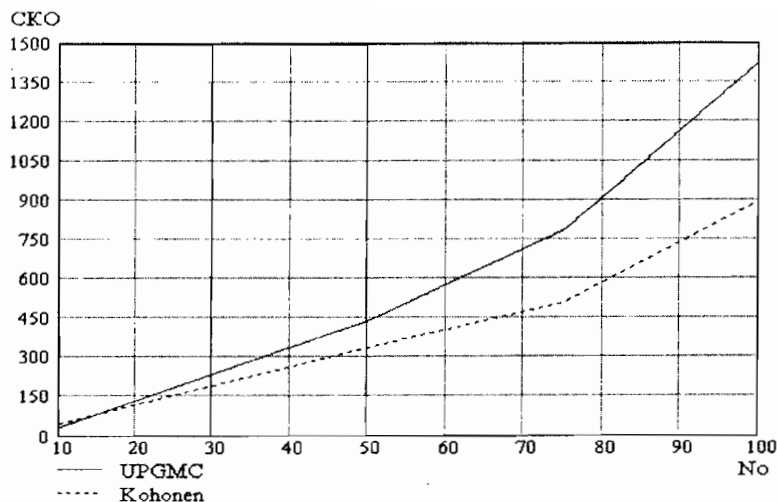


Рис. 2.15. Зависимость значения критерия качества (СКО) от количества объектов

В данном эксперименте положим число объектов в кластеризуемом множестве $N=50$, размерность признакового пространства $d=3$. Будем кластеризовать одно и то же множество для получения разного количества кластеров (напомним, что оно задается числом нейронов в слое НС Кохонена и неявно — критическим расстоянием в UPGMC). Результаты испытаний приведены в табл. 2.15.

Как следует из рис. 2.16, время кластеризации для метода UPGMC практически не зависит от количества кластеров (это подтверждается и теоретически, так как число итераций алгоритма невзвешенного попарного усреднения равно разности числа объектов кластеризуемого множества и количества выделяемых методом кластеров). Для НС Кохонена наблюдается примерно экспоненциальный рост, связанный с тем, что количество выделяемых НС кластеров определяется числом нейронов в единственном слое; следовательно, при одном и том же числе объектов в кластеризуемом множестве увеличение числа нейронов повышает также и число операций на каждой итерации алгоритма нейросетевой кластеризации.

Таблица 2.15

Параметр	Количество кластеров			
	2	3	4	5
Время UPGMC, мин:с.,мс	00:05.366	00:05.927	00:05.588	00:04.325
Время НС Кохонена, мин:с.,мс	00:00.143	00:00.317	00:00.400	00:00.477
Полная сумма внутрикластерных расстояний UPGMC	4574	2353	1784	1134
Полная сумма внутрикластерных расстояний НС Кохонена	3167	1832	1266	903
Сумма квадратов отклонений UPGMC	921	665	579	440
Сумма квадратов отклонений НС Кохонена	759	549	435	335

Таблица 2.15 (окончание)

Параметр	Количество кластеров			
	6	7	8	9
Время UPGMC, мин:сек.мс	00:05.336	00:04.847	00:04.920	00:05.257
Время НС Кохонена, мин:сек.мс	00:00.537	00:01.001	00:01.171	00:00.987
Полная сумма внутрикластерных расстояний UPGMC	949	905	825	681
Полная сумма внутрикластерных расстояний НС Кохонена	645	534	459	409
Сумма квадратов отклонений UPGMC	365	333	314	272
Сумма квадратов отклонений НС Кохонена	281	237	210	192

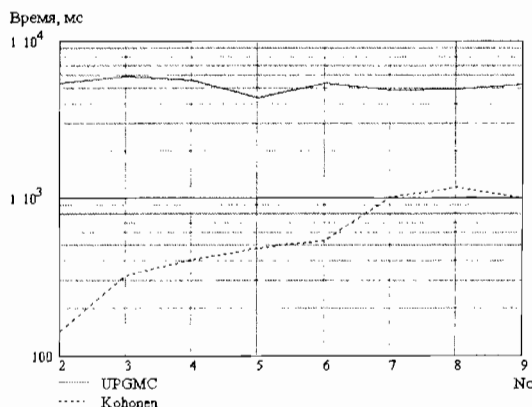


Рис. 2.16. Зависимость времени кластеризации от количества кластеров

Рис. 2.17 и 2.18 показывают превосходство алгоритма НС Кохонена по качеству кластеризации над методом UPGMC.

Зависимость времени и качества кластеризации от размерности пространства исследовалась при следующих условиях: количество кластеров $m = 5$, число объектов в кластеризуемом множестве $N = 50$. Результаты приведены в табл. 2.16.

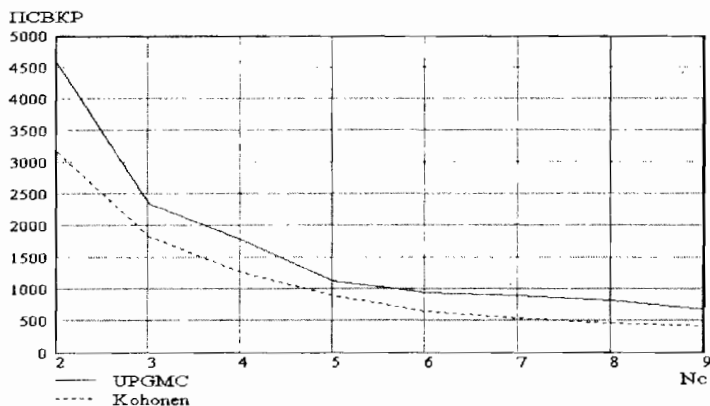


Рис. 2.17. Зависимость значения критерия качества от количества кластеров (ПСВКР)

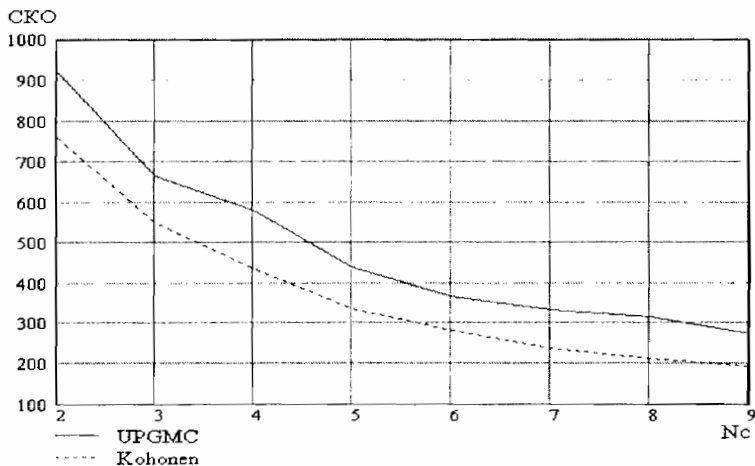


Рис. 2.18. Зависимость значения критерия качества от количества кластеров (СКО)

Таблица 2.16 (начало)

Параметр	Размерность пространства			
	2	3	4	5
Время UPGMC, мин:с.мс	00:04.966	00:04.325	00:05.268	00:05.106
Время НС Кохонена, мин:с.мс	00:00.637	00:00.477	00:01.155	00:00.777
Полная сумма внутрикластерных расстояний UPGMC	1033	1134	1647	4727
Полная сумма внутрикластерных расстояний НС Кохонена	585	903	1402	1692
Сумма квадратов отклонений UPGMC	255	440	821	1478
Сумма квадратов отклонений НС Кохонена	155	335	720	1058

Таблица 2.16 (окончание)

Параметр	Размерность пространства			
	6	7	8	9
Время UPGMC, мин:с.,мс	00:05.086	00:05.098	00:04.916	00:04.386
Время НС Кохонена, мин:с.,мс	00:00.607	00:00.683	00:00.854	00:01.233
Полная сумма внутрикластерных расстояний UPGMC	5834	2851	2775	6392
Полная сумма внутрикластерных расстояний НС Кохонена	1645	2089	2247	2658
Сумма квадратов отклонений UPGMC	1851	1937	2366	2811
Сумма квадратов отклонений НС Кохонена	1224	1627	2041	2504

На рис. 2.19 виден слабый рост времени для нейросети и отсутствие роста для иерархического алгоритма. Если рассмотреть суть последнего, то станет ясно, что подсчет центров кластеров на каждой итерации также приводит к увеличению времени, однако оно весьма незначительно. Для алгоритма Кохонена этот рост вызван введением с увеличением размерности признакового пространства дополнительных входов нейронов. При числе признаков в несколько десятков, возможно, нейросетевой алгоритм

уступит UPGMC, однако на практике задачи такой размерности редки.

Как любой итерационный метод, алгоритм кластеризации при помощи НС Кохонена чувствителен к выбору начального приближения. Графики минимальных, средних и максимальных значений времени и качества кластеризации можно видеть на рис. 2.20, 2.21, 2.22.

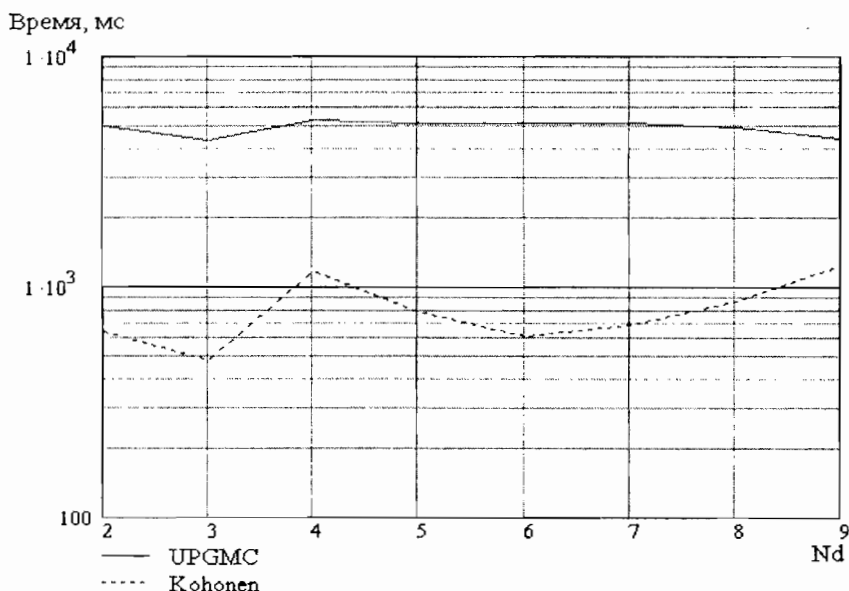


Рис. 2.19. Зависимость времени кластеризации от размерности пространства

ПСВКР

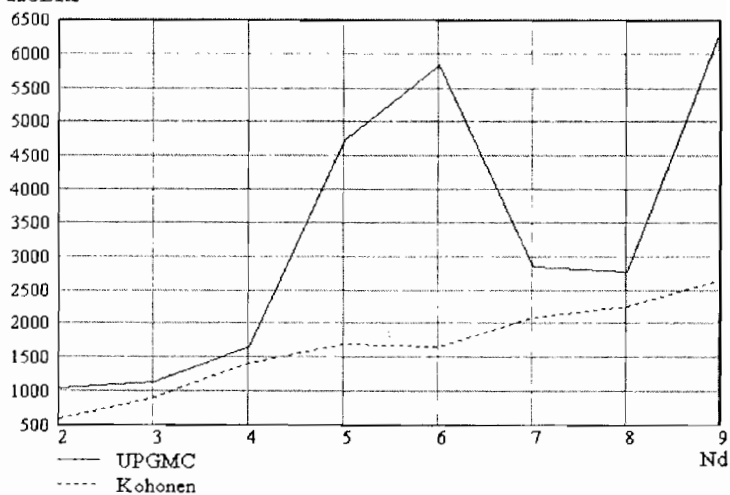


Рис. 2.20. Зависимость значения критерия качества от размерности пространства (ПСВКР)

СКО

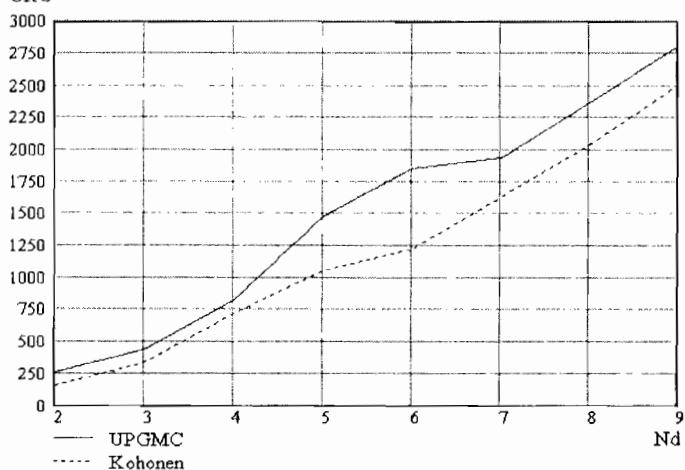


Рис. 2.21. Зависимость значения критерия качества от размерности пространства (СКО)

Для исследования диапазона значений времени и критериев качества кластеризации была проведена серия экспериментов со следующими параметрами: число кластеров $m = 5$, число объектов в кластеризуемом множестве $N = 50$, размерность П-пространства – перменная от 2 до 9. Для каждой совокупности из 50 ОТЕ в пространстве соответствующей размерности было проведено десять экспериментов. В качестве начального приближения берется случайная выборка из множества кластеризуемых объектов.

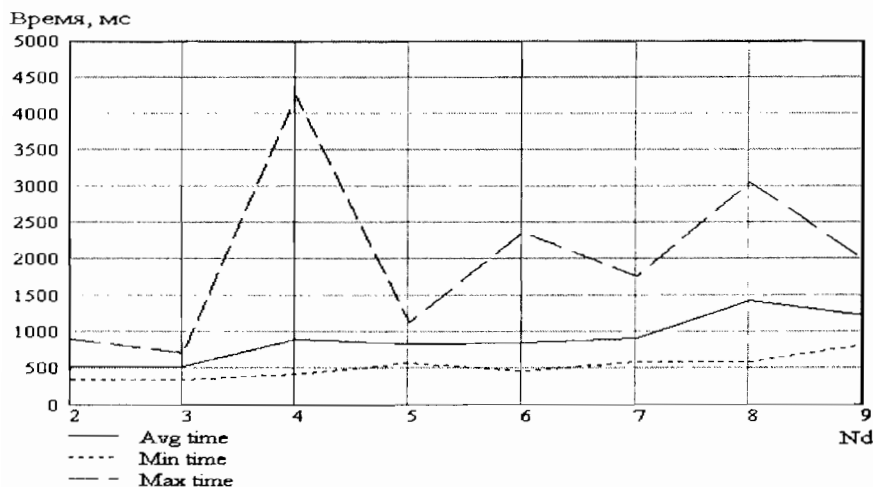


Рис. 2.22. Зависимость диапазона значений времени кластеризации при помощи НС Кохонена от размерности пространства

Как следует из рис. 2.20 – 2.22, разброс времени кластеризации может достигать сотен процентов, разброс значений критериев не превышает 40 % для критерия ПСВКР, а для критерия СКО – 15 %.

Можно сделать вывод о првосходстве метода кластеризации с помощью НС Кохонена над алгоритмом невзвешенного попарного центроидного усреднения как по времени работы, так и по качеству. Для 100 объектов в кластеризуемом множестве и 5 выделяемых кластеров НС Кохонена находит на 219 % (по критерию полной суммы внутрикластерных расстояний) более качественное решение в 186 раз быстрее.

В то же время метод кластеризации при помощи НС Кохонена обладает значительным разбросом времени и качества в зависимости от начального приближения (рис. 2.23, 2.24), что является существенным недостатком.

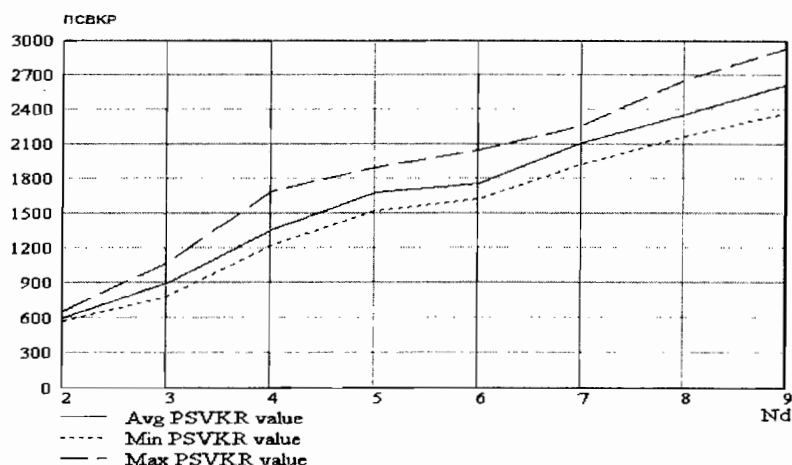


Рис. 2.23. Зависимость диапазона значений критерия ПСВКР при кластеризации с помощью НС Кохонена от размерности пространства

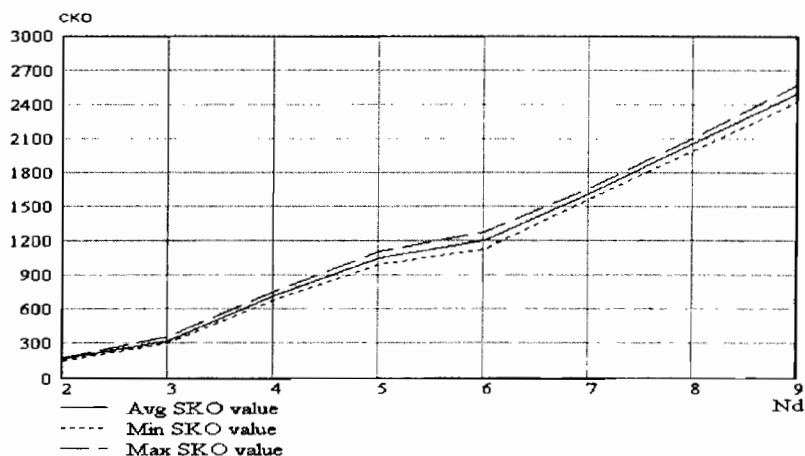


Рис. 2.24. Зависимость диапазона значений критерия СКО при кластеризации с помощью НС Кохонена от размерности пространства

В заключение необходимо подчеркнуть, что сделанные выводы касаются лишь сравнения приближенных методов кластеризации. Недавние сравнительные исследования нейросетевых подходов с модифицированным аддитивным алгоритмом «рекурсивного ветвления» [7] показали в некоторых классах задач кластеризации превосходство алгоритма оптимальной кластеризации над нейросетью Хопфилда по качеству при сравнимых скоростных характеристиках.

СПИСОК ЛИТЕРАТУРЫ

ГЛАВА 1

1. *Ашиманов С.А.* Линейное программирование. М.: Наука, 1981.
2. *Бахвалов Н.С.* Численные методы (анализ, алгебра, обыкновенные дифференциальные уравнения). М.: Наука, 1975.
3. *Васильев Ф.П., Иваницкий А.Ю.* Линейное программирование. М.: Факториал Пресс. 2003.
4. *Гвоздев С.Е.* Математическое программирование (двойственность, транспортные задачи): Новосибирск: Новосибирский государственный архитектурно-строительный университет, 2001.
5. *Карманов В.Г.* Математическое программирование. М.: Наука, 1986.
6. *Карманов В.Г.* Математическое программирование: учебное пособие. 5-е изд. М: Физматлит, 2004.
7. *Максимов Ю.Я.* Алгоритмы линейного и дискретного программирования. М.: МИФИ, 1980.
8. *Салмин И.Д.* Математические методы решения оптимизационных задач: учебное пособие. М.: МИФИ, 2004.
9. *Салмин И.Д.* Математическое программирование. Ч. 1, 2. М.: МИФИ, 1978.
10. *Струченков В.И.* Математическое программирование: методы, задачи, обучающие компьютерные программы: учебное пособие. М.: МИФИ, 2004.

ГЛАВА 2

1. *Кулябичев Ю.П., Анитова Т.В.* Теоретико-игровые методы исследования сложных систем. М.: МИФИ, 1994.

2. Кулябичев Ю.П., Крицына Н.А. Лабораторный практикум «Теоретико-игровые методы исследования сложных систем». М.: МИФИ, 1984.

3. Оуэн Г. Теория игр. М.: Мир, 1973.

4. Петросян Л.А., Зенкевич Н.А., Семина Е.А. Теория игр. М.: Высшая школа 1998.

5. Lemke C.E., Howson J.J. Equilibrium points of bimatrix games // J. of the Society for Industrial and Mathematics. 1964. V. 12. P. 413 – 423.

6. Золотов В.П. и др. Общие принципы исследования эффективности боевых действий и вооружения: тексты лекций. М.: ВАД, 1978.

7. Новикова Н.М. Основы оптимизации (курс лекций). М., 1998. (www.ccas.ru).

8. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. М.: Мир, 1985.

9. Бушин С.В., Крицына Н.А. Распределение авиационных ресурсов на основе методов исследования операций. Труды XIII Международного научно-технического семинара «Современные технологии в задачах управления, автоматизации и обработки информации». Алушта, 2004.

10. Владимиров В.И. Практические задачи по эксплуатации ядерных реакторов. М.: Атомиздат, 1976.

11. Загребяев А.М., Наумов В.И. О минимизации потери энерговыработки системы реакторов, работающих в переменном графике нагрузки // Атомная энергия. 1979. Т.47. Вып.3. С.165-166.

12. Ибраев О.Е., Шумилов Ю.Ю. Методика оптимального размещения радиолокационных станций. Труды XIV Международного научно-технического семинара “Современные технологии в задачах управления, автоматизации и обработки информации”. Алушта, 2005. С. 191.

13. Гудков Д.Е. Построение математической модели области действия радиолокационной станции // Научная сессия МИФИ-2000. Сборник научных трудов. Т.2 М.: МИФИ, 2000. С. 27.

14. Зимин Г.В., Бурмистров С.К., Букин Б.М. Справочник офицера ПВО. М.: Воениздат, 1987.

15. Дюран Б., Оделл П. Кластерный анализ. М.: Статистика, 1977.

16. Бородакий Ю.В., Муравьев С.К., Шумилов Ю.Ю. Нейросетевые методы оптимальной кластеризации воздушных объектов. Труды XIII Международного научно-технического семинара "Современные технологии в задачах управления, автоматизации и обработки информации". Алушта, 2004, Ч. II. С. 216-218.

17. Васильев Ф.П. Численные методы решения экстремальных задач. М.: Наука, 1988.

18. Мещеряков Р.В., Шумилов Ю.Ю. Сравнительный анализ алгоритмов кластеризации. Труды X Международного научно-технического семинара "Современные технологии в задачах управления, автоматизации и обработки информации". Алушта, 2001.

19. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его применение в экономике и бизнесе. М.: МИФИ, 1998.

20. Кулябичев Ю.П., Шляхов А.В., Шумилов Ю.Ю. Сравнительный анализ методов кластеризации. Труды XIII Международного научно-технического семинара "Современные технологии в задачах управления, автоматизации и обработки информации". Алушта, 2003. С. 362-363.

21. Муравьев С.К. Методика построения системы укрупнения и отображения воздушной обстановки. Труды X Международного научно-технического семинара "Современные технологии в задачах управления, автоматизации и обработки информации". Алушта, 2001. С. 247-248.

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ В СОВРЕМЕННЫХ ЗАДАЧАХ ОПТИМИЗАЦИИ

Учебное пособие

Редактор М.В. Макарова

Подписано в печать 18.06.2008. Формат 60х84 1/16
Уч.-изд.л. 11,75. Печ.л. 11,75. Тираж 100 экз.
Изд. № 041-1. Заказ № 226

*Московский инженерно-физический институт
(государственный университет).
Типография МИФИ.
115409 Москва, Каширское ш., 31*