МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Белгородский государственный технологический университет
им. В.Г. Шухова

А.Г.Брусенцев, В.И. Петрашев, Ю.Д. Рязанов

ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
И ТЕОРИЯ ИГР

Рекомендовано федеральным государственным бюджетным образовательным учреждением высшего профессионального образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)» в качестве учебного пособия для специалистов высших учебных заведений, обучающихся по специальности 090303 «Информационная безопасность автоматизированных систем»
Регистрационный номер рецензии 2143 от 7 декабря 2012 г.
МГУП имени Ивана Федорова

Белгород
2012

УДК 519.8
ББК 22.12
Б89

Рецензенты:

Доктор технических наук, профессор Белгородского государственного технологического университета им. В.Г.Шухова *Г.М Редькин*
Кандидат физико-математических наук, доцентБелгородского государственного национального исследовательского университета (НИУ «БелГУ») *В.В.Флоринский*

Брусенцев, А.Г.
Б89 Исследование операций и теория игр: учеб.пособие /А.Г. Брусенцев, В.И. Петрашев, Ю.Д. Рязанов. — Белгород: Изд-во БГТУ, 2012. — 258 с.

ISBN 978-5-361-00191-0

В пособии рассматриваются методы решения задач линейного, дискретного, нелинейного и динамического программирования. Представлены элементы теории игр, теории массового обслуживания и потоковые алгоритмы. Изложение иллюстрируется большим количеством примеров. Каждая глава снабжена контрольными вопросами и задачами для самостоятельной работы. Целью пособия является компактное популярное изложение, ведущее к полноценному усвоению основных методов и алгоритмов исследования операций и теории игр.

Учебное пособие предназначено для студентов высших учебных заведений, обучающихся по специальности 090303 «Информационная безопасность автоматизированных систем», а также может быть использовано и студентами других инженерно-экономических специальностей, изучающих дисциплину «Исследование операций и теория игр».

УДК 519.8
ББК 22.12

ISBN 978-5-361-00191-0

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Настоящее пособие написано на основе курсов лекций, в течение ряда лет читавшихся для студентов различных специальностей Белгородского государственного технологического университета им. В.Г. Шухова. В настоящее время исследование операций — одна из быстро развивающихся отраслей науки, которая широко используется в промышленности, сельском хозяйстве, торговле, на транспорте, при финансовых операциях и т.д. Это обусловливает важность оптимальной организации ее преподавания. Существует целый ряд хороших учебных пособий по этой дисциплине, однако большинство из них либо очень подробны, а потому имеют довольно большой объем, либо страдают неполнотой изложения. Поскольку учебное время, выделяемое на соответствующие курсы большинства специальностей, весьма ограничено, возникает потребность в пособии, которое, с одной стороны, обладает необходимой компактностью, а с другой — достаточно полно отражает содержание курса. Настоящее пособие написано с целью удовлетворения этим двум противоположным условиям.

Содержание любого пособия по дисциплине прикладной математики состоит из описаний:

1) источников возникновения задач этой дисциплины в предметной области;
2) построения алгоритмов решения этих задач;
3) обоснования алгоритмов решения.

Все три части являются необходимыми, однако возможны различные соотношения между ними. Гипертрофия или отсутствие каждой из этих частей отрицательно сказывается на усвоении курса студентами. В настоящем пособии предлагается определенный баланс частей 1) — 3), в котором больший вес имеет часть 2). Меньшее внимание уделяется математическим тонкостям части 3), а развернутые доказательства некоторых сложных теорем вовсе отсутствуют. Вместо этого суть дела разбирается на конкретных примерах. Авторы стремились к популярности изложения, которая, однако, не препятствовала бы полноценному усвоению основных методов и алгоритмов, а также всего курса в целом. Разумеется, охватить все содержание дисциплины в таком пособии невозможно. Здесь представлены лишь основы большинства разделов. Ряд вопросов излагается в обзорном порядке, и даются литературные ссылки для подробного углубленного изучения. Каждая глава снабжена контрольными вопросами и задачами для самостоятельного решения, которые позволяют проверить качество усвоения материала.

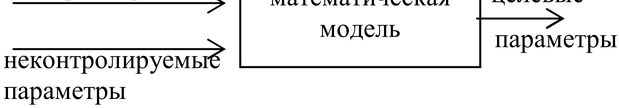
ВВЕДЕНИЕ

Операцией называется любое целенаправленное действие. Следовательно, исследование операций — это исследование целенаправленных действий. Такие исследования, очевидно, существуют столько же, сколько существует разумный человек. Для относительно простых операций исследования можно произвести без использования серьезных научных методов. Однако при планировании сложных операций не обойтись без проведения расчетов, опирающихся на *математические модели* тех или иных фрагментов операции. Для создания этих математических моделей необходим ряд организационных мероприятий и экспериментальных исследований. Для многих типичных операций математические модели построены. Они составляют математическую основу исследования операций, которой и посвящено настоящее пособие.

Под математической моделью явления или процесса понимают совокупность формул, уравнений, неравенств и т.д., отражающую существенные черты этого явления или процесса. При создании математических моделей для исследования операций первоочередной задачей является определение параметров, однозначно описывающих операцию. Эти параметры обычно подразделяют на контролируемые, неконтролируемые и целевые.

Контролируемые параметры $x_1, x_2, \dots, x_n$ являются переменными, принимающими чаще всего числовые значения, причем исследователь может придавать им эти значения по своему усмотрению. *Неконтролируемые параметры* нельзя менять по своему усмотрению; более того, их значения во многих случаях исследователю неизвестны. Они могут быть случайными величинами с известными или неизвестными вероятностными характеристиками. Наконец, *целевые параметры* характеризуют эффективность операции.

Математическая модель операции должна связывать контролируемые и неконтролируемые параметры с целевыми параметрами. Задача нахождения по контролируемым и неконтролируемым параметрам целевых параметров называется *прямой задачей* исследования операций. Математическая модель здесь используется по следующей схеме:



Наряду с прямой задачей часто решают *обратную задачу*, в которой требуется определить такие значения контролируемых параметров, при которых целевые параметры удовлетворяют тем или иным условиям оптимальности. Такие задачи часто еще называют *задачами оптимизации*. Решение подобной задачи позволяет спроектировать операцию наиболее эффективным, выгодным образом. В зависимости от разновидности и сложности математической модели применяют те или иные методы решения обратной задачи, то есть те или иные методы оптимизации. Отметим, что для некоторых разновидностей моделей методы оптимизации еще не разработаны, и их используют только для решения прямой задачи (модели систем массового обслуживания и т.д.).

В настоящее время существует множество математических моделей, используемых для исследования экономических и других видов операций. Они подразделяются на детерминированные и стохастические.

В детерминированных моделях неконтролируемые параметры, как правило, отсутствуют, а по значениям контролируемых параметров целевые параметры определяются математической моделью однозначно. К детерминированным моделям относятся модели линейного и нелинейного программирования, модели оптимального управления и т.д. Методы оптимизации здесь разработаны достаточно хорошо.

В стохастических моделях связь между контролируемыми и целевыми параметрами носит вероятностный характер. К таким моделям относятся модели теории игр, теории статистических решений, теории массового обслуживания и т.д.

Целью настоящего пособия является краткое изложение математического содержания наиболее часто используемых моделей исследования операций и их возможного применения в экономике и инженерном деле. Основное внимание уделяется методам и алгоритмам оптимизации для этих моделей.

1. ОСНОВНЫЕ ПОНЯТИЯ И МЕТОДЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В 1939 году Леонид Витальевич Канторович опубликовал работу «Математические методы организации и планирования производства», в которой сформулировал новый класс экстремальных задач с ограничениями и разработал эффективный метод их решения. Тем самым были заложены основы линейного программирования. Джордж Данциг разработал симплекс метод и считается «отцом линейного программирования» на западе. Слово программирование здесь означает составление оптимального плана (программы) производства.

1.1. Формулировка задач линейного программирования Основные формы линейных моделей

Сущность линейного программирования состоит в нахождении точек наибольшего или наименьшего значения некоторой линейной функции z при определенном наборе линейных ограничений, налагаемых на аргументы. Ограничения образуют *систему ограничений*, которая имеет, как правило, бесконечное множество решений. Каждая совокупность значений переменных (аргументов функции z), которые удовлетворяют системе ограничений, называется *допустимым планом* задачи линейного программирования. Функция z, максимум или минимум которой определяется, называется *целевой функцией задачи*. Допустимый план, на котором достигается максимум или минимум функции z, называется *оптимальным планом задачи*. Система ограничений, определяющая множество планов, диктуется условиями производства. Задачей линейного программирования является выбор из множества допустимых планов наиболее выгодного (оптимального).

Пример 1.1. Мастерская имеет два станка S_1, S_2, на которых последовательно обрабатывается два вида продукции P_1 и P_2. Станок S_1 обрабатывает единицу продукции P_1 за 1 час, а единицу продукции P_2 — за 2 часа. Станок S_2 затрачивает на единицу продукции P_1 2 часа, а на единицу продукции P_2 — 1 час. Станок S_1 может работать в сутки не более 10 часов, а станок S_2 — не более 8 часов. Стоимость единицы продукции P_1 составляет c_1 рублей, а стоимость единицы продукции P_2 — c_2 рублей. Требуется определить такой суточный план выпуска двух видов продукции P_1 и P_2 мастерской, чтобы выручка от реализации произведенной продукции была максимальна.

Обозначим через x_1 и x_2 количество продукции P_1 и P_2 соответственно, которые планируется произвести на станках S_1 и S_2. Стоимость произведенной продукции $z = c_1x_1 + c_2x_2$. Мы должны назначить x_1 и x_2 так, чтобы величина z была максимальной. Переменные x_1 и x_2 не могут принимать произвольных значений. Их значения ограничены условиями производства, а именно, тем, что станки могут работать ограниченное время. На изготовление продукции P_1 станок S_1 тратит $1 \cdot x_1$ часов, а на изготовление продукции P_2 — $2 \cdot x_2$ часов. Поскольку время работы станка S_1 не превосходит 10 часов, то величины x_1 и x_2 должны удовлетворять неравенству $x_1 + 2 \cdot x_2 \leq 10$. Аналогично можно получить неравенство для станка S_2: $2 \cdot x_1 + x_2 \leq 8$. Кроме того, величины x_1 и x_2 не могут быть отрицательными: $x_1 \geq 0$, $x_2 \geq 0$.

Таким образом, задача заключается в нахождении точек максимума функции z среди точек с координатами x_1 и x_2, которые удовлетворяют указанным неравенствам. Такие задачи кратко записываются следующим образом:

$$z = c_1 x_1 + c_2 x_2 \to \max;$$

$$\begin{cases} x_1 + 2x_2 \leq 10, \\ 2x_2 + x_1 \leq 8; \end{cases}$$

$$x_1 \geq 0;\ x_2 \geq 0.$$

Рассмотрим несколько экономико-математических моделей, допускающих формулировки в виде задач линейного программирования.

Задача об оптимальном составе смеси. Эта задача является одной из первых задач, решенных методами линейного программирования. Изложим ее как задачу определения оптимального рациона кормления скота. Пусть нам известно содержание необходимых для кормления животного питательных веществ в различных применяемых кормах. Известна также стоимость единицы каждого вида корма. Требуется составить рацион — набор и количество кормов — так, чтобы каждое питательное вещество содержалось в нем в необходимом количестве и чтобы суммарные расходы на этот рацион были минимальны. Введем следующие обозначения:

M — число различных питательных веществ (витаминов, белков, углеводов и т. д.);

N — число применяемых видов кормов (силос, сено и т. д.);

a_{ik} — количество единиц i-го питательного вещества, содержащегося в единице k-го вида кормов;

b_i — минимальная суточная потребность животного в i-ом питательном веществе;

c_k — стоимость единицы k-го вида корма;

x_k — количество единиц k-го вида корма, используемое в суточном рационе и подлежащее определению.

Оптимальным планом здесь являются такие числа x_1, x_2, …, x_n, которые удовлетворяют следующим ограничениям:

$x_1 \geq 0, x_2 \geq 0, \ldots, x_n \geq 0$ (количество корма не может быть отрицательным),

$\sum\limits_{k=1}^{n} a_{ik} x_k \geq b_i$, $i = 1, 2, \ldots, m$ (общее количество i-го питательного вещества в рационе не может быть ниже заданного b_i), и минимизируют суммарную стоимость составляемого рациона $z = \sum\limits_{k=1}^{n} c_k x_k \to \min$.

Отметим, что такая же математическая задача возникает при отыскании оптимального *состава шихты* в металлургическом производстве. Известно, что для получения легированной стали нужно использовать шихту определенного химического состава. В состав шихты входят весьма дорогостоящие ингредиенты. В то же время можно использовать и малоценные материалы (чугун, лом, отходы с определенным содержанием присадок). Возникает задача отыскания такого состава шихты, который содержал бы необходимые количества химических веществ и имел бы минимальную стоимость. Математическая модель этой задачи совпадает с сформулированной выше. Еще один пример возникновения такой модели дает задача об оптимальном *смешивании нефтепродуктов*. Бензины разных сортов получают путем смешивания нефтепродуктов, имеющих различные технические характеристики. Заданные показатели качества бензина должны выдерживаться весьма точно. С другой стороны, от того, какие нефтепродукты смешиваются и в каком количестве, зависит рентабельность производства. Требуется построить такой план смешивания, который обеспечивал бы максимальную рентабельность производства и позволял, в то же время, получать бензины заданных сортов.

Задача об оптимальном использовании ресурсов. Эта задача возникает при составлении планов выпуска продукции предприятием и имеет важное практическое значение. Пусть номенклатура выпускаемой продукции состоит из n наименований. Производство продукции использует m видов ресурсов. Обозначим через a_{ik} затраты i-го вида ресурсов (i= 1, 2, …, m) на производство единицы продукции k-го вида (k = 1, 2, …, n). Пусть b_i — полный объем ресурсов i-го вида (i = 1, 2, …, m), а c_k — прибыль, получаемая предприятием при изготовлении и реализации единицы k-го вида продукции (k = 1, 2, …, n). Требуется составить такой план выпуска продукции, который был бы технологически осуществим по имеющимся ресурсам и, в то же время, приносил наибольшую общую прибыль предприятию. Математическая модель задачи состоит в следующем. Обозначим через $X = \{x_1, x_2, \ldots, x_n\}$ номенклатурный план выпуска продукции. Величины x_k должны удовлетворять следующим условиям:

$x_1 \geq 0$, $x_2 \geq 0$, …, $x_n \geq 0$ (количество производимой продукции не может быть отрицательным),

$\sum_{k=1}^{n} a_{ik} x_k \leq b_i$, i=1, 2, …, m (технологические ограничения).

Кроме того, общая прибыль от производства и реализации продукции должна быть максимальной

$$z = \sum_{k=1}^{n} c_k x_k \to \max.$$

Приведение примеров математических моделей задач планирования и управления можно продолжить. Ниже мы еще вернемся к некоторым из таких моделей. Сейчас обсудим форму возникающих задач независимо от их содержания. Все задачи были задачами на минимум или максимум линейной целевой функции с ограничениями неравенствами. Переменные подчинялись условиям неотрицательности.

В других ситуациях могут возникать задачи, в систему ограничений которых, кроме неравенств, могут входить и равенства. Поэтому в наиболее общей форме задачу линейного программирования формулируют следующим образом:

$$z = c_0 + c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \to \max(\min);$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \ R_1 \ b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \ R_2 \ b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \ R_m \ b_m, \end{cases}$$

где R_j (j=1, 2,…, m)— один из знаков сравнения ($\leq$, $\geq$, =).

Одна и та же задача линейного программирования может быть сформулирована в различных эквивалентных формах. Наиболее важными формами задачи линейного программирования являются *каноническая* и *стандартная*.

В *канонической форме* задача является задачей на максимум некоторой линейной функции z, а ее система ограничений состоит только из равенств (уравнений); при этом переменные задачи $x_1, x_2, ..., x_n$ являются неотрицательными:

$$z = c_0 + c_1x_1 + c_2x_2 + ... + c_nx_n \to \max;$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2, \\ .... \\ a_{1m}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m; \end{cases}$$

$$x_1 \ge 0, x_2 \ge 0, ..., x_n \ge 0.$$

К канонической форме можно привести любую задачу линейного программирования. Если в исходной задаче некоторое ограничение (например, первое) было неравенством, то оно преобразуется в равенство введением в левую часть некоторой неотрицательной переменной. Действительно

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \le b_1.$$

Вводим переменную $x_{n+1} = b_1 - a_{11}x_1 - a_{12}x_2 - ... - a_{1n}x_n$. Тогда неравенство запишется в виде

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n + x_{n+1} = b_1.$$

В каждое из неравенств вводится своя “уравнивающая” переменная, после чего система ограничений становится системой уравнений. При этом “уравнивающая” переменная автоматически будет неотрицательной. Если в исходной задаче некоторая переменная x_i не подчинена условию неотрицательности, то ее заменяют разностью неотрицательных переменных x_i' и x_i'' :

$$x_i = x_i'' - x_i'; \; x_i', x_i'' \ge 0 .$$

Наконец, если исходная задача была задачей на минимум, то введением новой целевой функции $z_1 = -z$ мы преобразуем нашу задачу на минимум функции z в задачу на максимум функции z_1.

Таким образом, всякую задачу линейного программирования можно сформулировать в канонической форме.

В *стандартной форме* задача линейного программирования является задачей на максимум линейной целевой функции. Система ограничений ее состоит из линейных неравенств типа ≤. Все переменные задачи неотрицательны.

$$z = c_0 + c_1x_1 + c_2x_2 + ... + c_nx_n \to \max;$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m; \end{cases}$$

$$x_1 \ge 0,\ x_2 \ge 0,\ \ldots,\ x_n \ge 0.$$

Всякую задачу линейного программирования можно сформулировать в стандартной форме. Преобразование задачи на минимум в задачу на максимум, а также обеспечение неотрицательности переменных производится, как и раньше. Всякое равенство в системе ограничений равносильно системе взаимно противоположных неравенств:

$$a_{i1}x_i + a_{i2}x_2 + \ldots + a_{in}x_n = b_i \Leftrightarrow \begin{cases} a_{i1}x_i + a_{i2}x_2 + \ldots + a_{in}x_n \le b_i, \\ -a_{i1}x_i + a_{i2}x_2 - \ldots - a_{in}x_n \le -b_i. \end{cases}$$

Существуют и другие способы преобразования системы равенств в систему неравенств, то есть всякую задачу линейного программирования можно сформулировать в стандартной форме.

Рассмотренные выше записи задач линейного программирования в канонической и стандартной формах обычно называют скалярными записями. Иногда более удобной является векторно-матричная запись. Введем векторы

$$\vec{x} = \{x_1, x_2, \ldots, x_n\}, \quad \vec{c} = \{c_1, c_2, \ldots, c_n\}, \quad \vec{b} = \{b_1, b_2, \ldots, b_m\}$$

и матрицу $A = (a_{ij})$ коэффициентов в системе ограничений. Тогда задачи в канонической и стандартной формах можно записать соответственно в виде

$$z = c_0 + (\vec{c}, \vec{x}) \to \max, \qquad\qquad z = c_0 + (\vec{c}, \vec{x}) \to \max,$$
$$A\vec{x} = \vec{b},\ \vec{x} \ge \vec{0}. \qquad\qquad A\vec{x} \le \vec{b},\ \vec{x} \ge \vec{0}.$$

Здесь и ниже неравенство $\vec{a} \le \vec{e}$ $(\vec{a} < \vec{e})$ между векторами одинаковой размерности означает, что выполнены неравенства между всеми соответствующими координатами этих векторов. При этом условия неотрицательности (положительности) переменных часто записывают просто в виде $\vec{x} \ge 0$ $(\vec{x} > 0)$.

В заключение параграфа отметим, что произвольная задача линейного программирования может не иметь решения. Эта ситуация может возникнуть в двух случаях: 1) система ограничений несовместна (область допустимых планов пуста); 2) целевая функция задачи неограниченна на области допустимых планов. В первом случае задачу называют *недопустимой*.

1.2. Геометрическое истолкование задачи в стандартной форме в случае двух переменных

Задача линейного программирования в стандартной форме с двумя перемененными имеет вид:

$$z = c_0 + c_1 x_1 + c_2 x_2 \to \max;$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \le b_1, \\ a_{21}x_1 + a_{22}x_2 \le b_2, \\ \dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 \le b_m; \end{cases}$$

$$x_1, x_2 \ge 0.$$

Эти задачи допускают простое геометрическое истолкование. Рассмотрим вначале геометрическое истолкование системы ограничений задачи. Каждую совокупность значений переменных x_1, x_2 можно изобразить точкой на плоскости, если ввести систему координат и по одной оси откладывать x_1, а по другой — x_2. Выясним, что геометрически означает совокупность решений одного отдельно взятого неравенства

$$a_1 x_1 + a_2 x_2 \le b.$$

Рассмотрим прямую линию на плоскости с уравнением

$$a_1 x_1 + a_2 x_2 = b.$$

Эта прямая делит плоскость на две полуплоскости, в одной из которых справедливо наше неравенство, а в другой — противоположное неравенство. Для того чтобы проверить, какая из полуплоскостей состоит из решений нашего неравенства, следует взять точку из какой-либо полуплоскости и проверить, выполняется ли наше неравенство в этой точке. Множество решений отдельно взятого линейного неравенства представляет собой полуплоскость. Для системы из нескольких таких неравенств точки, координаты которых удовлетворяют всем неравенствам одновременно, должны находиться во всех соответствующих полуплоскостях, то есть принадлежать теоретико-множественному пересечению этих полуплоскостей. Множество точек на плоскости, удовлетворяющих системе ограничений, составляет, таким образом, некоторую выпуклую многоугольную область. Условия неотрицательности переменных x_1, x_2 ≥ 0 приводят к тому, что эта область находится в первой координатной четверти.

Пример 1.2. Построим на плоскости множество решений системы неравенств (рис.1.1)

$$\begin{cases} x_1 + 2x_2 \leq 10, \\ 2x_1 + x_2 \leq 8; \end{cases}$$

$$x_1 \geq 0;\, x_2 \geq 0.$$

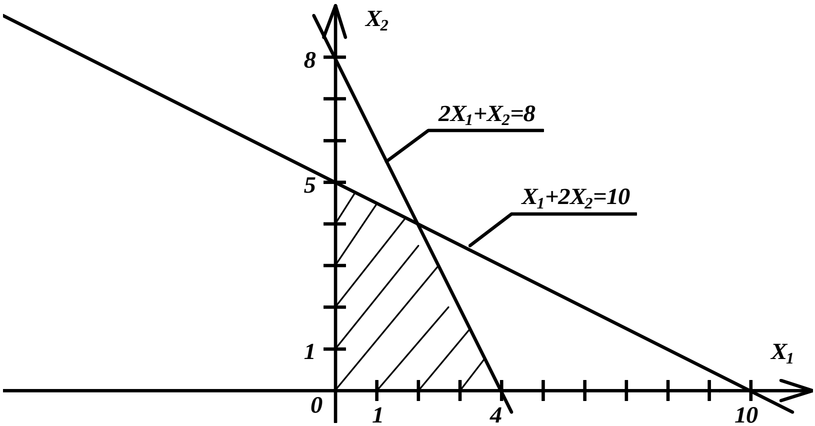


Рис. 1.1. Геометрическое истолкование системы ограничений.

Множество точек на плоскости, удовлетворяющих системе ограничений задачи, называют часто допустимым многоугольником. Эта область может быть неограниченной или вовсе пустой.

Целевая функция задачи геометрически изображается с помощью прямой уровня, то есть прямой, на которой

$$z = c_0 + c_1 x_1 + c_2 x_2$$

принимает постоянное значение. Если C — произвольная константа, то уравнение прямой уровня имеет вид $c_0 + c_1 x_1 + c_2 x_2 = C$.

При изменении константы C мы получаем различные прямые, параллельные друг другу. При увеличении C прямая уровня перемещается в направлении наискорейшего возрастания функции z, то есть, в направлении ее градиента. Вектор градиента

$$grad\, \mathrm{z} = \left\{ \frac{\partial\, \mathrm{z}}{\partial\, x_1}; \frac{\partial\, \mathrm{z}}{\partial\, x_2} \right\} = \{c_1, c_2\}.$$

Геометрический метод решения задачи состоит в следующем. Строится допустимый многоугольник и некоторое положение линии уровня целевой функции. Определяется направление перемещения прямой уровня. Точкой минимума z будет точка первого касания линии уровня с допустимым многоугольником. Точкой максимума — точка отрыва линии уровня от допустимого многоугольника. Эти точки чаще всего совпадают с некоторыми вершинами допустимого многоугольника. Таких точек может быть бесчисленное множество, если линия уровня z параллельна одной из сторон допустимого многоугольника.

Пример 1.3. Решим геометрически задачу примера 1.1 при $c_1 = 1$, $c_2 = 1$ (рис. 1.2).

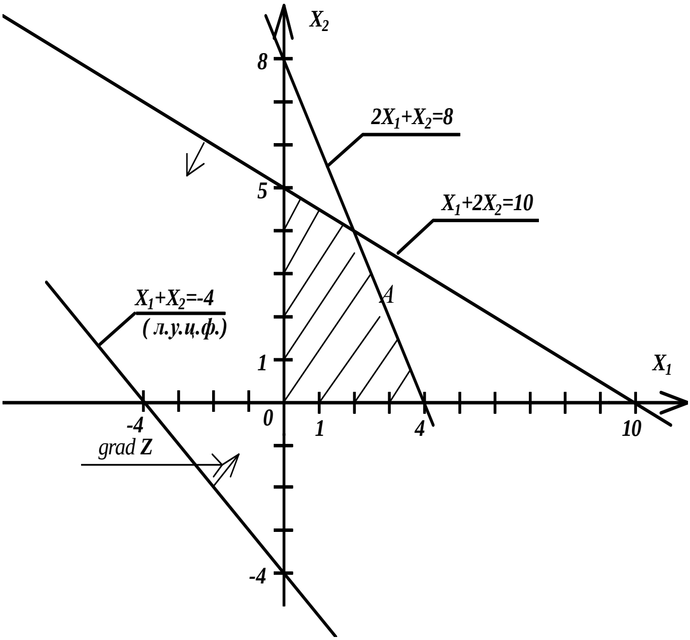


Рис 1.2. Графическое решение задачи.

Точкой максимума здесь является точка A, координаты которой определяются из следующей системы уравнений:

$$\begin{cases} x_1 + 2x_2 = 10, \\ 2x_1 + x_2 = 8. \end{cases}$$

Решая эту систему, получаем точку максимума $A(2,4)$, $z_{\max} = 6$.

1.3. Общие системы линейных уравнений. Базисный вид системы. Метод Гаусса-Жордана

Ниже нам понадобятся некоторые сведения о системах уравнений, состоящих из m линейных уравнений с n неизвестными $x_1, x_2, ..., x_n$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m. \end{cases}$$

Такая система может иметь множество решений, состоящее из единственного решения или бесконечного количества решений. Возможен также случай *несовместности системы*, когда множество решений является пустым. Две системы называются *эквивалентными*, если они имеют одинаковые множества решений. Следующие операции над системой приводят к новой системе, эквивалентной исходной:

1) перестановка уравнений системы;
2) умножение обеих частей уравнения системы на одно и то же отличное от нуля число;
3) прибавление к заданному уравнению системы любого другого уравнения, умноженного на произвольное число.

Эти операции называют элементарными операциями над системой.

Основной задачей теории линейных систем является задача нахождения всего множества решений системы или установления ее несовместности. Для ее решения нужно привести систему к эквивалентной системе, имеющей особый *базисный* вид. При этом систему условно записывают в виде расширенной матрицы системы

$$\left(\begin{array}{ccc|c} a_{11} & a_{12}\cdots & a_{1n} & b_1 \\ a_{21} & a_{22}\cdots & a_{2n} & b_2 \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2}\cdots & a_{mn} & b_m \end{array}\right).$$

Система называется имеющей базисный вид, если среди столбцов коэффициентов при неизвестных в ее расширенной матрице имеется столько различных единичных столбцов, сколько ненулевых строк в этой матрице. Единичным столбцом мы называем столбец, в котором на некотором месте стоит единица, а на всех остальных местах — нули. Единичные столбцы считаются различными, если единицы у них находятся на различных местах. Неизвестные, отвечающие этим различным единичным столбцам, называются *базисными* *неизвестными системы*. Остальные неизвестные называются *свободными неизвестными*. Базисные неизвестные входят по одному в каждое из уравнений и легко выражаются через свободные неизвестные. Если свободным неизвестным придавать произвольные значения, то базисные неизвестные по ним определятся однозначно, и мы получим все решения системы.

Пример 1.4. Рассмотрим систему

$$\begin{cases} x_1 + 3x_2 + 5x_3 = 6, \\ 7x_2 - 2x_3 + x_4 = -1. \end{cases}$$

Ее расширенная матрица

$$\left(\begin{array}{cccc|c} 1 & 3 & 5 & 0 & 6 \\ 0 & 7 & -2 & 1 & -1 \end{array}\right).$$

Система имеет базисный вид. Базисными неизвестными будут x_1 и x_4, а свободными — x_2 и x_3.

$$\begin{cases} x_1 = 6 - 3x_2 - 5x_3, \\ x_4 = -1 - 7x_2 + 2x_3. \end{cases}$$

Полагая, свободные неизвестные произвольными $x_2 = c_1$, $x_3 = c_2$, получаем двухпараметрическое семейство всех решений нашей системы $(6 - 3c_1 - 5c_2; c_1; c_2; -1 - 7c_1 + 2c_2)$.

Свободные неизвестные могут отсутствовать в базисном виде системы. Тогда, очевидно, система имеет только одно решение.

Метод Гаусса-Жордана представляет собой некоторый алгоритм, приводящий систему к базисному виду с помощью цепочки элементарных преобразований, которую удобно выполнять не над системой, а над ее расширенной матрицей. При этом элементарные операции над системой становятся следующими операциями над расширенной матрицей:

1) перестановка строк в матрице;
2) умножение всех элементов некоторой строки на одно и то же отличное от нуля число;
3) прибавление к данной строке любой другой, умноженной на произвольное число.

Метод Гаусса-Жордана состоит из ряда однотипных шагов. Опишем первый шаг алгоритма. Он состоит из трех этапов:

1) среди коэффициентов при неизвестных расширенной матрицы системы выбирается отличное от нуля число, которое в дальнейшем мы называем *разрешающим элементом* шага метода;

2) строка разрешающего элемента делится на разрешающий элемент и полученная строка, становясь основным инструментом для преобразования матрицы, называется нами в дальнейшем *ведущей строкой* шага алгоритма;

3) ведущая строка преобразует остальные строки матрицы путем прибавления ее к этим строкам после умножения на так подобранные числа, чтобы после преобразований в столбце бывшего разрешающего элемента стояли нули на всех местах, кроме места самого разрешающего элемента (на котором находится единица).

Описанные преобразования являются цепочкой элементарных операций над расширенной матрицей системы, и после завершения шага алгоритма метода Гаусса-Жордана в матрице появляется единичный столбец. Затем шаги повторяются, но на очередном шаге запрещается выбирать разрешающий элемент в строках, в которых он уже выбирался на предыдущих шагах. Шаги продолжаются до тех пор, пока количество единичных столбцов не сравняется с количеством ненулевых строк расширенной матрицы. Мы получаем систему в базисном виде.

При работе методом Гаусса-Жордана возможны следующие две особые ситуации. В результате выполнения очередного шага могут появиться либо нулевая строка $(0, 0, \ldots, 0 \,|\, 0)$, либо строка вида $(0, 0, \ldots, 0 \,|\, b)$, где $b \neq 0$. В первом случае в новой системе будет уравнение вида $0 \cdot x_1 + 0 \cdot x_2 + \ldots + 0 \cdot x_n = 0$, которое является тождеством, справедливым при любых значениях неизвестных. Отбрасывание этого уравнения не меняет множества решений системы, поэтому обычно нулевая строка отбрасывается, и работа алгоритма продолжается. Во втором случае в новой системе появляется уравнение $0 \cdot x_1 + 0 \cdot x_2 + \ldots + 0 \cdot x_n = b \neq 0$, которое не может выполняться. Это свидетельствует о том, что новая и первоначальная системы несовместны. В этом случае работа алгоритма прекращается.

Пример 1.5. Реализуем метод Гаусса-Жордана для системы

$$\begin{cases} x_1 + 3x_2 + 5x_3 = 2, \\ 3x_1 - 2x_2 + 16x_3 = -3. \end{cases}$$

В расширенной матрице системы выберем разрешающий элемент в первой строке и первом столбце. Получим сразу ведущую строку. Умножая первую строку на (–3) и прибавляя ко второй, получим

$$\left(\begin{array}{ccc|c} \boxed{1} & 3 & 5 & 2 \\ 3 & -2 & 16 & -3 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & 3 & 5 & 2 \\ 0 & -11 & 1 & -9 \end{array}\right).$$

Выбирая разрешающий элемент во второй строке и третьем столбце, умножая вторую строку на (–5) и прибавляя к первой, получим

$$\left(\begin{array}{ccc|c} 1 & 3 & 5 & 2 \\ 0 & -11 & \boxed{1} & -9 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & 58 & 0 & 47 \\ 0 & -11 & 1 & -9 \end{array}\right).$$

Последняя матрица соответствует системе уравнений

$$\begin{cases} x_1 + \quad 58x_2 \quad = \quad 47, \\ -11x_2 + \quad x_3 \quad = \quad -9, \end{cases}$$

с базисными неизвестными x_1, x_3 и свободной неизвестной x_2.

Отметим, что рассмотренный нами метод содержит большую долю произвола при выборе разрешающего элемента. Полученный в результате базисный вид системы тоже определяется неоднозначно. У совместной системы существует некоторая конечная совокупность базисных видов.

Переход от одного базисного вида к другому можно произвести с помощью, так называемой, *операции замещения*. Эта операция переводит заданную базисную неизвестную x_i в разряд свободных, а заданную свободную неизвестную x_j — в базисную. Операция замещения состоит в дополнительном шаге алгоритма Гаусса-Жордана с особым выбором разрешающего элемента. Этот элемент выбирается в строке, содержащей единицу при базисной неизвестной x_i и в столбце, отвечающем свободной неизвестной x_j. Выполним операцию замещения в базисном виде системы предыдущего примера, замещая свободной переменной x_2 базисную x_1

$$\left(\begin{array}{ccc|c} 1 & \boxed{58} & 0 & 47 \\ 0 & -11 & 1 & -9 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1/58 & \boxed{1} & 0 & 47/58 \\ 0 & -11 & 1 & -9 \end{array}\right) \sim$$

$$\left(\begin{array}{ccc|c} 1/58 & 1 & 0 & 47/58 \\ 11/58 & 0 & 1 & -5/58 \end{array}\right).$$

Тем самым получен новый базисный вид системы

$$\begin{cases} \dfrac{1}{58}x_1 + \quad x_2 \quad = \quad \dfrac{47}{58}, \\ \dfrac{11}{58}x_1 + \quad x_3 \quad = \quad -\dfrac{5}{58}. \end{cases}$$

В следующем параграфе при изучении симплекс-метода мы встретимся с базисным видом линейной системы уравнений и операцией замещения. При этом неизвестные будут называться переменными.

1.4. Симплекс-метод

Этот метод является универсальным, применимым к любой задаче линейного программирования в канонической форме. Система ограничений здесь — система линейных уравнений, в которой количество неизвестных обычно больше количества уравнений. Если ранг системы равен r, то мы можем выбрать r неизвестных, которые выразим через все остальные неизвестные. Для определенности предположим, что выбраны первые, идущие подряд, неизвестные $x_1, x_2, ..., x_r$. Тогда наша система уравнений может быть записана в виде

$$\begin{cases} x_1 = b_1' - a_{1,r+1}' x_{r+1} - ... - a_{1n}' x_n, \\ x_2 = b_2' - a_{2,r+1}' x_{r+1} - ... - a_{2n}' x_n, \\ \quad\quad \dots\dots\dots\dots\dots\dots \\ x_r = b_r' - a_{r,r+1}' x_{r+1} - ... - a_{rn}' x_n; \end{cases}$$

К такому виду можно привести любую совместную систему, например методом Гаусса-Жордана. Правда, не всегда можно выражать через остальные первые r неизвестных (мы это сделали для определенности записи). Однако такие r неизвестных обязательно найдутся. Эти неизвестные (переменные) называются *базисными*. Остальные переменные называются *свободными*. Придавая определенные значения свободным переменным и вычисляя значения базисных (выраженных через свободные), мы будем получать различные решения нашей системы ограничений. Таким образом, можно получить любое ее решение. Нас будут интересовать особые решения, которые получаются, когда свободные переменные равны нулю. Такие решения называются *базисными*. Базисных решений столько же, сколько различных базисных видов у данной системы ограничений. Базисное решение называется *допустимым базисным решением* или *опорным решением*, если в нем значения переменных неотрицательны. Если в качестве базисных взяты переменные $x_1, x_2, ..., x_r$, то решение $\{b_1', b_2', ..., b_r', 0, ..., 0\}$ будет опорным, если $b_1' \geq 0,\ b_2' \geq 0,\ ..., b_r' \geq 0.$ Симплекс-метод основан на следующей теореме, которая называется *фундаментальной теоремой симплекс-метода*.

Теорема 1.1. *Среди оптимальных планов задачи линейного программирования в канонической форме обязательно есть опорное решение ее системы ограничений. Если оптимальный план задачи единственен, то он совпадает с некоторым опорным решением.*

Различных опорных решений системы ограничений конечное число. Поэтому решение задачи в канонической форме можно было бы искать перебором опорных решений и выбором среди них того решения, для которого значение z самое большое. Но, во-первых, все опорные решения неизвестны, и их нужно находить, а во-вторых, в реальных задачах этих решений очень много, и прямой перебор вряд ли возможен. Симплекс-метод представляет собой некоторую процедуру направленного перебора опорных решений. Исходя из некоторого, найденного заранее, опорного решения по определенному алгоритму симплекс-метода, мы подсчитываем новое опорное решение, на котором значение целевой функции z не меньше, чем на старом. После ряда шагов мы приходим к опорному решению, которое является оптимальным планом.

Процедура симплекс-метода на примере

Пусть требуется найти решение следующей задачи линейного программирования

$$z = -x_4 + x_5 \to \max,$$

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

$$x_1, x_2, \dots, x_5 \geq 0.$$

Первое опорное решение имеет следующий вид $B_1=\{1;\ 2;\ 3;\ 0;\ 0\}$. На этом базисном решении системы ограничений целевая функция $z(B_1) = 0$. Из вида целевой функции заключаем, что она может быть увеличена при выходе из решения B_1 путем увеличении переменной x_5 от нуля. При этом нужно следить, чтобы соблюдались равенства нашей системы и все переменные оставались неотрицательными. Из первого ограничения видно, что x_1 останется неотрицательным при произвольном увеличении x_5. Второе ограничение показывает, что x_2 становиться отрицательным при $x_5 > 2$. Из третьего ограничения заключаем, что x_3 остается неотрицательным при увеличении x_5 до трех. Таким образом, все переменные остаются неотрицательными при увеличении x_5 до 2. Предположим, что $x_5 =2$, но, по-прежнему, $x_4= 0$. Тогда остальные переменные примут значения $x_1= 5$; $x_2= 0$; $x_3 = 1$. Мы перешли к некоторому новому решению системы ограничений $B_2=\{5;\ 0;\ 1;\ 0;\ 2\}$. Это решение будет опорным решением и соответствующий базисный вид можно получить с помощью операции замещения, при которой x_2 выводится из числа базисных, а x_5 становится базисной переменной. Другими словами, нужно x_5 выразить из второго равенства системы ограничений и полученное выражение подставить вместо x_5 в первое и третье равенства. В результате получим базисный вид системы

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

которому отвечает базисное решение B_2={5; 0; 1; 0; 2}. С помощью нового вида системы исключим x_5 из целевой функции задачи

$$z = -x_4 + x_5 = 2 - x_2 + 2x_4 - x_4 = 2 - x_2 + x_4.$$

Значение целевой функции на новом базисном решении $z(B_2) = 2$. При этом целевую функцию можно еще увеличить, если выйти из B_2, увеличивая переменную x_4. В первом и во втором равенствах перестроенной системы ограничений x_4 можно увеличивать неограниченно. В третьем равенстве x_4 можно увеличивать лишь до 1/5. В противном случае переменная x_3 станет отрицательной. Положим $x_4 = 1/5$, $x_2 = 0$. Тогда $x_1 = 5 + 3/5 = 28/5$; $x_5 = 2 + 2 / 5 = 12 / 5$; $x_3 = 0$. Получаем следующее опорное решение B_3={28/5; 0; 0; 1/5; 12/5}. При этом переменная x_3 должна быть выведена из состава базисных переменных, а вместо нее базисной переменной должна стать переменная x_4. Производя операцию замещения, получаем следующий базисный вид системы ограничений

$$\begin{cases} x_1 = (28/5) - (7/5)x_2 - (3/5)x_3, \\ x_5 = (12/5) - (3/5)x_2 - (2/5)x_3, \\ x_4 = (1/5) + (1/5)x_2 - (1/5)x_3, \end{cases}$$

с помощью которого можно исключить базисные переменные из выражения для целевой функции

$$z = 2 - x_2 + x_4 = 2 - x_2 + ((1/5) + (1/5)x_2 - (1/5)x_3) = (11/5) - (4/5)x_2 - (1/5)x_3.$$

Из последнего выражения видно, что увеличить значение целевой функции переходом к новому опорному решению нельзя. Поэтому $z_{max} = z(B_3) = 11/5$. Оптимальный план совпадает с B_3 = {28/5; 0; 0; 1/5; 12/5}.

Процедура симплекс-метода в общем случае

В рассмотренном примере мы сделали два шага, переходя последовательно от базисного решения B_1 к B_2, а затем — к B_3. Вычисления по симплекс-методу обычно организуются в виде так называемых симплекс-таблиц. Чтобы разобраться в устройстве симплекс-таблицы рассмотрим один шаг симплекс-метода в общем случае. Предположим, что система ограничений задачи в канонической форме приведена к допустимому базисному виду и базисными переменными являются x_1, x_2, …, x_r. Целевая функция при этом выражена через свободные переменные x_{r+1}, …, x_n, то есть задача имеет вид

$$z = \gamma_0 + \gamma_{r+1}x_{r+1} + \ldots + \gamma_n x_n \to \max,$$

$$\begin{cases} x_1 = b_1 - (a_{1r+1}x_{r+1} + \ldots + a_{1n}x_n), \\ \ldots \ldots \ldots \ldots \quad \ldots \quad \ldots \ldots \quad \ldots \\ x_r = b_r - (a_{rr+1}x_{r+1} + \ldots + a_{rn}x_n), \end{cases}$$

$$x_i \geq 0, \quad i = 1, 2, \ldots, n.$$

Поскольку базисный вид является допустимым, то b_1, b_2 …, $b_r \geq 0$. Работа по симплекс-методу начинается с просмотра коэффициентов целевой функции, то есть величин γ_{r+1}, …, γ_n. Здесь могут представиться два случая.

1) Все коэффициенты неположительные: γ_{r+1}, …, $\gamma_n \leq 0$. В этом случае исходное базисное решение будет оптимальным, так как переходом к другому базисному решению мы не можем увеличить целевую функцию.

2) Среди коэффициентов целевой функции есть положительные.

Пусть $\gamma_j > 0$. Это значит, что увеличение переменной x_j ведет к увеличению целевой функции. При этом увеличивать x_j нужно так, чтобы базисные переменные оставались неотрицательными. Для определения границы увеличения x_j просмотрим столбец коэффициентов при x_j в системе ограничений, то есть числа a_{1j}, a_{2j}, …, a_{rj}. Если все эти коэффициенты отрицательны, то x_j можно увеличивать неограниченно, сохраняя неотрицательными базисные переменные. Это означает, что целевая функция неограниченна на области допустимых значений и задача не имеет решений. Если же среди этих коэффициентов есть положительные, то в соответствующих уравнениях x_j нельзя увеличивать неограниченно. Предположим, что мы придаем x_j значение ρ ($x_j = \rho$), а остальные свободные переменные оставляем равными нулю. Система ограничений примет вид

$$\begin{cases} x_1 = b_1 - a_{1j}\rho, \\ x_2 = b_2 - a_{2j}\rho, \\ \ldots \ldots \ldots \ldots \ldots \\ x_r = b_\rho - a_{rj}\rho. \end{cases}$$

Значение ρ не может быть произвольным в i-ом равенстве, если $a_{ij}>0$. При этом ρ можно увеличивать в i-ом равенстве до величины b_i/a_{ij}.

Таким образом, граница увеличения ρ может быть найдена выбором среди чисел a_{1j}, a_{2j}, …, a_{rj} положительных, а затем для них минимального значения отношения b_i/a_{ij}. Это отношение и даст нам границу увеличения ρ. Если минимальным будет отношение $b_{i_0}/a_{i_0 j}$, то граница увеличения ρ определяется i_0-ым равенством и можно положить $\rho = b_{i_0}/a_{i_0 j}$. При этом $x_{i_0} = 0$. При перестройке системы ограничений переменная x_j должна быть введена в число базисных переменных, а переменная x_{i_0}. должна стать свободной. Перестройка производится с помощью операции замещения, которая выполняется с помощью разрешающего элемента $a_{i_0 j}$. Этот элемент принято называть разрешающим или генеральным элементом шага симплекс-метода. После операции однократного замещения мы получим новый базисный вид системы ограничений, новое опорное решение и новое выражение для целевой функции через новые свободные переменные. На этом шаг симплекс-метода завершается. Следующий шаг начинается с просмотра коэффициентов нового выражения целевой функции. Шаги симплекс-метода повторяются до тех пор, пока на очередном шаге все коэффициенты выражения целевой функции через свободные переменные не станут неположительными. В этом случае очередное опорное решение системы ограничений будет точкой максимума.

Вычисления по симплекс-методу организуются в виде симплекс-таблиц, которые являются сокращенной записью задачи линейного программирования в канонической форме. *Перед составлением симплекс-таблицы задача должна быть преобразована. Система ограничений приводится к допустимому базисному виду, с помощью которого из целевой функции должны быть исключены базисные переменные.* Вопрос об этих преобразованиях мы рассмотрим ниже. Сейчас же будем считать, что они уже выполнены и задача имеет вид

$$z = \gamma_0 + \gamma_{r+1}x_{r+1} + \ldots + \gamma_n x_n \to \max$$

$$\begin{cases} x_1 = b_1' - a_{1,r+1}' x_{r+1} - \ldots - a_{1n}' x_n, \\ x_2 = b_2' - a_{2,r+1}' x_{r+1} - \ldots - a_{2n}' x_n, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ x_r = b_r' - a_{r,r+1}' x_{r+1} - \ldots - a_{rn}' x_n; \end{cases}$$

$$x_i \geq 0,\ i = 1, 2, \ldots, n.$$

Здесь для определенности записи считается, что в качестве базисных можно взять переменные $x_1, x_2, \ldots, x_r$ и что при этом

$b_1' \geq 0, b_2' \geq 0, \ldots, b_r' \geq 0$. (соответствующее базисное решение является опорным).

Для составления симплекс-таблицы во всех равенствах в условии задачи члены, содержащие переменные, переносятся в левую часть, а свободные члены оставляются справа, т.е. задача записывается в виде следующей системы равенств:

$$\begin{cases} x_1 + a'_{1,r+1}x_{r+1} + \ldots + a'_{1n}x_n = b'_1, \\ x_2 + a'_{2,r+1}x_{r+1} + \ldots + a'_{2n}x_n = b'_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ x_r + a'_{r,r+1}x_{r+1} + \ldots + a'_{rn}x_n = b'_r, \\ z - \gamma_{r+1}x_{r+1} - \ldots - \gamma_n x_n = \gamma_0. \end{cases}$$

Далее эта система оформляется в виде таблицы:

Баз. пер	Св. чл.	x_1	x_2	...	x_r	x_{r+1}	x_{r+2}	...	x_N
x_1	b_1	1	0	...	0	$a_{1\ r+1}$	$a_{1\ r+2}$	...	$a_{1\ N}$
x_2	b_2	0	1	...	0	$a_{2\ r+1}$	$a_{2,r+2}$	...	$a_{2\ N}$
...	...	...	...	...	...	...	...	...	...
x_r	b_r	0	0	...	1	$a_{r,\ r+1}$	$a_{r,\ r+2}$	...	$a_{r\ N}$
z	γ_0	0	0	...	0	$-\gamma_{r+1}$	$-\gamma_{r+2}$	...	$-\gamma_N$

Еще раз напомним, что названия базисных переменных здесь взяты лишь для определенности записи и в реальной таблице могут оказаться другими.

Порядок работы с симплекс-таблицей

Первая симплекс-таблица подвергается преобразованию, суть которого заключается в переходе к новому опорному решению. Порядок перехода к следующей таблице такой.

1. Просматривается последняя строка таблицы и среди коэффициентов этой строки (исключая γ_0) выбирается отрицательное число. Если такового нет, то исходное базисное решение является оптимальным и данная таблица является последней.
2. Просматривается столбец таблицы, отвечающий выбранному отрицательному коэффициенту в последней строке, и в этом столбце выбираются положительные коэффициенты. Если таковых нет, то

целевая функция неограниченна на области допустимых значений переменных, и задача решений не имеет.

3. Среди отобранных коэффициентов столбца выбирается тот, для которого отношение соответствующего свободного члена, находящегося в столбце свободных членов, к этому элементу, минимально. Этот коэффициент называется *разрешающим* или *генеральным элементом* таблицы. В дальнейшем базисная переменная, отвечающая строке разрешающего элемента, должна быть переведена в разряд свободных, а свободная переменная, отвечающая столбцу разрешающего элемента, вводится в число базисных.

4. Строится новая таблица, содержащая новые названия базисных переменных. Строка разрешающего элемента делится на этот элемент, и полученная строка записывается в новую таблицу на то же место. В остальные клетки новой таблицы записываются результаты преобразования элементов старой таблицы. Для этого умножают первую из заполненных строк (строку разрешающего элемента) на некоторые числа и складывают ее со строками старой таблицы. Числа эти подбираются так, чтобы в столбце разрешающего элемента получились нули, кроме клетки разрешающего элемента, в которой стоит единица. В результате получают новую симплекс-таблицу, которая отвечает новому базисному решению.

5. Теперь следует обратиться к пункту 2, т.е. просмотреть строку целевой функции и повторить все вышеперечисленное. Составление новых симплекс-таблиц производится до тех пор, пока все коэффициенты последней строки (кроме стоящего на месте γ_0) в очередной таблице не станут неотрицательными. После этого считается, что задача решена и по последней симплекс-таблице прочитывается ответ задачи. Максимальное значение $z_{\max}$ целевой функции стоит в первой клетке последней строки на месте γ_o. Неотрицательные значения новых базисных переменных стоят в остальных клетках столбца свободных членов. Остальные переменные в точке максимума равны нулю.

Замечание. Изложенный алгоритм содержит возможность неопределенности при выборе разрешающего элемента. Могут появиться несколько столбцов, пригодных для его выбора. Да и в заданном столбце может быть несколько чисел, каждое из которых можно назвать разрешающим элементом. Для однозначной организации вычислений приходится вводить добавочные условные правила. При выборе столбца разрешающего элемента в последней строке симплекс-таблицы выбирается максимальный по модулю отрицательный коэффициент. Если есть несколько таких коэффициентов с одинаковым максимальным модулем, выбирается тот, что отвечает переменной с минимальным номером и т.д. Отметим также, что если в столбце, пригодном для выбора разрешающего элемента, нет положительных чисел, то задача не имеет решений по причине неограниченности целевой функции на области допустимых планов.

Рассмотрим порядок решения задачи с помощью симплекс-таблиц на примерах.

Пример 1.6. Решить следующую задачу, уже рассмотренную в качестве примера:

$$z = x_5 - x_4 \to \max;$$

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

$$x_i \geq 0, \; i = 1, \dots, 5.$$

Запишем задачу в виде равенств:

$$\begin{cases} x_1 + x_4 - 2x_5 = 1, \\ x_2 - 2x_4 + x_5 = 2, \\ x_3 + 3x_4 + x_5 = 3, \\ z + x_4 - x_5 = 0. \end{cases}$$

Составляем первую симплекс-таблицу. Находим разрешающий элемент.

Базисные переменные	Свободные члены	x_1	x_2	x_3	x_4	$\downarrow x_5$
x_1	1	1	0	0	1	–2
$\leftarrow x_2$	2	0	1	0	–2	1
x_3	3	0	0	1	3	1
z	0	0	0	0	1	–1

Составляем новую симплекс-таблицу. Снова находим разрешающий элемент.

Базисные переменные	Свободные члены	x_1	x_2	x_3	$\downarrow x_4$	x_5
x_1	5	1	2	0	–3	0
x_5	2	0	1	0	–2	1
$\leftarrow x_3$	1	0	–1	1	5	0
z	2	0	1	0	–1	0

Переходим к следующей таблице:

Базисные переменные	Свободные члены	x_1	x_2	x_3	x_4	x_5
x_1	$\frac{28}{5}$	1	$\frac{7}{5}$	$\frac{3}{5}$	0	0
x_5	$\frac{12}{5}$	0	$\frac{3}{5}$	$\frac{2}{5}$	0	1
x_4	$\frac{1}{5}$	0	$-\frac{1}{5}$	$\frac{1}{5}$	1	0
z	$\frac{11}{5}$	0	$\frac{4}{5}$	$\frac{1}{5}$	0	0

Эта таблица является последней, по ней читаем ответ задачи: $z_{\max} = \frac{11}{5}$ Координаты точки максимума: $x_1 = \frac{28}{5}$; $x_2 = 0$; $x_3 = 0$; $x_4 = \frac{1}{5}$; $x_5 = \frac{12}{5}$.

Пример 1.7. Решить задачу:

$$z = x_1 + x_2 \to \max;$$

$$\begin{cases} x_3 = 1 + x_1 - x_2, \\ x_4 = 2 - x_1 + 2x_2; \end{cases}$$

$$x_i \ge 0 \ (i = 1,2,3,4).$$

Составляем первую симплекс-таблицу и находим разрешающий элемент.

	Базис.перем.	Свобод. члены	$\downarrow x_1$	x_2	x_3	x_4
	x_3	1	–1	1	1	0
	$\leftarrow x_4$	2	1	–2	0	1
	z	0	–1	–1	0	0

Вторая таблица имеет вид:

	Базис. перем.	Свобод. члены	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	$\boldsymbol{x_4}$
	$\boldsymbol{x_3}$		0	–1	1	1
	$\boldsymbol{x_1}$	2	1	–2	0	1
	$\boldsymbol{z}$	2	0	–3	0	1

Поскольку в последней таблице в столбце, пригодном для выбора разрешающего элемента, нет положительных чисел, то целевая функция неограниченна на области допустимых значений, то есть задача решения не имеет.

Зацикливание симплекс алгоритма

При работе симплекс методом мы переходим от одного опорного решения системы ограничений задачи к другому, причем так, чтобы значение целевой функции на следующем решении не уменьшалось. Поскольку опорных решений конечное число, мы должны прийти в оптимальное опорное решение, существование которого гарантируется фундаментальной теоремой. Однако мы можем не достигнуть оптимального решения, если в процессе перебора симплекс методом вернемся в опорное решение, которое уже встречалось ранее, и затем будем повторять цепочку опорных решений, пройденных ранее. Вычисления по симплекс методу войдут в бесконечно повторяющийся цикл и никогда не закончатся. Такое явление называется *зацикливанием симплекс алгоритма*.

Зацикливание — явление очень редкое. В литературе по линейному программированию имеются лишь несколько примеров задач, в которых возможно возникновение зацикливания.

Несмотря на возможность зацикливания, любую задачу линейного программирования надлежащей формы можно решить симплекс методом до конца. При рассмотрении симплекс алгоритма мы видели, что на очередном шаге разрешающий элемент может выбираться неоднозначно. Для однозначной организации вычислений приходится вводить добавочные правила. Можно показать, что зацикливание наступает лишь в случае возможности неоднозначного выбора разрешающего элемента. Если зацикливание наступило, следует изменить порядок вычислений, выбирая разрешающий элемент по-другому. Произойдет выход из цикла. Для борьбы с зацикливанием используют особые подпрограммы, гарантирующие выход из цикла в случае наступления зацикливания.

1.5. Метод искусственного базиса

Симплекс-метод применяется к задачам частного вида, у которых система ограничений имеет допустимый базисный вид. Непосредственное отыскание первого допустимого базисного вида системы ограничений обычно очень затруднительно. Существует два метода преодоления этой трудности. Первый из них называется *методом искусственного базиса*, а второй — *методом больших штрафов*.

Рассмотрим задачу в канонической форме

$$z=c_0+c_1x_1+c_2x_2+\ldots+c_nx_n\to\max,$$
$$\begin{cases} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n=b_1, \\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n=b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n=b_m; \end{cases}$$
$$x_i\ge 0,\quad i=1,2,\ldots,n.$$

Непосредственно к такой задаче симплекс-метод не применим. Следует привести систему ограничений к допустимому базисному виду. Это можно сделать с помощью симплекс-метода. Не ограничивая общности, мы можем считать, что свободные члены b_1, $b_2,\ldots$, b_m неотрицательны, так как в противном случае умножением некоторых уравнений на (–1) этого всегда можно добиться. Введем в нашу систему уравнений дополнительные переменные y_1, $y_2,\ldots$, y_m следующим образом

$$\begin{cases} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n+y_1=b_1, \\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n+y_2=b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n+y_m=b_m. \end{cases}$$

Эта новая система ограничений имеет допустимый базисный вид с базисными переменными y_1, $y_2,\ldots$, y_m. Новая и исходная системы эквивалентны при условии, что $y_1, y_2, \ldots, y_m=0$. Если путем эквивалентных преобразований переменные y_1, $y_2,\ldots$, y_m вывести из числа базисных, заменяя их другими переменными, и в полученном базисном виде положить $y_1, y_2, \ldots, y_m=0$, то мы получим базисный вид исходной системы. Преобразование новой системы к допустимому базисному виду, в котором $y_1, y_2, \ldots, y_m$ являются свободными переменными можно провести в процессе решения следующей вспомогательной задачи линейного программирования

$$f=-y_1-y_2-\ldots-y_m\to\max,$$
$$\begin{cases} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n+y_1=b_1, \\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n+y_2=b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n+y_m=b_m; \end{cases}$$
$$x_i, y_j\ge 0,\quad i=1,2,\ldots,n;\ j=1,2,\ldots,m.$$

При решении этой задачи могут представиться следующие два случая.

1) $\max f < 0$

В этом случае система ограничений задачи не имеет допустимого базисного вида, в котором искусственные переменные y_1, y_2,…, y_m являются свободными. Это означает, что исходная система не имеет допустимого базисного вида, т.е. любая задача линейного программирования в канонической форме с этой системой ограничений недопустима.

2) $\max f = 0$.

В этом случае в точке максимума искусственные переменные $y_1, y_2, \dots, y_m = 0$. При решении вспомогательной задачи симплекс-методом могут встретиться два случая:

а) В результате цепочки шагов все искусственные переменные становятся свободными. Полагая их равными нулю, мы получим допустимый базисный вид исходной системы ограничений, который можно использовать при решении исходной задачи симплекс-методом.

б) Не все искусственные переменные выводятся из состава базисных. Некоторыми простыми преобразованиями симплекс-таблицы всегда можно добиться вывода искусственных переменных из базиса, завершив тем самым подготовку к решению исходной задачи. Для этого следует произвести ряд прямых замещений базисных искусственных переменных еще оставшимися свободными переменными x_i. Поскольку в последнем опорном решении $y_1, y_2, \dots, y_m = 0$, столбец свободных членов симплекс-таблицы при этом не изменяется.

Замечание. В исходной системе ограничений некоторые переменные могут входить по одному в соответствующие уравнения, т.е. быть «уединенными». При составлении вспомогательной задачи можно не вводить искусственные переменные в уравнения, содержащие «уединенные» переменные, знаки коэффициентов при которых не противоположны знакам свободных членов. После деления этих уравнений на коэффициенты при «уединенных» переменных, эти переменные будут играть роль базисных во вспомогательной задаче.

Пример 1.8. Решить следующую задачу:

$$z = -x_1 - 2x_2 \to \max,$$

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

$$x_i \geq 0, \quad i = 1,2,\dots,5.$$

В первое уравнение можно не вводить искусственную переменную. Роль базисной переменной может играть x_3. Сформулируем вспомогательную задачу

$$f = -y_1 - y_2 \to \max,$$

$$\begin{cases} 3x_1 - 5x_2 + x_3 + 2x_4 = 1, \\ -2x_1 + 2x_2 - x_4 + x_5 + y_1 = 4, \\ -x_1 + 3x_2 - 2x_4 + x_5 + y_2 = 5; \end{cases}$$

$$y_j,\ x_i \geq 0 \quad (j = 1, 2;\ i = 1, 2, \dots, 5).$$

Исключим базисные переменные y_1, y_2, из целевой функции и составим первую симплекс-таблицу

Таблица 1

Баз.пер.	Св. чл.	x_1	x_2	x_3	x_4	$\downarrow x_5$	y_1	y_2
x_3	1	3	–5	1	2	0	0	0
$\leftarrow y_1$	4	–2	2	0	–1	1	1	0
y_2	5	–1	3	0	–2	1	0	1
f	–9	3	–5	0	3	–2	0	0

Таблица 2

Баз.пер.	Св. чл.	x_1	$\downarrow x_2$	x_3	x_4	x_5	y_1	y_2
x_3	1	3	–5	1	2	0	0	0
x_5	4	–2	2	0	–1	1	1	0
$\leftarrow y_2$	1	1	1	0	–1	0	–1	1
f	–1	–1	–1	0	1	0	2	0

Таблица 3

Баз.пер.	Св. чл.	x_1	x_2	x_3	x_4	x_5	y_1	y_2
x_3	6	8	0	1	–3	0	0	5
x_5	2	–4	0	0	1	1	1	–2
x_2	1	1	1	0	–1	0	–1	1
f	0	0	0	0	0	0	1	1

Подготовка системы ограничений завершена. Отбрасывая в последней таблице два последних столбца и последнюю строку, получим после исключения базисных переменных из целевой функции исходной задачи ее первую симплекс-таблицу

Таблица 1

Баз.пер.	Св. чл.	$\downarrow x_1$	x_2	x_3	x_4	x_5
$\leftarrow x_3$	6	8	0	1	–3	0
x_5	2	–4	0	0	1	1
x_2	1	1	1	0	–1	0
z	–2	–1	0	0	2	0

Таблица 2

Баз.пер.	Св. чл.	x_1	x_2	x_3	x_4	x_5
x_1	3/4	1	0	1/8	–3/8	0
x_5	5	0	0	1/2	–1/2	1
x_2	1/4	0	1	–1/8	–5/8	0
z	–5/4	0	0	1/8	13/8	0

По последней симплекс-таблице читаем ответ исходной задачи: $z_{\max} = -5/4$, точка максимума $X(3/4;\ 1/4;\ 0;\ 0;\ 5)$.

Иногда при решении вспомогательной задачи в последней симплекс-таблице не все искусственные переменные выводятся из состава базисных.

Пример 1.9. Привести к допустимому базисному виду следующую систему ограничений

$$\begin{cases} -x_1 - 2x_2 + 3x_3 - 3x_4 - x_5 - x_6 = -1, \\ x_1 + 4x_2 - 5x_3 - 5x_4 - 3x_5 - x_6 = 2, \\ -4x_1 + 4x_2 - 12x_3 - 2x_5 + 2x_6 = 2; \end{cases}$$

$$x_i \ge 0,\ i = 1, 2, \ldots, 6.$$

Сформулируем вспомогательную задачу

$$f = -y_1 - y_2 - y_3 \to \max,$$

$$\begin{cases} x_1 + 2x_2 - 3x_3 + 3x_4 + x_5 + x_6 + y_1 = 1, \\ x_1 + 4x_2 - 5x_3 - 5x_4 - 3x_5 - x_6 + y_2 = 2, \\ -4x_1 + 4x_2 - 12x_3 - 2x_5 + 2x_6 \quad y_3 = 2; \end{cases}$$

$$x_i, y_j \ge 0,\ i = 1, 2, \ldots, 6;\ j = 1, 2, 3.$$

Составим первую симплекс-таблицу, производя попутно исключение базисных переменных из целевой функции.

Таблица 1

Баз. пер.	Св. чл.	x_1	$\downarrow x_2$	x_3	x_4	x_5	x_6	y_1	y_2	y_3
$\leftarrow y_1$	1	1	2	–3	3	1	1	1	0	0
y_2	2	1	4	–5	–5	–3	–1	0	1	0
y_3	2	–4	4	–12	0	–2	2	0	0	1
f	–5	2	–10	20	2	4	–2	0	0	0

Таблица 2

Баз. пер.	Св. чл.	x_1	x_2	$\downarrow x_3$	x_4	x_5	x_6	y_1	y_2	y_3
x_2	1/2	1/2	1	–3/2	3/2	1/2	1/2	1/2	0	0
$\leftarrow y_2$	0	–1	0	1	–11	–5	–3	–2	1	0
y_3	0	–6	0	–6	–6	–4	0	–2	0	1
f	0	7	0	5	17	9	3	5	0	0

Таблица 2 является последней в решении вспомогательной задачи. Однако искусственные переменные еще не выведены из состава базисных. Мы выведем переменную y_2 из состава базисных, вводя вместо нее переменную x_3, несмотря на то, что правило выбора разрешающего элемента здесь нарушено. Строку для f можно при этом вовсе отбросить.

Таблица 3

Баз. пер.	Св. чл.	$\downarrow x_1$	x_2	x_3	x_4	x_5	x_6	y_1	y_2	y_3
x_2	1/2	–1	1	0	–15	–7	–4	–5/2	3/2	0
x_3	0	–1	0	1	–11	–5	–3	–2	1	0
$\leftarrow y_3$	0	–12	0	0	–72	–34	–18	–14	6	1

Произведем замещение базисной переменной y_3, например, на переменную x_1.

Таблица 4

Баз. пер.	Св. чл.	x_1	x_2	x_3	x_4	x_5	x_6	y_1	y_2	y_3
x_2	1/2	0	1	0	–11	–23/6	–5/2	–8/6	1	$-\frac{1}{12}$
x_3	0	0	0	1	–5	–13/6	–3/2	–5/6	1/2	$-\frac{1}{12}$
x_1	0	1	0	0	6	17/6	3/2	7/6	–1/2	$-\frac{1}{12}$

Теперь можно записать допустимый базисный вид исходной системы ограничений, меняя для удобства записи уравнения местами.

$$\begin{cases} x_1 + 6x_4 + \dfrac{17}{6}x_5 + \dfrac{3}{2}x_6 = 0, \\ x_2 - 11x_4 - \dfrac{23}{6}x_5 - \dfrac{5}{2}x_6 = \dfrac{1}{2}, \\ x_3 - 5x_4 - \dfrac{13}{6}x_5 - \dfrac{3}{2}x_6 = 0. \end{cases}$$

1.6. Метод больших штрафов

Метод искусственного базиса решает задачу линейного программирования в два этапа. Вначале при подготовке исходной задачи решается вспомогательная задача, а затем симплекс-метод применяется к решению исходной задачи. В методе больших штрафов ограничиваются одним этапом. По данной задаче строят некоторую вспомогательную задачу (*M*-задачу), имеющую нужную форму, и ее решают симплекс-методом. Затем по решению этой задачи находят и решение исходной задачи. Пусть дана произвольная задача в канонической форме:

$$z = c_0 + c_1x_1 + c_2x_2 + \ldots + c_nx_n \to \max;$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m; \end{cases}$$

$$x_i \ge 0,\ i = 1, 2, \ldots, n.$$

Как и раньше считаем, что свободные члены b_1, b_2, … , $b_m \ge 0$. Введем искусственные переменные y_1, y_2, … , $y_m \ge 0$ в систему ограничений и целевую функцию исходной задачи следующим образом:

$$z_M = c_0 + c_1x_1 + c_2x_2 + \ldots + c_nx_n - M\left(y_1 + y_2 + \ldots + y_m\right) \longrightarrow \max;$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + y_1 = b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + y_2 = b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + y_m = b_m; \end{cases}$$

$$x_i \ge 0,\ i = 1, 2, \ldots, n;\ y_j \ge 0,\ j = 1, 2, \ldots, m.$$

Здесь $M > 0$ — достаточно большое число. Построенная задача называется *M*-задачей по отношению к исходной задаче. Система ограничений этой задачи имеет допустимый базисный вид. Правда, целевая функция содержит базисные переменные $y_1, y_2, \dots, y_m$. Но они легко исключаются из выражения для z_M с помощью самих уравнений системы ограничений. Имеют место следующие утверждения:

1. Если для всех достаточно больших $M>0$ M-задача имеет решение, то и исходная задача имеет решение, причем $z_{\max} = z_{M\max}$, а точка максимума для z может быть получена из точки максимума для z_M отбрасыванием значений искусственных переменных. При этом в точке максимума z_M значения искусственных переменных равны нулю.

2. Если при всех достаточно больших $M>0$ M-задача не имеет решения, то и исходная задача не имеет решения.

Таким образом, для решения исходной задачи следует выбрать некоторое достаточно большое M и решить M-задачу. Если число M выбрано недостаточно большим, то точка максимума для z_M может иметь отличные от нуля значения искусственных переменных. В этом случае определить по решению M-задачи решение исходной задачи нельзя. Обычно число M берётся на порядок больше, чем коэффициенты в системе ограничений и целевой функции исходной задачи. Отметим также, что при введении искусственных переменных нет нужды вводить их во все ограничения задачи. Искусственные переменные можно не вводить в ограничения, содержащие уединённые переменные, знак коэффициентов которых совпадает со знаком соответствующих свободных членов.

Пример 1.10. Решить задачу:

$$z = -3x_1 - 2x_2 \to \max;$$
$$\begin{cases} x_1 + x_2 = 10, \\ x_1 - x_3 = 4; \end{cases}$$
$$x_1, x_2, x_3 \ge 0.$$

Система ограничений этой задачи имеет базисный вид, однако он не соответствует опорному решению. В первом уравнении роль базисной может играть переменная x_2. Во второе уравнение придется ввести искусственную переменную y. M-задачу построим при M=20:

$$z_M = -3x_1 - 2x_2 - 20y \to \max;$$
$$\begin{cases} x_1 + x_2 = 10, \\ x_1 - x_3 + y = 4; \end{cases}$$
$$x_1, x_2, x_3, y \ge 0.$$

Выразим базисные переменные x_2, y из системы ограничений и подставим в целевую функцию z_M. Задача примет вид

$$z_M = -100 + 19x_1 - 20x_3 \longrightarrow \max;$$
$$\begin{cases} x_1 + x_2 = 10, \\ x_1 - x_3 + y = 4; \end{cases}$$
$$x_1, x_2, x_3, y \geq 0.$$

Для этой задачи можно составить симплекс-таблицы:

Б.п.	Св.чл.	$\boldsymbol{x}_1$	$\boldsymbol{x}_2$	$\boldsymbol{x}_3$	$\boldsymbol{y}$
$\boldsymbol{x}_2$	10	1	1	0	0
$\boldsymbol{y}$	4	**1**	0	–1	1
$\mathbf{z}_{\boldsymbol{M}}$	–100	–19	0	20	0

Б.п.	Св.чл.	x_1	x_2	x_3	y
$\boldsymbol{x}_2$	6	0	1	1	–1
$\boldsymbol{x}_1$	4	1	0	–1	1
z_M	–24	0	0	0	19

M-задача имеет следующее решение:

$$z_{M\max} = -24; \;\; x_1 = 4; \;\; x_2 = 6; \; x_3 = 0; \; y = 0.$$

Решение исходной задачи:

$$z_{\max} = -24\,;\; x_1 = 4\,;\; x_2 = 6\,;\; x_3 = 0\,.$$

1.7. Понятие о «трудоемкости» симплекс алгоритма и о методах внутренних точек

Полувековая практика решения задач линейного программирования показала высокую эффективность симплекс-метода и различных его модификаций. При решении задач с m ограничениями и n переменными, как правило, оказывалось достаточно m итераций. При этом количество элементарных арифметических операций имело порядок n^2m. Однако теоретические исследования по оценке «трудоемкости» решения задач линейного программирования симплекс-методом показала, что эта «трудоемкость» может быть значительно больше. В 1972 г. американские ученые В. Кли и Дж. Минти построили пример задачи линейного программирования с n переменными и $2n$ ограничениями, для решения которой требуется не менее $2^n - 1$ итераций симплекс-метода. Тем самым было показано, что симплекс-метод на классе всех линейных задач является алгоритмом «экспоненциальной трудоемкости» — количество необходимых вычислений оценивается экспоненциальной функцией параметров задачи. Этот факт означает, что существуют задачи не слишком большой размерности, решение которых симплекс-методом невозможно за обозримое время. Хотя все примеры такого рода искусственны и для задач, пришедших из практики, ничего подобного не происходит, возник естественный вопрос о существовании алгоритма, для которого необходимый объем вычислений при решении произвольной задачи линейного программирования оценивается полиномом от параметров задачи. Другими словами, обладает ли класс задач линейного программирования экспоненциальной сложностью или эта сложность полиномиальная? Вопрос этот нашел свое решение в теореме, доказанной советским математиком Л.Г. Хачияном [30]. Он применил к задачам линейного программирования новый алгоритм (метод эллипсоидов), построенный усилиями советских ученых А.С. Немировского, Н.З. Шоша и Д.Б. Юдина. Для формулировки теоремы Хачияна обозначим через h максимум модулей коэффициентов и свободных членов в целевой функции и системе ограничений задачи в стандартной форме. Для произвольных функций $f(t)$, $g(t)$ будем писать $f = O(g)$, если существует такая константа C, что $f(t) \leq C\, g(t)$ для всех t.

Теорема 1.2. *Для решения задачи линейного программирования достаточно* $O(n^4(n+m)\ln hn)$ *элементарных операций* (*сложение, вычитание, умножение, деление, извлечение квадратного корня, нахождение наибольшего из двух чисел*). *При этом достаточно использовать для вычислений число разрядов, равное* $O(n \ln hn)$.

Таким образом, *класс задач линейного программирования имеет полиномиальную сложность*. К сожалению, в вычислительном плане метод эллипсоидов оказался неперспективным. Однако факт полиномиальной сложности задач линейного программирования привёл в дальнейшем к созданию целого класса эффективных алгоритмов линейного программирования, которые получили название методов внутренней точки [7]. Первым из этих методов был алгоритм Н. Кармаркара, предложенный в 1984 г. Метод внутренней точки тоже переходит от точки к точке, улучшая значение целевой функции, но остаётся при этом во внутренности области допустимых значений задачи. Подобравшись достаточно близко к оптимальному опорному решению, он находит это решение. Алгоритм Кармаркара имеет время работы $O(n^4L)$, где L — длина битовой записи входных данных.

В заключение отметим, что приведенные оценки указывают на более высокую эффективность методов внутренней точки по сравнению с симплекс-методом лишь для самых «плохих» задач линейного программирования, далеких от реальных. На практике же симплекс-метод по-прежнему является основным инструментом в линейном программировании.

Контрольные вопросы и задачи для самостоятельного решения

1. Как формулируется общая задача линейного программирования?

2. Когда задача линейного программирования называется имеющей каноническую форму?

3. Какая форма задачи линейного программирования называется стандартной?

4. В чем заключается геометрическое истолкование системы ограничений и целевой функции задачи в случае двух переменных?

5. Дайте определения базисного вида системы линейных уравнений, базисного и опорного решений такой системы.

6. Сформулируйте фундаментальную теорему симплекс-метода.

7. К какому виду должна быть приведена задача линейного программирования перед применением симплекс-метода?

8. Как составить первую симплекс-таблицу?

9. Опишите порядок работы с симплекс-таблицей. В чем заключается признак того, что симплекс-таблица является последней? Как прочесть решение задачи по последней симплекс-таблице? В каком случае по последней симплекс-таблице можно заключить, что задача не имеет решения по причине неограниченности целевой функции на области допустимых значений?

10. Для чего применяется метод искусственного базиса? Какие основные случаи могут представиться при работе этим методом?

11. Опишите метод больших штрафов. Как составить M-задачу для задачи линейного программирования в канонической форме?

12. Как избежать зацикливания симплекс алгоритма?

13. Что понимается под трудоемкостью симплекс метода? Что означает его экспоненциальная трудоемкость на классе всех задач линейного программирования?

14. Существуют ли алгоритмы решения задач линейного программирования полиномиальной трудоемкости? Обладает ли класс всех задач линейного программирования полиномиальной сложностью?

Построить математические модели в задачах 1.1 — 1.4.

1.1. Для изготовления трех видов изделий A, B, C используется токарное, фрезерное, сварочное и шлифовальное оборудование. Затраты времени на обработку одного изделия для каждого из типов оборудования указаны в следующей ниже таблице. В ней же указан общий фонд рабочего времени каждого из типов оборудования, а также прибыль от реализации одного изделия каждого вида.

Требуется определить, сколько изделий каждого вида следует изготовить предприятию, чтобы прибыль от их реализации была максимальной.

Тип оборудования	Затраты времени (станко-ч) на обработку одного изделия вида			Общий фонд рабочего времени оборудования (ч)
	A	*B*	*C*	
Фрезерное	2	4	5	120
Токарное	1	8	6	280
Сварочное	7	4	5	240
Шлифовальное	4	6	7	360
Прибыль (руб.)	10	14	12	

1.2. Кондитерская фабрика для производства трех видов карамели *A*, *B* и *C* использует три вида основного сырья: сахарный песок, патоку и фруктовое пюре. Нормы расхода сырья каждого вида на производство 1 т. карамели данного вида приведены в таблице. В ней же указано общее количество сырья каждого вида, которое может быть использовано фабрикой, а также приведена прибыль от реализации 1 т. карамели каждого вида.

Вид сырья	Нормы расхода сырья (т) на 1 т карамели			Общее количество сырья (т)
	A	*B*	*C*	
Сахарный песок	0,8	0,5	0,6	800
Патока	0,4	0,4	0,3	600
Фруктовое пюре	–	0,1	0,1	120
Прибыль от реализации 1 т карамели (руб.)	108	112	126	

Найти оптимальный план производства карамели, обеспечивающий максимальную прибыль от ее реализации.

1.3. При откорме животных каждое животное ежедневно должно получать не менее 60 ед. питательного вещества *A*, не менее 50 ед. вещества *B* и не менее 12 ед. вещества *C*. Указанные питательные вещества содержаться в трех видах корма. Содержание единиц питательных веществ в 1 кг каждого из видов корма приведено в следующей таблице:

Питательные вещества	Количество единиц питательных веществ в 1 кг корма вида		
	I	II	III
A	1	3	4
B	2	4	2
C	1	4	3

Составить дневной рацион, обеспечивающий получение необходимого количества питательных веществ при минимальных денежных затратах, если цена 1 кг корма I-го вида составляет 9 коп., корма II-го вида — 12 коп., а корма III-го вида — 10 коп.

1.4. При производстве чугунного литья используется n различных исходных шихтовых материалов (чугун различных марок, стальной лом, феррофосфор и др.) Химический состав чугунного литья определяется содержанием в нем химических элементов (кремния, марганца, фосфора и др.). Готовый чугун должен иметь строго определенный химический состав, который определяется величинами H_j, представляющими собой доли (в процентах) j-го химического элемента в готовом продукте (j=1,2,…, m). При этом считаются известными величины h_{ij} — содержание (в процентах) j-го химического элемента в i-ом исходном шихтовом материале, а также величины c_i — цены единицы каждого шихтового материала ($i = 1,2,\ldots,n$). Определить состав шихты, обеспечивающий получение литья заданного качества при минимальной общей стоимости используемых шихтовых материалов.

В задачах 1.5 — 1.8 *привести математическую модель линейного программирования к каноническому виду.*

1.5.

$$z = -2x_1 - x_2 + x_3 \to \min,$$

$$\begin{cases} 2x_1 - x_2 + 6x_3 \le 12, \\ 3x_1 + 5x_2 - 12x_3 = 14, \\ -3x_1 + 6x_2 - 4x_3 \le 18, \end{cases}$$

$$x_1, x_2, x_3 \ge 0.$$

1.6.

$$z = -2x_1 + x_2 + 5x_3 \to \max,$$

$$\begin{cases} 4x_1 + 2x_2 + 5x_3 \le 12, \\ 6x_1 - 3x_2 + 4x_3 = 15, \\ 3x_1 + 3x_2 - 2x_3 \le 16, \end{cases}$$

$$x_1, x_2, x_3 \ge 0.$$

1.7.

$$z = 2x_1 - 5x_2 + 3x_3 \to \min,$$

$$\begin{cases} -x_1 + x_2 + x_3 \ge 12, \\ x_1 + 5x_2 - 6x_3 \le 16, \\ 3x_1 + x_2 + x_3 \ge 18, \end{cases}$$

$$x_1, x_2 \ge 0.$$

1.8.

$$z = -3x_1 + x_2 - 5x_3 \to \min,$$

$$\begin{cases} 2x_1 + 5x_2 - 7x_3 \le 4, \\ -4x_1 - 3x_2 + 8x_3 \ge 15, \\ 3x_1 - 2x_2 + 10x_3 \le 11, \end{cases}$$

$$x_2, x_3 \ge 0.$$

Используя геометрическое истолкование задач линейного программирования, найти решения задач 1.9 — 1.13.

1.9.

$$z = x_1 + x_2 \to \max;$$

$$\begin{cases} x_1 + 2x_2 \le 14, \\ -5x_1 + 3x_2 \le 15, \\ 4x_1 + 6x_2 \ge 24, \end{cases}$$

$$x_1, \quad x_2 \ge 0.$$

Ответ: точка максимума (14; 0); z_{max}=14.

1.10.

$$z = x_1 + 2x_2 \to \max;$$

$$\begin{cases} 4x_1 - 2x_2 \le 12, \\ -x_1 + 3x_2 \le 6, \\ 2x_1 + 4x_2 \ge 16, \end{cases}$$

$$x_1, \quad x_2 \ge 0.$$

Ответ: точка максимума (4,8;3,6); z_{max}=12.

1.11.

$$z = -2x_1 + x_2 \to \min;$$

$$\begin{cases} 3x_1 - 2x_2 \le 12, \\ -x_1 + 2x_2 \le 8, \\ 2x_1 + 3x_2 \ge 6, \end{cases}$$

$$x_1, \quad x_2 \ge 0.$$

Ответ: точка минимума (10; 9); z_{min}= –11

1.12.

$$z = -x_1 + 4x_2 + 2x_4 - x_5 \to \max;$$

$$\begin{cases} x_1 - 5x_2 + x_3 = 5, \\ -x_1 + x_2 + x_4 = 4, \\ x_1 + x_2 + x_5 = 8, \end{cases}$$

$$x_1, \quad x_2, \ldots, x_5 \ge 0.$$

Ответ: точка максимума (2;6;33;0;0;); z_{max}=22.

1.13.

$$z = -5x_1 + x_2 - x_3 \to \max;$$

$$\begin{cases} 3x_1 - x_2 - x_3 = 4, \\ x_1 - x_2 + x_3 - x_4 = 1, \\ 2x_1 + x_2 + 2x_3 + x_5 = 7, \end{cases}$$

$$x_1, \quad x_2, \ldots, x_5 \ge 0.$$

Ответ: точка максимума (4/3; 0; 0; 1/3; 13/3;); z_{max}=–20/3.

В задачах 1.14 — 1.17 *привести систему уравнений к какому-нибудь базисному виду.*

1.14.

$$\begin{cases} x_1 - 2x_2 + 3x_3 + 4x_4 = 1, \\ 4x_1 - 7x_2 + 2x_3 + x_4 = 3, \\ 3x_1 - 5x_2 - x_3 - 3x_4 = 2. \end{cases}$$

1.15.

$$\begin{cases} x_1 + 4x_2 - 2x_3 + 3x_5 = 2, \\ 2x_1 + 9x_2 - x_3 - 4x_4 = 5, \\ x_1 + 5x_2 + x_3 - 4x_4 + 3x_5 = 3. \end{cases}$$

1.16.

$$\begin{cases} x_1 + 3x_2 - x_3 - 2x_4 = 1, \\ 2x_1 + 7x_2 - 4x_3 - 3x_4 = 3, \\ x_1 + 4x_2 - 3x_3 - x_4 = 2. \end{cases}$$

1.17.

$$\begin{cases} x_1 - 5x_2 + 3x_3 + 4x_4 = 4, \\ 2x_1 - 9x_2 + 2x_3 + x_5 = 7, \\ x_1 - 4x_2 - x_3 - 4x_4 + x_5 = 3. \end{cases}$$

С помощью симплекс-метода и его модификаций найти решение задач 1.18 — 1.27.

1.18.

$$z = 3x_1 + 2x_3 - 6x_6 \to \max;$$

$$\begin{cases} 2x_1 + x_2 - 3x_3 + 6x_6 = 18, \\ -3x_1 + 2x_3 + x_4 - 2x_6 = 24, \\ x_1 + 3x_3 + x_5 - 4x_6 = 36, \end{cases}$$

$x_i \geq 0 \ (i = \overline{1,6}).$

Ответ: точка максимума (18; 0; 6; 66; 0; 0); z_{max}=66.

1.19.

$$z = 2x_1 + 3x_2 - x_4 \to \max;$$

$$\begin{cases} 2x_1 - x_2 - 2x_4 + x_5 = 16, \\ 3x_1 + 2x_2 + x_3 - 3x_4 = 18, \\ -x_1 + 3x_2 + 4x_4 + x_6 = 24, \end{cases}$$

$x_i \geq 0 \ (i = \overline{1,6}).$

Ответ: точка максимума (6/11; 90/11; 0; 0; 254/11; 0); z_{max}=282/11.

1.20.

$$z = 8x_2 + 7x_4 + x_6 \to \max;$$

$$\begin{cases} x_1 - 2x_2 - 3x_4 - 2x_6 = 12, \\ 4x_2 + x_3 - 4x_4 - 3x_6 = 12, \\ 5x_2 + 5x_4 + x_5 + x_6 = 25, \end{cases}$$

$x_i \geq 0 \ (i = \overline{1,6}).$

Ответ: точка максимума (23; 4; 0; 1; 0; 0); z_{max}=39.

1.21.

$z = x_1 + 3x_2 - 5x_4 \rightarrow \max;$

$$\begin{cases} 2x_1 + 4x_2 + x_3 + 2x_4 = 28, \\ -3x_1 + 5x_2 - 3x_4 + x_5 = 30, \\ 4x_1 - 2x_2 + 8x_4 + x_6 = 32, \end{cases}$$

$x_i \geq 0 \ (i = \overline{1,6}).$

Ответ: точка максимума (10/11; 72/11; 0; 0; 0; 456/11); z_{max}=226/11.

1.22.

$z = 3x_1 + 2x_5 - 5x_6 \rightarrow \max;$

$$\begin{cases} 2x_1 + x_2 - 3x_5 + 5x_6 = 34, \\ 4x_1 + x_3 + 2x_5 - 4x_6 = 28, \\ -3x_1 + x_4 - 3x_5 + 6x_6 = 24, \end{cases}$$

$x_i \geq 0 \ (i = \overline{1,6}).$

Ответ: точка максимума (0; 76; 0; 66; 14; 0); z_{max}=28.

1.23.

$z = x_1 + 2x_2 - x_3 \rightarrow \max;$

$$\begin{cases} -x_1 + 4x_2 - 2x_3 \leq 6, \\ x_1 + x_2 + 2x_3 \geq 6, \\ 2x_1 - x_2 + 2x_3 = 4, \end{cases}$$

$x_1,\ x_2\ x_3 \geq 0.$

Ответ: точка максимума (2,8; 2,4; 0,4); z_{max}=7,2.

1.24.

$z = 8x_1 - 3x_2 + x_3 + 6x_4 - 5x_5 \rightarrow \max;$

$$\begin{cases} 2x_1 + 4x_2 + x_3 + x_4 - 2x_5 = 28, \\ x_1 - 2x_2 + x_4 + x_5 = 31, \\ -x_1 + 3x_2 + 5x_3 + 4x_4 - 8x_5 = 118, \end{cases}$$

$x_i \geq 0 \quad (i = \overline{1,5}).$

Ответ: точка максимума (0; 0; 6; 28; 3); z_{max}=159.

1.25.

$$z = 2x_1 - 3x_2 + 4x_3 + 5x_4 - x_5 + 8x_6 \to \max;$$

$$\begin{cases} x_1 + 5x_2 - 3x_3 - 4x_4 + 2x_5 + x_6 = 120, \\ 2x_1 + 9x_2 - 5x_3 - 7x_4 + 4x_5 + 2x_6 = 320, \end{cases}$$

$x_i \ge 0$ $(i = \overline{1,6})$.

Ответ: точка максимума (0; 0; 0; 80; 0; 440); z_{max}=3920.

1.26.

$$z = -3x_1 + 5x_2 - 3x_3 + x_4 + x_5 + 8x_6 \to \max;$$

$$\begin{cases} x_1 - 3x_2 + 4x_3 + 5x_4 - 6x_5 + x_6 = 60, \\ 7x_1 - 17x_2 + 26x_3 + 31x_4 - 35x_5 + 6x_6 = 420, \end{cases}$$

$x_i \ge 0$ $(i = \overline{1,6})$.

Ответ: точка максимума (0; 0; 0; 0; 60; 420); z_{max}=3420.

1.27.

$$z = 5x_1 - x_2 + 8x_3 + 10x_4 - 5x_5 + x_6 \to \max;$$

$$\begin{cases} 2x_1 - x_2 + 3x_4 + x_5 - x_6 = 36, \\ -x_1 + 2x_2 + x_3 + 2x_4 + 2x_6 = 20, \\ 3x_1 - x_2 + 2x_3 - x_4 + 3x_5 + x_6 = 30, \end{cases}$$

$x_i \ge 0$ $(i = \overline{1,6})$.

Ответ: точка максимума (6; 0; 10; 8; 0; 0); z_{max} = 190.

2. ТРАНСПОРТНАЯ ЗАДАЧА

2.1. Закрытая транспортная задача. Нахождение опорных планов

Одной из задач линейного программирования является транспортная задача, состоящая, в общей постановке, в отыскании оптимального плана перевозок некоторого однородного груза с m баз A_1, A_2, ..., A_m n потребителям B_1, B_2, ..., B_n. Пусть имеются определенные запасы груза на базах A_1, A_2, ..., A_m, которые мы обозначим a_1, a_2, ... , a_m соответственно. Заказы каждого из потребителей (потребности) обозначим b_1, b_2, ... , b_n. Общее количество имеющегося груза обозначим A ($A = a_1 + a_2 + ...+ a_m$), а общие потребности — через B ($B = b_1 + b_2 + ... + b_n$). При условии $A = B$ мы имеем закрытую модель, а при $A \neq B$ — открытую модель транспортной задачи.

В этом параграфе мы рассмотрим закрытую транспортную задачу. Предположим, что базы и потребители соединены коммуникациями, например железными дорогами. Обозначим стоимость перевозки единицы груза из пункта A_i в пункт B_j через c_{ij}. Числа c_{ij} можно расположить в виде матрицы

$$\begin{bmatrix} c_{11}c_{12}\dots c_{1n} \\ c_{21}c_{22}\dots c_{2n} \\ \dots\dots\dots\dots \\ c_{m1}c_{m2}\dots c_{mn} \end{bmatrix}$$

Эта матрица называется *матрицей стоимостей*. Задача заключается в нахождении такого плана перевозок, чтобы общая стоимость их была минимальной. При этом необходимо выполнить следующие два условия:

1) все запасы груза должны быть вывезены;
2) заказы потребителей должны быть выполнены.

Такие требования можно соблюсти лишь при условии $A = B$, то есть в случае закрытой модели. Обозначим через x_{ij} количество груза, которое планируется перевезти из пункта A_i в пункт B_j. Эти величины будут представлять собой переменные в нашей задаче. Их также можно расположить в виде матрицы, которая называется матрицей перевозок. Общая стоимость перевозок (целевая функция нашей задачи) имеет вид

$$z = \sum_{i,\,j} c_{ij} x_{ij}$$

Эту функцию требуется минимизировать. При этом, однако, величины x_{ij} не могут принимать произвольные значения. Они *неотрицательны* и удовлетворяют следующим ограничениям, которые выражают требования 1) и 2), сформулированные выше:

$$\begin{cases} x_{11}+x_{12}+\ldots+x_{1n}=a_1; \\ x_{21}+x_{22}+\ldots+x_{2n}=a_2; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{m1}+x_{m2}+\ldots+x_{mn}=a_m. \end{cases} \qquad \begin{cases} x_{11}+x_{21}+\ldots+x_{m1}=b_1; \\ x_{12}+x_{22}+\ldots+x_{m2}=b_2; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_{1n}+x_{2n}+\ldots+x_{mn}=b_n. \end{cases}$$

Приведенная система ограничений состоит из двух подсистем, которые называются системой горизонтальных и вертикальных уравнений. Таким образом, закрытая транспортная задача является задачей линейного программирования с ограничениями равенствами. Система ее ограничений содержит $n+m$ уравнений с $n{\cdot}m$ неизвестными. Для решения транспортной задачи также применяют симплекс-метод, но в силу специфики задачи здесь можно обойтись без симплекс-таблиц. План перевозок с указанием запасов и потребностей, а также стоимостей перевозок, удобно записывать в виде таблицы данных:

Запасы	Потребности			
	b_1	b_2	. . .	b_n
a_1	$\boxed{c_{11}}$ x_{11}	$\boxed{c_{12}}$ x_{12}	. . .	$\boxed{c_{1n}}$ x_{1n}
a_2	$\boxed{c_{12}}$ x_{21}	$\boxed{c_{22}}$ x_{22}	. . .	$\boxed{c_{2n}}$ x_{2n}
. . .	. . .	. . .	. . .	. . .
a_m	$\boxed{c_{m1}}$ x_{m1}	$\boxed{c_{m2}}$ x_{m2}	. . .	$\boxed{c_{mn}}$ x_{mn}

Решение задачи можно получить путем некоторых преобразований этой таблицы, которые соответствуют переходу от одного плана перевозок к другому. Как и в общем случае, оптимальное решение ищется среди опорных решений системы ограничений транспортной задачи. Ранг этой системы равен $n + m - 1$, поэтому среди $m{\cdot}n$ переменных x_{ij} выделяются $m + n - 1$ базисных переменных, а остальные $(m - 1){\cdot}(n - 1)$ переменные являются свободными. В базисном решении свободные переменные равны нулю. Обычно эти нули в таблицу не вписывают, оставляя соответствующие клетки пустыми. Таким образом, при внесении в таблицу перевозок вместо x_{ij} чисел, представляющих опорный план, мы имеем $m + n - 1$ заполненных и $(m - 1)(n - 1)$ пустых клеток. Как и в общем случае, решение транспортной задачи начинается с отыскания первого опорного плана (исходного опорного решения системы ограничений). Мы рассмотрим два метода построения такого опорного плана. Суть обоих методов состоит в том, что опорный план составляется последовательно в несколько шагов (точнее, $m + n - 1$ шагов). На каждом из шагов заполняется одна клетка. При рассмотрении клетки с номерами (k, r) на первом шаге может представиться три случая:

а) $a_k > b_r$;

б) $a_k = b_r$;

в) $a_k < b_r$

В случае а) в клетку ставится число b_r, вычеркивается r-тый столбец, а запасы в пункте A_k полагаются равными $a_k - b_r$. В случае в) в клетку записывается число a_k, вычеркивается k-тая строка, а потребности в пункте b_r полагаются равными $b_r - a_k$. В случае б) в клетку ставится число $a_k = b_r$ и вычеркивается, по выбору, строка или столбец (вычеркивать и строку, и столбец нельзя). Оставшиеся потребности в пункте b_r или запасы в пункте a_k полагаются равными нулю. После первого шага наша таблица сократилась на одну строку или на один столбец, а потребности или запасы соответственно подправлены. В сокращенной таблице снова выбираем для заполнения клетку и повторяем все сначала. Начиная с первоначально данной таблицы и повторив $m + n - 2$ раз описанный шаг, мы придем к “таблице”, состоящей из одной клетки. Заполнив эту последнюю клетку и совершив $m + n - 1$ шаг, мы получим искомый опорный план. Правда, мы не указали, каким образом на каждом шаге мы выбираем клетку для заполнения. Различие методов отыскания первого опорного плана как раз и состоит в различии способов выбора заполняемой клетки.

Метод северо-западного угла. В этом методе на каждом шаге построения первого опорного плана заполняется верхняя левая клетка ("северо-западный угол") оставшейся таблицы.

Метод наименьшей стоимости. В этом методе на каждом шаге построения первого опорного плана заполняется та клетка оставшейся таблицы, которая имеет наименьший тариф (c_{kr}). Если такая клетка не единственная, то заполняется любая из них.

Пример 2.1. Найдем первый опорный план следующей транспортной задачи методом северо-западного угла.

(10) (30)

	Запасы	Потребности		
		40	30	10
	10	5 10	7	5
	20	2 20	1	4
(10) (40)	50	6 10	3 30	2 10

В клетку x_{11} ставим 10 и вычеркиваем строку. Потребности в B_1 станут равными 30. В клетку x_{21} ставим 20 и вычеркиваем строку. Потребности в B_1 равны 10. В клетку x_{31} ставим 10 и вычеркиваем столбец. В клетку x_{32} ставим 30 и вычеркиваем столбец. Запасы в A_3 станут равными 10. В клетку x_{33} ставим 10. Процесс нахождения опорного плана окончен. Базисными переменными здесь будут x_{11}, x_{21}, x_{31}, x_{32}, x_{33}. Остальные переменные свободные.

Метод наименьшей стоимости дает здесь другой опорный план:

(10) (0)

	Запасы	Потребности		
		40	30	10
	10	5 10	7	5
	20	2	1 20	4
(40)	50	6 30	3 10	2 10

Часто значение целевой функции на опорном плане, найденном методом наименьшей стоимости, меньше, чем на плане, полученном методом северо-западного угла. Этот опорный план как бы ближе к оптимальному. Хотя это и не всегда так, но часто нахождение оптимального решения, исходя из опорного плана, построенного методом наименьшей стоимости, требует меньше вычислений.

2.2. Комбинаторные свойства циклов в матрице

Под матрицей в этом параграфе понимается таблица, состоящая из клеток. *Циклом* в матрице будем называть ломаную линию (рис. 2.1), звенья которой располагаются по строкам и столбцам матрицы, и которая удовлетворяет следующим двум условиям:

1) эта ломаная является связной, т.е. из любой её вершины можно попасть в любую другую по звеньям ломаной;

2) в каждой вершине сходятся ровно два звена, причем одно из них располагается по строке, а другое по столбцу таблицы.

Циклы в таблице могут иметь *самопересечения*, т.е. звенья ломаной могут пересекаться в точке, не являющейся вершиной цикла (рис. 2.1, б).

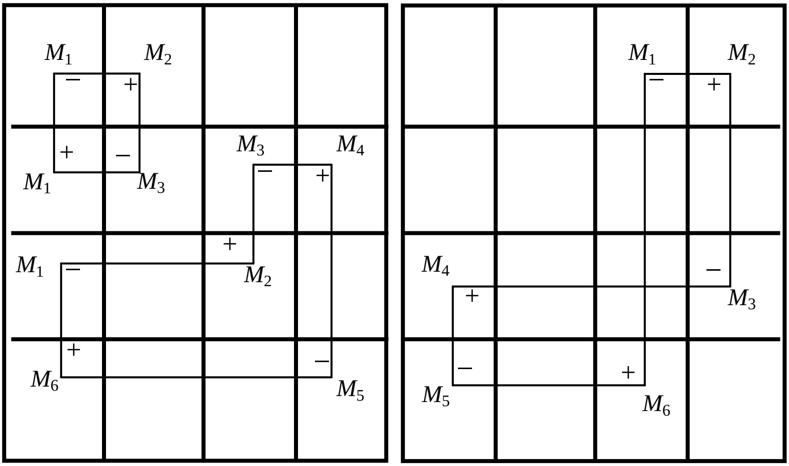


Рис. 2.1. Примеры циклов в матрице

Циклы в матрице обладают рядом свойств, которые описываются леммами 2.1 — 2.3.

Лемма 2.1. *Пусть дана матрица, состоящая из* n *строк и* m *столбцов, для которой выполнено неравенство* $n{\cdot}m \geq n + m$. *Если в этой матрице произвольно отмечено* $n+m$ *клеток, то существует цикл с вершинами в отмеченных клетках* (*быть может не во всех*).

Доказательство. Докажем лемму по индукции, которую будем вести по величине $n + m$. Проверим утверждение леммы для случая $n + m = 4$. Для меньших значений не может выполняться неравенство

$n{\cdot}m \geq n + m$. Единственной возможностью здесь является $n = m = 2$. Легко увидеть цикл, лежащий в четырех отмеченных клетках. Предположим, что утверждение доказано для всех матриц, для которых $n + m < p$ при некотором натуральном p. Докажем, что наше утверждение справедливо и при $n + m = p$. Действительно, может представиться лишь две возможности

1) для каждой отмеченной клетки в строке и в столбце матрицы находится другая отмеченная клетка;

2) есть хотя бы одна отмеченная клетка, в столбце или строке которой других отмеченных клеток нет.

Рассмотрим вначале случай 2). Если есть отмеченная клетка, для которой нет других отмеченных клеток в строке (или столбце), то ее можно удалить, вычеркивая эту строку (или столбец). Получим матрицу, для которой $n + m < p$. По индуктивному предположению должен существовать цикл с вершинами в отмеченных клетках этой новой матрицы, и утверждение доказано. Если же имеет место случай 1), то из любой отмеченной клетки можно перейти в другую отмеченную клетку по строке (или столбцу). Из новой отмеченной клетки можно перейти в следующую и т. д. Поскольку клеток конечное число, то через ряд шагов мы вернемся в исходную клетку, то есть получим искомый цикл. Лемма 1 доказана.

Лемма 2.2. *Количество вершин любого цикла в матрице четно.*

Доказательство. Поскольку в каждой вершине сходится ровно два взаимно перпендикулярных звена, то при переходе через вершину звено поворачивается на угол $\pm\ \pi/2$. В результате обхода цикла суммарный угол поворота должен быть равен $2\pi q$, где $q \in \mathrm{Z}$. Обозначим через k количество вершин, в которых звено поворачивается на угол $\pi/2$, а через l — количество вершин, в которых оно поворачивается на угол $-\pi/2$. Тогда

$$\frac{\pi}{2}k - \frac{\pi}{2}l = 2\pi q \quad \textit{или} \quad k - l = 4q.$$

Поэтому количество всех вершин $k + l = (k - l) + 2l$ будет четным. Лемма 2 доказана.

Определение 2.1. *Цикл в матрице называется означенным, если каждой вершине его сопоставлен знак* «+» *или* «–», *причем при обходе цикла знаки чередуются, т. е. если вершина имеет какой-то знак, то соседним вершинам сопоставляется противоположный знак* (*рис.* 2.1, *а*), *б*)).

Из леммы 2.2 следует, что любой цикл в матрице можно сделать означенным, причем двумя различными способами.

Лемма 2.3. *Во всяком означенном цикле число положительных вершин, лежащих в каждой строке или столбце, равняется количеству отрицательных вершин в этой строке или столбце.*

Доказательство. Для каждой вершины в строке (столбце) имеется единственная соседняя вершина и для двух различных вершин в строке (столбце) соседние вершины различны. Рассмотрим положительные вершины в какой-нибудь строке (столбце). Для них в этой строке (столбце) имеется столько же соседних вершин. Они должны быть отрицательными
по определению означенного цикла. Отсюда следует справедливость леммы 3.

2.3. Преобразование решения системы ограничений транспортной задачи сдвигом по означенному циклу

Пусть имеется некоторое решение системы ограничений транспортной задачи. Запишем это решение в виде матрицы перевозок. Пусть в матрице перевозок задан некоторый означенный цикл.

Определение 2.2. *Сдвигом по означенному циклу на число x матрицы перевозок называется такое преобразование этой матрицы, при котором меняются лишь элементы в клетках, где находятся вершины цикла, причем элемент в клетке с положительной вершиной увеличивается на число x, а элемент в клетке с отрицательной вершиной уменьшается на это же число.*

Теорема 2.1. *При сдвиге по означенному циклу на число x решение системы ограничений транспортной задачи переходит снова в решение этой же системы ограничений.*

Доказательство. Рассмотрим горизонтальное уравнение, в котором слева стоит сумма элементов некоторой строки матрицы. Некоторые из этих элементов увеличиваются на число x, а некоторые уменьшаются на это же число. В силу леммы 2.3 количество положительных вершин в строке равно количеству отрицательных. Поэтому, если уравнение удовлетворялось до сдвига, то оно будет удовлетворяться и после сдвига, так как его левая часть не изменится. То же можно сказать и о каждом вертикальном уравнении. Теорема доказана.

Лемма 2.4. *Если матрица перевозок является базисным решением системы ограничений транспортной задачи, то не существует цикла с вершинами только в базисных клетках.*

Доказательство. Переменные, отвечающие базисным клеткам, выражаются через переменные свободных клеток. Поэтому, если свободные переменные равны нулю, то базисные определяются однозначно. Предположим, что утверждение леммы неверно, т. е. существует цикл с вершинами в базисных клетках. Означим произвольно этот цикл и произведем сдвиг по нему на произвольное число x. Матрица останется решением системы ограничений транспортной задачи. При этом базисные переменные меняются, а свободные нет. Этого не может быть, так как базисные переменные определяются свободными однозначно. Лемма 4 доказана.

Определение 2.3. *Циклом пересчета для данной свободной клетки называется означенный цикл, одна из вершин которого находится в данной свободной клетке, а остальные — в базисных клетках. Цикл пересчета означивается так, что вершине в данной свободной клетке приписывается знак* «+».

Теорема 2.2. *Для каждой свободной клетки опорного решения системы ограничений транспортной задачи существует цикл пересчета и притом только один.*

Доказательство. Рассмотрим произвольную свободную клетку и добавим к ней $m+n-1$ базисную клетку. Получим $m+n$ отмеченных клеток. Согласно лемме 2.1 существует цикл с вершинами в этих отмеченных клетках. Поскольку по лемме 2.4 нет цикла с вершинами в базисных клетках, то этот цикл должен иметь вершину в данной свободной клетке, т. е. существование цикла пересчета доказано. Если бы существовало два различных цикла пересчета для данной свободной клетки, то из них можно было бы скомбинировать цикл с вершинами только в базисных клетках, а это невозможно. Теорема доказана.

Цикл пересчета является инструментом с помощью, которого производится переход от одного опорного решения к другому. Действительно, произведем сдвиг по циклу пересчета для некоторой свободной клетки на число x равное наименьшей из перевозок, стоящих в клетках с вершинами, имеющими знак “–”. При этом, по крайней мере, одна из прежних базисных переменных имеет значение 0 и мы можем перевести ее в число свободных переменных, сделав вместо нее базисной ту переменную, которая была свободной.

Решение, полученное в результате сдвига по циклу пересчета на выбранное число x, по-прежнему будет опорным, но отвечающим уже другому базисному виду системы ограничений.

Пример 2.2. Преобразуем первое опорное решение примера 1, вводя в качестве новой базисной переменной x_{12}.

Составляем цикл пересчета для клетки x_{12}. Минимальное из чисел, стоящих в клетках с отрицательными вершинами, равно 10. Производя сдвиг по циклу на число 10, получим новое опорное решение.

Запасы	Потребности		
	40	30	10
10	10 −	+	
20	20		
50	10 +	− 30	10

Запасы	Потребности		
	40	30	10
10		10	
20	20		
50	20	20	10

2.4. Нахождение коэффициентов при свободных переменных в базисном виде системы ограничений транспортной задачи

Пусть система ограничений транспортной задачи приведена к базисному виду. Базисная переменная x_{kl} выражается через свободные переменные, причем коэффициенты этого выражения определяет следующая

Теорема 2.3. *Коэффициент при свободной переменной* x_{ij} *в выражении для базисной переменной* x_{kl} *может принимать лишь три возможных значения* 0, 1 *и* –1. *При этом*

а) *этот коэффициент равен* 0, *если в клетке переменной* x_{kl} *нет вершины цикла пересчета для переменной* x_{ij};

б) *этот коэффициент равен* 1, *если в клетке переменной* x_{kl} *положительная вершина цикла пересчета для переменной* x_{ij};

в) *этот коэффициент равен* –1, *если в клетке переменной* x_{kl} *отрицательная вершина цикла пересчета для переменной* x_{ij};

Доказательство. Пусть дан некоторый базисный вид и соответствующее базисное решение системы ограничений транспортной задачи. Рассмотрим базисную переменную x_{kl} и свободную переменную x_{ij}. В базисном решении x_{ij}=0. Построим цикл пересчета для клетки x_{ij} и произведем сдвиг рассматриваемого решения на некоторое число x_0. Получим новое решение системы ограничений транспортной задачи, в котором изменилась лишь одна свободная переменная x_{ij}. Поэтому базисная переменная получает новое значение $x_{kl}^{(1)} = x_{kl} + qx_0$, где q — является коэффициентом, с которым переменная x_{ij} входит в выражение для x_{kl}. С другой стороны, это новое решение получено сдвигом по означенному циклу. Поэтому q=0, если в клетке x_{kl} нет вершины цикла пересчета для клетки x_{ij}; $q = 1$, если в клетке x_{kl} — положительная вершина и $q = -1$, если в клетке x_{kl} — отрицательная вершина. Теорема доказана.

2.5. Выражение целевой функции транспортной задачи через свободные переменные

В выражении целевой функции транспортной задачи $z = \sum_{i,j} c_{ij} x_{ij}$ суммирование распространяется на все клетки. Если система ограничений приведена к базисному виду, то базисные переменные можно из этого выражения исключить. Получим выражение

$$z = \gamma_0 + \sum_{\substack{\text{по своб.} \\ \text{перем.}}} \gamma_{ij} x_{ij} \ ,$$

где суммирование производится по свободным клеткам. Для дальнейшего важно знать коэффициенты γ_{ij}. Назовем *алгебраической суммой стоимостей по означенному циклу* в таблице данных транспортной задачи разность между суммой стоимостей в клетках с положительными вершинами и суммой стоимостей в клетках с отрицательными вершинами этого означенного цикла.

Теорема 2.4. *Для любого базисного вида системы ограничений транспортной задачи коэффициент γ_{ij} в выражении через свободные переменные целевой функции этой задачи равен алгебраической сумме стоимостей по циклу пересчета для свободной клетки x_{ij}.*

Доказательство. Зафиксируем некоторую свободную переменную x_{ij} и подсчитаем коэффициент γ_{ij} для нее. В первоначальное выражение для z переменная x_{ij} входит непосредственно с коэффициентом c_{ij}, а также посредством базисных переменных. Если учесть теорему 2.3, то становиться очевидным, что каждая базисная переменная x_{kl}, в клетке которой есть вершина цикла пересчета для переменной x_{ij}, вносит в коэффициент γ_{ij} слагаемое $\pm c_{kl}$. Причем, если в клетке x_{kl} находится положительная вершина, то это слагаемое равно c_{kl}, а в случае отрицательной вершины — $(-c_{kl})$. Теорема 4 доказана.

2.6. Распределительный метод

Этот метод, по сути, является симплекс-методом для транспортной задачи. При этом вычисления организуются без симплекс-таблиц, а используются циклы пересчета для всевозможных свободных клеток опорных решений. Пусть имеется первое опорное решение. В нем свободные переменные равны нулю. Мы переходим к другому опорному решению, увеличивая от нуля некоторую свободную переменную. Поскольку мы решаем задачу на минимум, то для уменьшения целевой функции нужно выбрать для увеличения от нуля ту свободную переменную, для которой $\gamma_{ij} < 0$. Увеличение выбранной переменной x_{ij} нельзя производить неограниченно, поскольку решение может стать недопустимым. Граница увеличения x_{ij} определяется способом перехода к следующему опорному решению. Этот переход мы производим с помощью цикла пересчета для клетки x_{ij}. Для того, чтобы сдвиг по циклу не привел к недопустимому решению, нужно следить за тем, чтобы базисные переменные не стали отрицательными. Для этого просматривают клетки цикла пересчета с отрицательными вершинами и выбирают среди них ту, в которой стоит минимальная величина перевозки. Эта клетка и определяет базисную переменную, которую мы выведем из числа базисных, вводя вместо нее переменную x_{ij}. Порядок работы по распределительному методу можно описать следующим образом.

1. Находится первое опорное решение одним из рассмотренных выше способов.
2. Для каждой свободной клетки строим цикл пересчета и определяем коэффициент γ_{ij} как алгебраическую сумму стоимостей по циклу пересчета. Если все коэффициенты $\gamma_{ij} \geq 0$, то задача решена и найденное опорное решение является точкой минимума задачи. В противном случае переходим к пункту 3.
3. Выбираем свободную клетку с отрицательным значением γ_{ij} и рассматриваем величины перевозок в клетках с отрицательными вершинами цикла пересчета для x_{ij}. Из этих перевозок выбираем наименьшую, которую обозначим через x.

4. Производим сдвиг по циклу пересчета для клетки, выбранной в пункте 3 на число x. Получаем новое опорное решение, на котором значение целевой функции будет меньше, чем на старом.
5. Переходим к пункту 2, то есть снова подсчитываем коэффициенты γ_{ij} для новых свободных клеток. Описанные шаги производятся до тех пор, пока на очередном шаге все коэффициенты γ_{ij} не станут неотрицательными.

2.7. Метод потенциалов

В распределительном методе наиболее трудоемкой частью является подсчет коэффициентов γ_{ij} для каждой свободной клетки. Существует метод, позволяющий подсчитывать эти коэффициенты без составления циклов пересчета. Этот метод называется методом потенциалов.

Каждому пункту отправления (базе) A_i сопоставим величину u_i (потенциал пункта отправления), каждому потребителю B_j сопоставим потенциал v_j. Будем находить эти потенциалы из того условия, что *для каждой базисной клетки x_{kl} сумма потенциалов равна стоимости в этой клетке*

$$u_k + v_l = c_{kl}.$$

Поскольку базисных клеток $m + n - 1$, то мы имеем систему $m+n-1$ уравнений с $m + n$ неизвестными. Эта система имеет бесконечно много решений. Если положить один из потенциалов равным нулю, то мы легко найдем некоторое решение этой системы уравнений.

Теорема 2.5. *Для любой свободной клетки x_{ij} алгебраическая сумма стоимостей по циклу пересчета γ_{ij} равна разности между стоимостью в этой клетке и суммой соответствующих потенциалов*

$$\gamma_{ij} = c_{ij} - (u_i + v_j)$$

Доказательство. Будем обходить цикл пересчета, выйдя из свободнойклетки x_{ij}, например, по столбцу. Сначала мы попадем в базисную клетку x_{kj}, расположенную в k-ой строке и в том же j-ом столбце, что и клетка x_{ij}. Затем, двигаясь по k-ой строке, мы перейдем из клетки x_{kj} в базисную клетку x_{kl}, расположенную в l-ом столбце и в той же k-ой строке, что и клетка x_{kj}, и т. д.. Наконец, выйдя из некоторой базисной клетки x_{iv}, лежащей в v-ом столбце, мы вернемся, и при том по i-ой строке, в свободную клетку x_{ij}, так как вышли из нее по столбцу. Получаем следующую последовательность клеток (стрелки указывают направление обхода)

$$x_{ij} \to x_{kj} \to x_{kl} \to x_{sl} \to \ldots \to x_{uv} \to x_{iv}.$$

Составим алгебраическую сумму стоимостей γ_{ij} по этому циклу пересчета.

$$\gamma_{ij} = c_{ij} - c_{kj} + c_{kl} - c_{sl} + \ldots + c_{uv} - c_{iv}.$$

Все клетки, кроме x_{ij}, в этом цикле являются базисными. Поэтому все стоимости, кроме c_{ij}, равны суммам соответствующих потенциалов и $\gamma_{ij} = c_{ij} - (u_k + v_j) + (u_k + v_l) - (u_s + v_l) + \ldots + (u_u + v_v) - (u_i + v_v)$.

После раскрытия скобок все потенциалы, кроме u_i и v_j уничтожаются и $\gamma_{ij} = c_{ij} - (u_i + v_j)$, что и требовалось доказать.

Опираясь на порядок работы по распределительному методу, сформулируем правила работы по методу потенциалов.

1. Нахождение первого опорного решения.
2. Нахождение потенциалов пунктов отправления и назначения.
3. Нахождение коэффициентов $\gamma_{ij} = c_{ij} - (u_i + v_j)$ для свободных клеток. Если все $\gamma_{ij} \geq 0$, то данное опорное решение является оптимальным и задача решена. В противном случае переходим к пункту 4).
4. Выбор свободной клетки с $\gamma_{ij} < 0$. Построение цикла пересчета для выбранной клетки. Сдвиг по этому циклу опорного решения на величину минимальной перевозки среди тех, что стоят в клетках с отрицательными вершинами цикла пересчета. Получение нового опорного решения. Переход к пункту 2.
5. Операции, указанные в пунктах 1) — 4) повторяются до тех пор, пока величины γ_{ij} для всех свободных клеток не станут неотрицательными. Соответствующее опорное решение будет точкой минимума транспортной задачи.

Пример 2.3. Решить следующую транспортную задачу, находя первое опорное решение методом наименьшей стоимости.

		v_1	v_2	v_3
	Запасы	Потребности		
		40	30	10
u_1	10	5	7	1 10
u_2	20	2 +	1 − 20	4
u_3	50	6 40 −	3 + 10	2 0

Составляем и решаем систему уравнений для потенциалов по базисным клеткам:

$$\begin{cases} u_1+v_3=1, \\ u_2+v_2=1, \\ u_3+v_1=6, \\ u_3+v_2=3, \\ u_3+v_3=2; \end{cases} \quad \begin{cases} u_1=-1, \\ u_2=-2, \\ u_3=0; \end{cases} \quad \begin{cases} v_1=6, \\ v_2=3, \\ v_3=2. \end{cases}$$

Здесь мы положили u_3 = 0 и решили систему. Для каждой свободной клетки вычисляем γ_{ij}.

$$\gamma_{11}=5-(-1+6)=0;$$
$$\gamma_{12}=7-(-1+3)=5;$$
$$\gamma_{21}=2-(-2+6)=\underline{-2};$$
$$\gamma_{23}=4-(-2+2)=4.$$

Имеется одна свободная клетка x_{21} с отрицательным γ_{21} = −2. Строим цикл пересчета для этой клетки. Наименьшее число, стоящее в клетках с отрицательными вершинами, равно 20. Производим сдвиг на 20 по этому циклу пересчета. Получаем новое опорное решение.

		v_1	v_2	v_3
	Запасы	Потребности		
		40	30	10
u_1	10	5	7	1 10
u_2	20	2 20	1	4
u_3	50	6 20	3 30	2 0

Снова составляем систему уравнений для потенциалов:

$$\begin{cases} u_1+v_3=1, \\ u_2+v_1=2, \\ u_3+v_1=6, \\ u_3+v_2=3, \\ u_3+v_3=2; \end{cases} \quad \begin{cases} u_1=-1, \\ u_2=-4, \\ u_3=0; \end{cases} \quad \begin{cases} v_1=6, \\ v_2=3, \\ v_3=2. \end{cases}$$

Снова подсчитываем коэффициенты γ_{ij} для свободных клеток:

$$\gamma_{11}=5-(-1+6)=0;$$

$$\gamma_{12}=7-(-1+3)=5;$$
$$\gamma_{21}=1-(-4+3)=2;$$
$$\gamma_{23}=4-(-4+2)=6.$$

Здесь все коэффициенты неотрицательны. Это означает, что последнее опорное решение является оптимальным. При этом $z_{min}=1\cdot10+2\cdot20+6\cdot20+3\cdot30+2\cdot0=260$. Координаты точки минимума: $x_{11}=0$; $x_{12}=0$; $x_{13}=10$; $x_{21}=20$; $x_{22}=0$; $x_{23}=0$; $x_{31}=20$; $x_{32}=30$; $x_{33}=0$.

2.8. Об одном свойстве закрытой транспортной задачи

Матрицу стоимостей транспортной задачи можно определенным образом изменить, не меняя оптимального плана перевозок. Справедлива следующая

Теорема 2.6. *Если к элементам произвольного столбца или произвольной строки матрицы стоимостей транспортной задачи прибавить одно и то же произвольное число, то целевая функция задачи изменится на постоянное слагаемое. Таким образом, получаются две транспортные задачи с разными матрицами стоимостей, но одинаковыми оптимальными планами.*

Доказательство. Пусть дана транспортная задача с матрицей стоимостей $C=(c_{ij})$. Рассмотрим транспортную задачу с другой матрицей стоимостей $C_1=(c_{ij}+\alpha_i+\beta_j)$, где α_i ($i=1, \ldots, n$) и β_j ($j=1, \ldots, m$) — произвольные числа. Запишем целевую функцию второй задачи

$$z_1=\sum_{i,j}(c_{ij}+\alpha_i+\beta_j)x_{ij}=\sum_{i,j}c_{ij}x_{ij}+\sum_{i=1}^{n}\alpha_i\sum_{j=1}^{m}x_{ij}+\sum_{j=1}^{m}\beta_j\sum_{i=1}^{n}x_{ij}.$$

Учитывая горизонтальные и вертикальные уравнения системы ограничений, получим

$$z_1=\sum_{i,j}c_{ij}x_{ij}+\sum_{i=1}^{n}\alpha_i a_i+\sum_{j=1}^{m}\beta_j b_j=z+\sum_{i=1}^{n}\alpha_i a_i+\sum_{j=1}^{m}\beta_j b_j,$$

что и доказывает утверждение теоремы. Очевидно, что оптимальные планы перевозок задач с матрицами перевозок C и C_1 совпадают.

Сформулированное свойство транспортной задачи позволяет использовать преобразование матрицы перевозок для нахождения оптимального опорного решения. Можно так изменить элементы матрицы стоимостей, чтобы они, оставаясь неотрицательными, стали равными нулю в базисных клетках некоторого опорного плана. На этом основан так называемый венгерский метод решения транспортной задачи (см. [11], с. 180).

2.9. Открытые транспортные задачи

Выше мы предполагали, что общие запасы A равны общим потребностям B. В случае $A \neq B$ формулировка транспортной задачи различна в случаях $A < B$ и $A > B$. В случае $A < B$ мы можем обеспечить вывоз груза из пунктов отправления, однако удовлетворить все пункты назначения (потребителей) мы не можем. При математической формулировке горизонтальные уравнения остаются без изменений, а вертикальные уравнения превратятся в неравенства типа ”≤“. Транспортная задача будет формулироваться так:

$$z = \sum_{(i,j)} c_{ij} x_{ij} \to \min;$$

$$\begin{cases} x_{11} + x_{12} + \ldots + x_{1n} = a_1, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ x_{m1} + x_{m2} + \ldots + x_{mn} = a_m; \end{cases}$$

$$\begin{cases} x_{11} + x_{21} + \ldots + x_{m1} \le b_1, \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ x_{1n} + x_{2n} + \ldots + x_{mn} \le b_n; \end{cases}$$

$$x_{ij} \ge 0, i = \overline{1,m};\ j = \overline{1,n}.$$

В случае $A > B$ горизонтальные уравнения превращаются в неравенства, то есть система ограничений состоит из первой группы неравенств и второй группы равенств. Если в условиях открытой транспортной задачи все пункты отправления и назначения равноправны, то такая задача легко сводится к закрытой транспортной задаче. Равноправие пунктов понимается в том смысле, что нет потребителей, которых необходимо обязательно удовлетворить (при $A < B$), и нет баз, которые необходимо обязательно освободить от груза (при $A > B$).

Для сведения нашей задачи к закрытой транспортной задаче в случае $A < B$ введем фиктивный пункт отправления A_{m+1} с запасами

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

Рассмотрим закрытую задачу с пунктами отправления A_1, A_2, ... A_m, A_{m+1} и с теми же пунктами назначения B_1, B_2, ..., B_n. В клетках, отвечающих фиктивному пункту A_{m+1}, стоимости будем считать равными 0. Решение этой закрытой задачи даст решение нашей исходной задачи. Перевозки, отвечающие фиктивному пункту, имеют смысл недопоставок груза соответствующим потребителям. В случае A>B следует ввести фиктивный пункт назначения B_{n+1} с потребностями $B_{n+1} = A - B$ и положить стоимости в клетках, связанных с этим пунктом, равными 0. Фиктивные перевозки будут означать количество груза, которое останется на соответствующих базах.

2.10. Открытые транспортные задачи с неравноправными пунктами. Блокирование клеток

В некоторых случаях в формулировку транспортной задачи входят также условия обязательного удовлетворения некоторых выделенных пунктов потребления или обязательного освобождения от груза некоторых баз. Первые условия возникают при $A < B$, а вторые — при $A > B$. В математической формулировке подобные условия означают, что в подсистеме неравенств некоторые неравенства должны быть на самом деле равенствами. Такие задачи с выделенными пунктами также могут быть сведены к закрытым задачам. Соответствующий прием сведения называется *блокированием клеток*.

Поскольку есть выделенные пункты, то в случае $A < B$ эти пункты надо удовлетворить за счет реальных баз, а в случае $A > B$ требуется вывозить груз из выделенных баз в реальные пункты потребления. Таким образом, в задачах с выделенными пунктами запрещается рассматривать решения, в которых отличны от нуля некоторые переменные, связанные с фиктивными пунктами. Исключение этих планов и осуществляется приемом, который называется блокированием клеток. При этом в закрытой транспортной задаче, построенной так, как описано выше, стоимости, отвечающие выделенным пунктам и фиктивным пунктам, полагаются равными достаточно большому числу $M > 0$. В остальных клетках, связанных с фиктивными пунктами, стоимости остаются равными нулю. Из-за большой стоимости выделенные клетки как бы блокируются, то есть в оптимальном решении закрытой задачи соответствующие перевозки получаются равными нулю. При этом в случае $A < B$ выделенные пункты назначения удовлетворяются за счет реальных баз, а в случае $A > B$ груз из выделенных баз вывозится целиком в реальные пункты потребления.

Пример 2.4. Свести к закрытой транспортной задаче следующую открытую задачу при дополнительном условии обязательного удовлетворения пункта B_1. Тот факт, что пункт B_1

выделен, условно записывается в виде звездочки над столбцом B_1 (рис. 2.2). Эта задача является открытой, так как общие потребности $B = 120$ больше общих запасов $A = 100$. Вводим фиктивный пункт отправления A_4 с запасами $a_4 = 120 - 100 = 20$. В клетке, соответствующей пунктам B_1 и A_4, стоимость полагаем $M = 100$. Остальные стоимости в фиктивных клетках остаются равными нулю (таких клеток всего одна). Получаем обычную закрытую транспортную задачу, которую можно решить методом потенциалов.

Запасы	Потребности	
	70 *	50
30	1	2
20	4	6
50	7	1

Запасы	Потребности	
	70	50
30	1	2
20	4	6
50	7	1
20	100	0

Рис. 2.2. Сведение открытой транспортной задачи к закрытой

Наконец, если строго выделенных пунктов нет, но недополучение груза пунктами назначения (при $A < B$) или остаток на базах груза (при $A > B$) приводит к определенным убыткам, то стоимости в фиктивных пунктах полагают равными этим убыткам.

Пример 2.5. Рассмотрим транспортную задачу предыдущего примера при другом дополнительном условии. Предположим, что недополучение единицы груза пунктом B_1 приводит к убытку в 8 единиц стоимости, а для пункта B_2 этот убыток равен 10 единицам стоимости. Как и раньше вводим фиктивный пункт отправления A_4 с запасами $a_4 = 20$. В клетках фиктивного пункта стоимости полагаем равными соответственно $c_{41} = 8$, $c_{42} = 10$. Получаем обычную закрытую транспортную задачу

Запасы	Потребности	
	70	50
30	1	2
20	4	6
50	7	1
20	8	10

Отметим, что прием *блокирования клеток* применяется для решения транспортных задач (закрытых или открытых) при дополнительном условии невозможности перевозок по некоторым коммуникациям. В некоторых случаях перевозки груза из пункта A_i в пункт B_j не могут быть осуществлены. При определении оптимальных планов таких задач полагают, что стоимость перевозки единицы груза, отвечающая клетке x_{ij} является достаточно большой величиной M. После этого известными методами находится решение новой транспортной задачи.

2.11. О других типах транспортных и подобных им задач

В реальных условиях формулировка транспортной задачи может осложняться дополнительными ограничениями, которые меняют тип задачи и требуют значительной модификации метода ее решения.

Например, коммуникации, связывающие пункты отправления и назначения могут иметь ограниченные пропускные способности, т. е. по этим коммуникациям за рассматриваемое время можно перевезти ограниченное количество груза. Возникает *транспортная задача с ограничениями по пропускной способности*, в которой добавляются дополнительные ограничения вида

$$0 \leq x_{ij} \leq d_{ij},$$

где d_{ij} — пропускная способность коммуникации из A_i в B_j. Такая задача значительно отличается от обычной транспортной задачи. В частности, она не всегда имеет решение. Если сумма пропускных способностей коммуникаций, идущих в данный пункт назначения меньше потребности в этом пункте, то задача неразрешима. Существует модификация метода потенциалов, которая позволяет либо решить задачу с ограничениями по пропускной способности, либо установить ее неразрешимость.

Груз может обрабатываться на промежуточных станциях (подвергаться перевалке). Это ведет к дополнительным ограничениям, связанным с временем, которое необходимо для обработки груза. При этом транспортная задача значительно усложняется.

В приложениях часто возникает *транспортная задача по критерию времени*, в которой оптимизируется не стоимость перевозок, а время, необходимое на эти перевозки. Такая задача вообще не является задачей линейного программирования. В ней вместо матрицы стоимостей задается матрица $T=(t_{ij})$, где t_{ij} — время, необходимое для перевозки груза из пункта A_i в пункт B_j. Предполагается, что оно не зависит от объема перевозимой продукции, но в случае нулевой перевозки считается равной нулю. Целевая функция задачи имеет вид

$z = \max_{i,j} \; t_{ij}$. При обычных ограничениях ставиться задача на минимум этой функции. Очевидно, что за время z будут осуществлены все перевозки по заданному допустимому плану. Требуется выбрать такой допустимый план, при котором время осуществления всех перевозок будет минимальным. Решение этой задачи может быть сведено к последовательному решению нескольких задач линейного программирования.

К модели транспортной задачи иногда приводят задачи, по своему содержанию никак не связанные с транспортом, с планированием перевозок. В таких случаях говорят, что задача может быть сформулирована в терминах транспортной задачи. Примером такой задачи может служить задача о назначениях, о которой пойдет речь в главе о дискретном программировании. Поскольку для транспортной задачи имеются эффективные алгоритмы решения, всегда полезно, если возможно, сформулировать данную задачу в терминах транспортной задачи.

Наконец многие практические задачи приводят к модели, по отношению к которой обычная транспортная задача является частным случаем. Как и в транспортной задаче, здесь ограничения делятся на две группы, такие, что каждая переменная входит явно лишь в одно ограничение каждой группы, то есть сохраняется одна из основных особенностей транспортной задачи.

Однако в отличие от нее в одной из групп ограничений допускаются произвольные, не обязательно единичные, ненулевые коэффициенты при входящих в ограничения переменных. Такие задачи называются *распределительными задачами*. Для них разработаны различные алгоритмы решения, хотя и более сложные, чем для транспортной задачи, однако значительно более простые, чем общие методы линейного программирования.

Отметим, что нами рассматривалась транспортная задача в матричной постановке. Существуют сетевые постановки транспортной задачи. Подробно с задачами транспортного типа, в том числе, и в сетевой постановке можно ознакомиться по книге [11].

Контрольные вопросы и задачи для самостоятельного решения

1. Как формулируется транспортная задача? Что такое матрица перевозок? Как выглядит математическая модель закрытой транспортной задачи?
2. Как записать транспортную задачу в форме таблицы данных?

3. Нахождение первого опорного решения системы ограничений транспортной задачи. В чем заключаются метод северо-западного угла и метод наименьшей стоимости?

4. Что называют циклом в матрице? Какими комбинаторными свойствами обладают циклы?

5. Означенный цикл. Что называют сдвигом по означенному циклу в матрице перевозок? Каким основным свойством обладает этот сдвиг?

6. Что называется циклом пересчета для данной свободной клетки?

7. Как находятся коэффициенты при свободных переменных в базисном виде системы ограничений транспортной задачи?

8. Как находится выражение целевой функции транспортной задачи через свободные переменные для произвольного базисного вида системы ограничений?

9. В чем заключается распределительный метод решения закрытой транспортной задачи?

10. Опишите порядок работы по методу потенциалов.

11. При каких преобразованиях матрицы стоимостей транспортной задачи оптимальный план перевозок не меняется?

12. Открытые транспортные задачи. Как сводится открытая транспортная задача с равноправными пунктами к закрытой задаче?

13. В каких случаях при решении открытой транспортной задачи используется прием блокирования клеток?

14. Какие другие типы транспортных и подобных им задач Вы знаете?

Составить математические модели транспортных задач 2.1 — 2.4.

2.1. Четыре предприятия данного экономического района для производства продукции используют три вида сырья. Потребности в сырье каждого из предприятий соответственно равны 120, 50, 190 и 110 ед. Сырье сосредоточено в трех местах его получения, а запасы его соответственно равны 160, 140 и 170 ед. На каждое из предприятий сырье может завозиться из любого пункта его получения. Тарифы перевозок являются известными величинами и задаются матрицей стоимостей

$$C=\begin{pmatrix} 7 & 8 & 1 & 2 \\ 4 & 5 & 9 & 8 \\ 9 & 2 & 3 & 6 \end{pmatrix}.$$

Требуется составить такой план перевозок, при котором общая стоимость перевозок является минимальной.

2.2. На трех складах оптовой базы сосредоточен однородный груз в количествах 180, 40 и 80 ед. Этот груз необходимо перевести в четыре магазина. Каждый из магазинов должен получить соответственно 120, 40, 60, и 80 ед. груза. Тарифы перевозок единицы груза из каждого из складов во все магазины задаются матрицей

$$C=\begin{pmatrix} 2 & 3 & 4 & 3 \\ 5 & 3 & 1 & 2 \\ 2 & 1 & 4 & 2 \end{pmatrix}.$$

Требуется составить такой план перевозок, при котором общая стоимость перевозок является минимальной.

2.3. Производственное объединение имеет в своем составе три филиала, которые производят однородную продукцию соответственно в количествах 50, 30 и 10 ед. Эту продукцию получают четыре потребителя, расположенные в разных местах. Их потребности соответственно равны 30, 30, 10 и 20 ед. Тарифы перевозок единицы продукции от каждого из филиалов соответствующим потребителям задаются матрицей

$$C=\begin{pmatrix} 1 & 2 & 4 & 1 \\ 2 & 3 & 1 & 5 \\ 3 & 2 & 4 & 4 \end{pmatrix}.$$

Требуется составить такой план прикрепления получателей продукции к ее поставщикам, чтобы общая стоимость перевозок была минимальной.

2.4. Три предприятия производственного объединения производят однородную продукцию в количествах равных соответственно 180, 350 и 20 ед. Эта продукция должна быть поставлена пяти потребителям в количествах равных соответственно 110, 90, 120, 80 и 150 ед. Затраты, связанные с производством и доставкой единицы продукции, задаются матрицей

$$C=\begin{pmatrix} 7 & 12 & 4 & 6 & 5 \\ 1 & 8 & 6 & 5 & 3 \\ 6 & 13 & 8 & 7 & 4 \end{pmatrix}.$$

Требуется составить такой план прикрепления потребителей продукции к ее поставщикам, чтобы общие затраты были минимальными.

2.5 — 2.8. *Записать формулировки транспортных задач* 2.1 — 2.4 *с помощью таблицы данных и найти для каждой из задач первое опорное решение методами северо-западного угла и наименьшей стоимости. Результаты сравнить между собой.*

2.9 — 2.11. *Найти оптимальное решение задач* 2.1 — 2.3 *распределительным методом.*

Задачи 2.12 — 2.21 *решить методом потенциалов. В этих задачах задаются векторы запасов* $\vec{a} = (a_1, a_2, \ldots, a_m)$ *и потребностей* $\vec{b} = (b_1, b_2, \ldots, b_n)$ *, а также матрица стоимостей C.*

2.12

$\vec{a} = (4, 6, 10, 10);$

$\vec{b} = (7, 7, 7, 7, 2);$

$$C = \begin{pmatrix} 16 & 30 & 17 & 10 & 16 \\ 30 & 27 & 26 & 9 & 23 \\ 13 & 4 & 22 & 3 & 1 \\ 3 & 1 & 5 & 4 & 24 \end{pmatrix}.$$

Ответ: z_{min}=191.

2.13

$\vec{a} = (20, 20, 20, 20);$

$\vec{b} = (19, 19, 19, 19, 4);$

$$C = \begin{pmatrix} 15 & 1 & 22 & 19 & 1 \\ 21 & 18 & 11 & 4 & 3 \\ 26 & 29 & 23 & 26 & 24 \\ 21 & 10 & 3 & 19 & 27 \end{pmatrix}.$$

Ответ: z_{min}=684.

2.14

$\vec{a} = (13, 17, 17, 13);$

$\vec{b} = (12, 12, 12, 12, 12);$

$$C = \begin{pmatrix} 20 & 26 & 24 & 26 & 29 \\ 15 & 20 & 29 & 26 & 23 \\ 4 & 10 & 27 & 30 & 7 \\ 9 & 16 & 29 & 20 & 3 \end{pmatrix}.$$

Ответ: z_{min}=868.

2.15

$\vec{a} = (18, 12, 17, 13);$

$\vec{b} = (8, 8, 8, 8, 28);$

$$C = \begin{pmatrix} 21 & 22 & 2 & 13 & 7 \\ 27 & 10 & 4 & 24 & 9 \\ 3 & 16 & 25 & 5 & 4 \\ 28 & 11 & 17 & 10 & 29 \end{pmatrix}.$$

Ответ: z_{min}=392.

2.16.

$\vec{a} = (15, 15, 15, 15);$

$\vec{b} = (11, 11, 11, 11, 16);$

$$C = \begin{pmatrix} 17 & 20 & 29 & 26 & 25 \\ 3 & 4 & 5 & 15 & 24 \\ 19 & 2 & 22 & 4 & 13 \\ 20 & 27 & 1 & 17 & 19 \end{pmatrix}.$$

Ответ: z_{min}=542.

2.17.

$\vec{a} = (15, 15, 19, 11);$

$\vec{b} = (9, 24, 9, 9, 9);$

$$C = \begin{pmatrix} 10 & 17 & 9 & 20 & 30 \\ 13 & 4 & 24 & 26 & 26 \\ 22 & 24 & 30 & 27 & 29 \\ 25 & 12 & 11 & 24 & 23 \end{pmatrix}$$

Ответ: z_{min}=859.

2.18.

$\vec{a} = (21, 19, 15, 25);$

$\vec{b} = (15, 15, 15, 15, 20);$

$$C = \begin{pmatrix} 30 & 24 & 11 & 12 & 25 \\ 26 & 4 & 29 & 20 & 24 \\ 27 & 14 & 14 & 10 & 18 \\ 6 & 14 & 28 & 8 & 2 \end{pmatrix}.$$

Ответ: z_{min}=693.

2.19.

$\vec{a} = (9, 11, 14, 16);$

$\vec{b} = (8, 9, 13, 8, 12);$

$$C = \begin{pmatrix} 5 & 15 & 3 & 6 & 10 \\ 23 & 8 & 13 & 27 & 12 \\ 30 & 1 & 5 & 24 & 25 \\ 8 & 26 & 7 & 28 & 9 \end{pmatrix}$$

Ответ: z_{min}=339.

2.20.

$\vec{a} = (22, 13, 17, 18);$

$\vec{b} = (7, 7, 7, 7, 42);$

$$C = \begin{pmatrix} 9 & 17 & 29 & 28 & 8 \\ 13 & 21 & 27 & 16 & 29 \\ 20 & 30 & 24 & 7 & 26 \\ 11 & 19 & 30 & 6 & 2 \end{pmatrix}$$

Ответ: z_{min}=726.

2.21.

$\vec{a} = (16, 15, 14, 15);$

$\vec{b} = (6, 6, 13, 20, 15);$

$$C = \begin{pmatrix} 30 & 2 & 5 & 6 & 15 \\ 5 & 29 & 9 & 5 & 7 \\ 16 & 24 & 14 & 6 & 26 \\ 13 & 28 & 4 & 25 & 8 \end{pmatrix}$$

Ответ: z_{min}=329.

3. ЭЛЕМЕНТЫ ТЕОРИИ ДВОЙСТВЕННОСТИ

Для любой задачи линейного программирования можно сформулировать некоторую задачу, называемую двойственной. Решение каждой из этой пары задач часто автоматически приводит к решению другой задачи, то есть количество решенных задач увеличивается вдвое. В некоторых случаях одна из двойственных задач проще и ее решать удобнее. Но важнее всего то, что во многих приложениях линейного программирования требуется решать обе двойственные задачи. Поэтому изучение взаимосвязи этих задач очень полезно. Мы кратко опишем эту взаимосвязь, не останавливаясь подробно на обоснованиях.

3.1. Симметрично-двойственные задачи

Рассмотрим задачу линейного программирования в стандартной форме.

$$z = c_0 + c_1 x_1 + \ldots + c_n x_n \to \max;$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m; \end{cases}$$

$$x_1, x_2, \ldots, x_n \ge 0.$$

Назовем эту задачу задачей 1. Построим по задаче 1 задачу 2, которая имеет следующий вид:

$$f = c_0 + b_1 y_1 + \ldots + b_m y_m \to \min;$$

$$\begin{cases} a_{11}y_1 + a_{21}y_2 + \ldots + a_{m1}y_m \ge c_1, \\ a_{12}y_1 + a_{22}y_2 + \ldots + a_{m2}y_m \ge c_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{mn}y_m \ge c_n; \end{cases}$$

$$y_1, y_2, \ldots, y_m \ge 0.$$

Задача 2 построена по задаче 1 с помощью следующих правил:

1. Число переменных задачи 2 равно числу ограничений задачи 1.
2. Матрица системы ограничений задачи 2 является транспонированной по отношению к соответствующей матрице задачи 1, то есть, получена из нее заменой строк столбцами с сохранением порядка.
3. Свободный член целевой функции f задачи 2 равен свободному члену в выражении целевой функции z задачи 1, а коэффициенты

при переменных в выражении для f равны правым частям неравенств системы ограничений задачи 1.

4. Неравенства в системе ограничений задачи 2 имеют смысл "≥", а правые части этих неравенств равны коэффициентам при переменных в целевой функции задачи 1.
5. Задача 2 является задачей на минимум, в то время как задача 1 является задачей на максимум.

Отметим, что если рассматривать задачу 2 как исходную для построения двойственной задачи по сформулированным правилам, то двойственная задача будет равносильна задаче 1. Разумеется, при этом задачу 2 нужно сформулировать в стандартной форме.

Пример 3.1. Исходная задача 1:

$$z = 1 + 2x_1 - 3x_2 \to \max;$$

$$\begin{matrix} y_1 \\ y_2 \\ y_3 \end{matrix}\begin{cases} 2x_1 + 4x_2 \le 10, \\ 4x_1 - 2x_2. \le 1, \\ 5x_1 + 6x_2 \le 12; \end{cases}$$

$$x_1, x_2 \ge 0.$$

Двойственная задача 2:

$$f = 1 + 10y_1 + y_2 + 12y_3 \to \min;$$

$$\begin{cases} 2y_1 + 4y_2 + 5y_3 \ge 2, \\ 4y_1 - 2y_2 + 6y_3 \ge -3, \end{cases}$$

$$y_1, y_2, y_3 \ge 0.$$

Запишем задачу 2 в стандартной форме:

$$f_1 = -1 - 10y_1 - y_2 - 12y_3 \to \max;$$

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

$$y_1, y_2, y_3 \ge 0.$$

Двойственная к ней задача:

$$z_1 = -1 - 2x_1 + 3x_2 \to \min;$$

$$\begin{cases} -2x_1 - 4x_2 \ge -10, \\ -4x_1 + 2x_2. \ge -1, \\ -5x_1 - 6x_2 \ge -12; \end{cases}$$

$$x_1, x_2 \ge 0.$$

Полученная задача равносильна первоначально взятой задаче и совпадает с ней, если перейти к стандартной форме. Отмеченное свойство взаимности является причиной того, что пара задач (1;2) называется парой взаимно-двойственных задач.

3.2. Экономический смысл взаимно-двойственных задач

Рассмотрим задачу *оптимального использования ресурсов* предприятием, которое производит n видов продукции: P_1, $P_{2, \ldots ,}$ P_n и использует для этого m видов ресурсов R_1, R_2, ... , R_m. Будем считать, что ресурсы при необходимости можно продать, не изготавливая продукции. Если b_1, b_2 , ... , b_m — наличные количества ресурсов; a_{ij} — расход ресурса R_i на изготовление единицы продукции P_j, а x_j — плановое количество продукции P_j , то переменные x_1, ... , x_n должны удовлетворять системе неравенств

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_n + \ldots + a_{mn}x_n \le b_m; \end{cases}$$

$$x_1, x_2, \ldots, x_n \ge 0.$$

Для максимизации прибыли от реализации продукции предприятие должно решить задачу на максимум для функции $z = c_1x_1 + \ldots + c_nx_n$,

где c_1, c_2, …, c_n — прибыли от реализации единицы соответствующего вида продукции.

Пусть наше предприятие имеет возможность продать ресурсы некоторому другому предприятию или лицу. Обозначим через y_1, ... , y_m соответствующие цены ресурсов R_1, R_2, ... , R_m. Покупатель заинтересован в понижении цен, но продавец не будет продавать ресурсы, если продажа менее выгодна, чем производство продукции. Станем на точку зрения покупателя и определим наиболее выгодные цены ресурсов с его точки зрения. При полной покупке ресурсов он должен минимизировать общую сумму выплаты за ресурсы:

$$f = b_1y_1 + \ldots + b_my_m \to \min.$$

При этом стоимость ресурсов, идущих на изготовление единицы каждого вида продукции, не должна быть меньше, чем прибыль от ее реализации, иначе продавцу невыгодно продавать ресурсы. Поэтому цены y_1, y_2, ... , y_m должны удовлетворять неравенствам:

$$\begin{cases} a_{11}y_1 + a_{21}y_2 + \ldots + a_{m1}y_m \geq c_1, \\ a_{12}y_1 + a_{22}y_2 + \ldots + a_{m2}y_m \geq c_2, \\ \ldots\ldots\ldots\ \ldots\ldots\ldots\ \ldots\ldots\ldots\ \ldots\ldots\ldots\ . \\ a_{1n}y_1 + a_{2n}y_2 + \ldots + a_{mn}y_m \geq c_n; \end{cases}$$

$$y_1, y_2, \ldots, y_m \geq 0.$$

Таким образом, предприятие-покупатель ресурсов при назначении цен решает задачу линейного программирования двойственную по отношению к задаче, которую решает предприятие-производитель при составлении оптимального плана выпуска продукции. Величины $y_1^0,\ y_2^0, \ldots, y_m^0$, составляющие точку минимума для задачи покупателя, называют *двойственными ценами* (*оценками*) *ресурсов*. Они имеют и более глубокий смысл, о котором речь пойдет ниже.

3.3. Несимметрично-двойственные задачи

Рассмотрим задачу линейного программирования в общем форме. Приведем эту задачу к стандартной форме с помощью рассмотренных выше приемов. Равенства в системе ограничений заменим парами противоположных неравенств, а переменные, принимающие значения произвольного знака, заменим разностями неотрицательных переменных. К полученной задаче в стандартной форме применим сформулированные выше правила построения двойственной задачи. Для простоты рассмотрим эту ситуацию на конкретном примере.

Исходная задача:

$$z = 5 - x_1 + 3x_2 - x_3 \to \min;$$

$$\begin{cases} x_1 - 2x_2 + x_3 = 2, \\ x_1 + x_2 - x_3 \geq -3; \end{cases} \quad x_1, x_2 \geq 0.$$

Переменная x_3 может принимать значения произвольного знака, поэтому полагаем $x_3 = x_3' - x_3'';\ x_3',\ x_3'' \geq 0$. Запишем задачу в стандартной форме.

$$z_1 = -5 + x_1 - 3x_2 + x_3' - x_3'' \to \max;$$

$$\begin{array}{l} y_1 \\ y_2 \\ y_3 \end{array} \begin{cases} x_1 - 2x_2 + x_3' - x_3'' \leq 2, \\ -x_1 + 2x_2 - x_3' + x_3'' \leq -2, \\ -x_1 + x_2 - 3x_3' + 3x_3'' \leq 3; \end{cases}$$

$$x_1, x_2, x_3', x_3'' \geq 0.$$

Построим двойственную задачу:

$$f = -5 + 2y_1 - 2y_2 + 3y_3 \to \min;$$

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

$$y_1, y_2, y_3 \geq 0.$$

Отметим, что переменные y_1 и y_2 входят в эту задачу лишь в комбинации $y_1 - y_2$, а третье и четвертое неравенства системы ограничений двойственной задачи являются взаимно противоположными и могут быть заменены равенством $y_1 - y_2 - 3y_3 = 1$. Обозначая $y_1 - y_2$ через y, можем записать двойственную задачу в виде

$$f = -5 - 2y + 3y_3 \to \min;$$

$$\begin{cases} y - y_3 \geq 1, \\ -2y + y_3 \geq -3, \\ y - 3y_3 = 1; \end{cases} \quad y_3 \geq 0.$$

Отметим, что эта же задача может быть получена из первоначальной задачи, записанной в виде задачи на максимум с неравенством типа " $\leq$ " :

$$z_1 = -5 + x_1 - 3x_2 + x_3 \to \max;$$

$$\begin{matrix} y \\ y_3 \end{matrix} \begin{cases} x_1 - 2x_2 + x_3 = 2, \\ -x_1 + x_2 - 3x_3 \leq 3; \end{cases}$$

$$x_1, x_2 \geq 0,$$

с помощью изложенного выше правила составления симметрично-двойственной задачи со следующими добавлениями:

а) если в исходной задаче в системе ограничений есть равенство, то в двойственной задаче соответствующая переменная не подчинена условию неотрицательности;

б) если в исходной задаче переменная не подчинена условию неотрицательности, то в двойственной задаче соответствующее ограничение является равенством.

В общем случае, если задачу общего вида сформулировать как задачу на максимум, все неравенства в системе ограничений которой имеют смысл "$\leq$", то двойственную задачу можно сформулировать по правилам для симметрично двойственных задач с добавлениями а) и б).

3.4. Первая и вторая теоремы двойственности

Все рассматриваемые ниже утверждения относятся к парам двойственных задач общего вида, то есть *необязательно* симметрично-двойственных.

Теорема 3.1. *Для всякой пары двойственных задач если исходная задача имеет решение, то двойственная задача также имеет решение, и оптимальные значения целевых функций этих задач совпадают, то есть* $z_{max} = f_{min}$. *Если исходная задача не имеет решения по причине неограниченности целевой функции на области допустимых значений, то двойственная задача недопустима.*

Отметим, что первая теорема двойственности не позволяет по решению одной из задач найти точку оптимума другой. С ее помощью отыскивается лишь оптимальное значение целевой функции. Связь между точкой минимума и максимума пары двойственных задач определяет следующая

Теорема 3.2. *Для того, чтобы n-мерный вектор* $\left\{x_1^0, x_2^0, \ldots, x_n^0\right\}$, *удовлетворяющий системе ограничений исходной задачи, и m-мерный вектор* $\left\{y_1^0, y_2^0, \ldots, y_m^0\right\}$ *удовлетворяющий системе ограничений двойственной задачи, были соответственно точкой максимума исходной и точкой минимума двойственной задач, необходимо и достаточно, чтобы выполнялись следующие условия:*

$$\begin{cases} y_1^0\left(a_{11}x_1^0 + a_{12}x_2^0 + \ldots + a_{1n}x_n^0 - b_1\right) = 0, \\ y_2^0\left(a_{21}x_1^0 + a_{22}x_2^0 + \ldots + a_{2n}x_n^0 - b_2\right) = 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y_m^0\left(a_{m1}x_1^0 + a_{m2}x_2^0 + \ldots + a_{mn}x_n^0 - b_m\right) = 0, \\ x_1^0\left(a_{11}y_1^0 + a_{21}y_2^0 + \ldots + a_{m1}y_m^0 - c_1\right) = 0, \\ x_2^0\left(a_{12}y_1^0 + a_{22}y_2^0 + \ldots + a_{m2}y_m^0 - c_2\right) = 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n^0\left(a_{1n}y_1^0 + a_{2n}y_2^0 + \ldots + a_{mn}y_m^0 - c_n\right) = 0. \end{cases}$$

Вторая теорема двойственности содержит равенства, которые называются условиями дополнительной нежесткости. Она позволяет решать следующие задачи:

1) определять оптимальный план одной из двойственных задач, если известно решение другой;

2) проверять, не решая задачи, является ли некоторая совокупность чисел оптимальным планом одной из двойственных задач.

Иногда одну из двойственных задач можно решить графически, а решение двойственной задачи определить с помощью условий дополнительной нежесткости.

Пример 3.2. Пусть дана задача

$$z = x_1 + x_2 + x_3 \to \max;$$
$$\begin{array}{l} y_1 \\ y_2 \end{array}\begin{cases} 2x_1 + 3x_2 - x_3 \le 2, \\ x_1 - x_2 + 2x_3 \le 3; \end{cases}$$
$$x_1, x_2, x_3 \ge 0.$$

Двойственную задачу

$$f = 2y_1 + 3y_2 \to \min;$$
$$\begin{cases} 2y_1 + y_2 \ge 1, \\ 3y_1 - y_2 \ge 1, \\ -y_1 + 2y_2 \ge 1; \end{cases}$$
$$y_1, y_2 \ge 0.$$

можно решить графически (рис.3.1.).

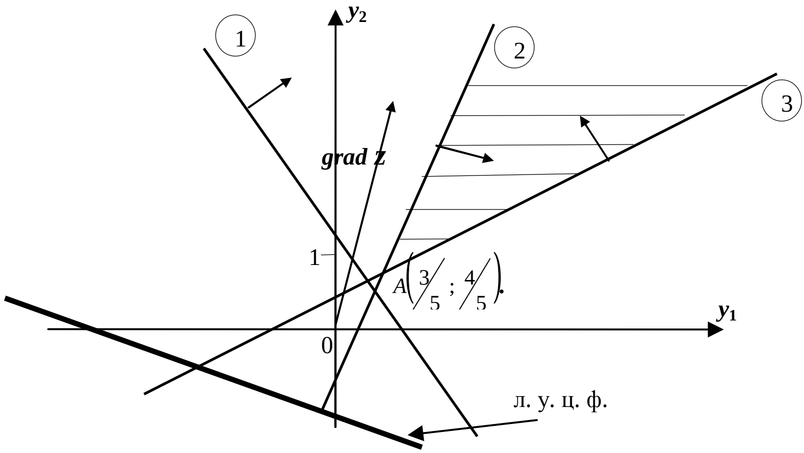


Рис. 3.1. Графическое решение двойственной задачи
.

Точкой минимума двойственной задачи является точка $A\left(\frac{3}{5};\frac{4}{5}\right)$. $z_{\max} = f_{\min} = f(A) = 2\cdot\frac{3}{5} + 3\cdot\frac{4}{5} = \frac{6}{5} + \frac{12}{5} = \frac{18}{5}$. Для определения точки максимума исходной задачи составим условия дополнительной нежесткости:

$$\begin{cases} y_1^0\left(2x_1^0 + 3x_2^0 - x_3^0 - 2\right) = 0, \\ y_2^0\left(x_1^0 - x_2^0 + 2x_n^0 - 3\right) = 0, \\ x_1^0\left(2y_1^0 + y_2^0 - 1\right) = 0, \\ x_2^0\left(3y_1^0 - y_2^0 - 1\right) = 0, \\ x_3^0\left(-y_1^0 + 2y_2^0 - 1\right) = 0. \end{cases}$$

Подставим в эти равенства $y_1^0 = \frac{3}{5}; y_2^0 = \frac{4}{5}$. Учитывая, что четвертое и пятое равенства при этом выполняются при всех x_3^0 и x_2^0, для определения x_1^0, x_2^0, x_3^0 получаем систему уравнений

$$\begin{cases} 2x_1^0 + 3x_2^0 - x_3^0 = 2, \\ x_1^0 - x_2^0 + 2x_3^0 = 3, \\ x_1^0 = 0, \end{cases}$$

которая равносильна системе

$$\begin{cases} 3x_2^0 - x_3^0 = 2, \\ -x_2^0 + 2x_3^0 = 3, \\ x_1^0 = 0. \end{cases}$$

Ее решение: $x_1^0 = 0;\ x_2^0 = \frac{7}{5};\ x_3^0 = \frac{11}{5}$. Таким образом, точка максимума исходной задачи (0; 7/5; 11/5).

Пример 3.3. Будет ли план $\vec{x} = \{1;\ 1;\ 1\}$ оптимальным планом задачи

$$z = x_1 + 2x_2 + 3x_3 \to \max;$$

$$\begin{matrix} y_1 \\ y_2 \end{matrix}\begin{cases} x_1 - x_2 - x_3 \le 1, \\ x_1 + x_2 - x_3 \le 1; \end{cases}$$

$$x_1, x_2, x_3 \ge 0?$$

Составим двойственную задачу:

$$f = y_1 + y_2 \to \min;$$

$$\begin{cases} y_1 + y_2 \ge 1, \\ -y_1 + y_2 \ge 2, \\ -y_1 - y_2 \ge 3; \end{cases}$$

$$y_1, y_2 \ge 0.$$

Составим условия дополнительной нежесткости:

$$\begin{cases} y_1^0\left(x_1^0 - x_2^0 - x_3^0 - 1\right) = 0, \\ y_2^0\left(x_1^0 + x_2^0 - x_n^0 - 1\right) = 0, \\ x_1^0\left(y_1^0 + y_2^0 - 1\right) = 0, \\ x_2^0\left(-y_1^0 + y_2^0 - 2\right) = 0, \\ x_3^0\left(-y_1^0 - y_2^0 - 3\right) = 0. \end{cases}$$

Подставим в эти равенства $x_1^0 = x_2^0 = x_3^0 = 1.$

Получим систему уравнений

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

которая является несовместной. Следовательно, рассматриваемый план оптимальным планом не будет.

3.6. Третья теорема двойственности

Пусть некоторая задача линейного программирования решена и получено некоторое значение z_{max}. При изменении правых частей ограничений b_1, b_2, ... , b_m задача меняется и, в конечном счёте, меняется значение z_{max}. Будем рассматривать z_{max} как функцию величин b_1, b_2, ... , b_m, то есть рассмотрим функцию $z_{max}(b_1, b_2, \dots, b_m)$.

Теорема 3.3. *В оптимальном плане двойственной задачи координата* y_i^0 *численно равна частной производной функции* $z_{max}(b_1, b_2, \dots, b_m)$ *по переменной* b_i, *то есть*

$$\frac{\partial z_{\max}}{\partial b_i} = y_i^0; \quad i = 1, 2, \dots, m..$$

Эта теорема может быть истолкована и следующим образом. Пусть величинам b_1, b_2, ... , b_m придаются приращения Δb_1, Δb_2, ... , Δb_m. Функция $z_{max}(b_1, b_2, ..., b_m)$ получает приращение Δz_{max}. По сформулированной теореме при малых приращениях Δb_1, Δb_2, ... , Δb_m справедливо следующее приближенное равенство

$$\Delta z_{\max} \approx y_1^0 \Delta b_1 + y_2^0 \Delta b_2 + ... + y_m^0 \Delta b_m. \qquad (*)$$

На самом деле функция $z_{max}(b_1, b_2, ..., b_m)$ такова, что при изменении Δb_i в некоторых пределах, равенство (*) является точным. При этом величины Δb_i могут быть и не малыми по сравнению с b_i.

Определить эти пределы изменения величин Δb_i можно с помощью второй теоремы двойственности. Равенство (*) будет точным, если при изменении b_i величины y_i^0 не изменяются. Пределы же изменения приращений Δb_i, при которых оптимальный план двойственной задачи не меняется, можно определить следующим образом. Придадим фиксированным значениям $b_1^0, b_2^0, ..., b_m^0$ неопределенные приращения $\Delta b_1, \Delta b_2, \ldots, \Delta b_m$ и рассмотрим условия дополнительной нежесткости с наращенными значениями $b_i = b_i^0 + \Delta b_i$, в которые подставим старые значения y_i^0. Относительно оптимального плана исходной задачи $x_1^0, x_2^0, \ldots, x_n^0$ получится система линейных уравнений, которую можно решить при определенных условиях с помощью, например, метода обратной матрицы. Получим зависимость

$$x_i^0 = x_i^0\left(\Delta b_1, \Delta b_2, ..., \Delta b_m\right)$$

оптимального плана от приращений. Поскольку x_i^0 должны удовлетворять неравенствам $x_i^0 \geq 0$, $i = 1, \ldots n$, а также системе ограничений исходной задачи, то мы получим систему условий, которым должны удовлетворять Δb_i, чтобы оптимальный план двойственной задачи не менялся. Эта область изменения Δb_i называется *областью устойчивости двойственных оценок*.

Пример 3.4. Рассмотрим пару двойственных задач примера 3.2. Оптимальный план двойственной задачи здесь $y_1^0 = \frac{3}{5}; \; y_2^0 = \frac{4}{5}$.

Наращенные значения правых частей ограничений исходной задачи равны: $b_1 = 2 + \Delta b_1$, $b_2 = 3 + \Delta b_2$. Условия дополнительной нежесткости имеют вид

$$\begin{cases} \dfrac{3}{5}\left(2x_1^0 + 3x_2^0 - x_3^0 - 2 - \Delta b_1\right) = 0, \\ \dfrac{4}{5}\left(x_1^0 - x_2^0 + 2x_3^0 - 3 - \Delta b_2\right) = 0, \\ x_1^0\left(2 \cdot \dfrac{3}{5} + \dfrac{4}{5} - 1\right) = 0, \\ x_2^0\left(\dfrac{9}{5} - \dfrac{4}{5} - 1\right) = 0, \\ x_3^0\left(-\dfrac{3}{5} + \dfrac{8}{5} - 1\right) = 0. \end{cases}$$

Четвёртое и пятое равенства — тождества, а из третьего следует, что $x_1^0 = 0.$ Для x_2^0 и x_3^0 получаем систему равенств

$$\begin{cases} 3x_2^0 - x_3^0 = 2 + \Delta b_1, \\ -x_2^0 + 2x_3^0 = 3 + \Delta b_{2.}. \end{cases}$$

Отсюда $x_2^0 = \dfrac{2}{5}\left(2 + \Delta b_1\right) + \dfrac{1}{5}\left(3 + \Delta b_2\right) = \dfrac{7}{5} + \dfrac{2}{5}\Delta b_1 + \dfrac{1}{5}\Delta b_2,.$

$$x_3^0 = \frac{1}{5}\left(2 + \Delta b_1\right) + \frac{3}{5}\left(3 + \Delta b_2\right) = \frac{11}{5} + \frac{1}{5}\Delta b_1 + \frac{3}{5}\Delta b_2.$$

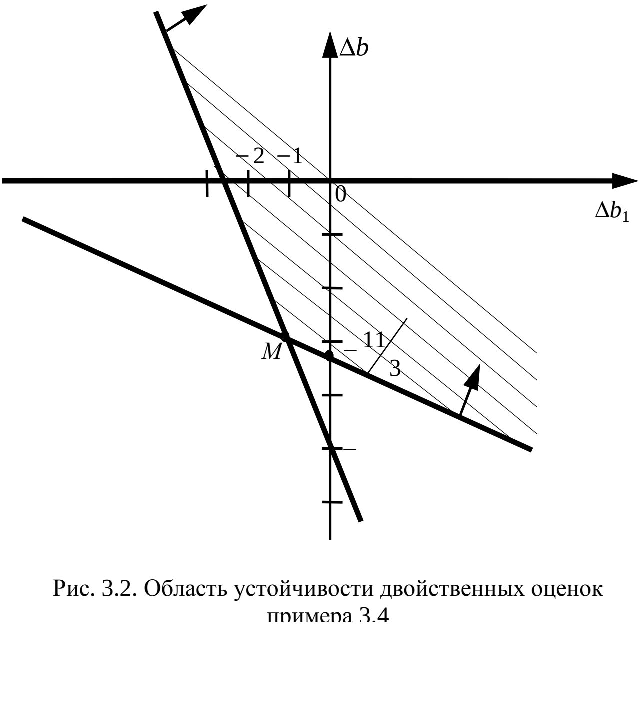


Рис. 3.2. Область устойчивости двойственных оценок примера 3.4

Приращения Δb_1 и Δb_2 должны удовлетворять неравенствам

$$\begin{cases} \frac{7}{5}+\frac{2}{5}\Delta b_1+\frac{1}{5}\Delta b_2 \geq 0, \\ \frac{11}{5}+\frac{1}{5}\Delta b_1+\frac{3}{5}\Delta b_2 \geq 0. \end{cases}$$

Система ограничений исходной задачи с правыми частями $2+\Delta b_1$ и $3+\Delta b_2$ удовлетворяется автоматически. Таким образом, можно изобразить на плоскости область изменения величин Δb_1, Δb_2 в которой оптимальный план двойственной задачи не меняется.

В области, изображенной на рис. 3.2, приращение Δz_{max} может быть найдено по точной формуле:

$$\Delta z_{\max} = y_1^0 \Delta b_1 + y_2^0 \Delta b_2 = \frac{3}{5}\Delta b_1 + \frac{4}{5}\Delta b_2.$$

3.7. Послеоптимизационный анализ

Пусть рассматривается задача оптимального использования ресурсов. Пусть решена и сама эта задача, и двойственная к ней. Знание двойственных оценок $y_1^0, y_2^0, \dots, y_m^0$ позволяет судить о том, насколько ценным в производстве является данный ресурс. Если величина $y_i^0 \neq 0$, то в силу условий дополнительной нежесткости ресурс при реализации оптимального плана выпуска продукции используется полностью. Не полностью используются ресурсы, для которых $y_i^0 = 0$. Таким образом, двойственные оценки ресурсов характеризуют их дефицитность. Различные варианты наращивания ресурсов (или их сокращения) приводят к разным величинам изменения оптимальной прибыли Δz_{max}. Чем больше величина y_i^0, тем больше приращение оптимальной прибыли мы получим при увеличении объема ресурса R_i на 1 единицу. В этом и заключается более глубокий смысл двойственных оценок ресурсов по сравнению с изложенным ранее.

При анализе возможностей расширения производства знание двойственных оценок ресурсов играет большую роль. Обычно такой анализ называют послеоптимизационным, поскольку он выполняется после решения прямой и двойственной задач. Здесь, однако, следует помнить, что для использования двойственных оценок в экономическом анализе нужно знать область их устойчивости, то есть область изменения ресурсов, в которой двойственные оценки ресурсов не меняются.

3.8. Двойственный симплекс-метод для пары симметрично-двойственных задач

Если размерности исходной и двойственной задач велики, то определение решения одной из задач по решению другой с помощью условий дополнительной нежесткости затруднительно. В этом случае применяют двойственный симплекс-метод, который позволяет по последней симплекс-таблице одной из задач прочесть решение двойственной задачи. Этот метод мы рассмотрим сначала в упрощенной форме для пары симметрично-двойственных задач, причем для такой пары, в которой одна из задач после уравнивания неравенств системы ограничений имеет эту систему ограничений в допустимом базисном виде, то есть, может решаться симплекс-методом непосредственно. В следующем параграфе будет рассмотрен подобный метод для произвольной задачи в канонической форме.

Пусть при уравнивании системы ограничений исходной задачи мы ввели переменные $x_{n+1}, x_{n+2}, \dots, x_{n+m}$. Эти переменные играют роль базисных. При уравнивании системы ограничений двойственной задачи вводятся переменные $y_{m+1}, y_{m+2}, \dots, y_{m+n}$, которые также играют роль базисных. Определим соответствие между переменными исходной и двойственной задач, сопоставляя каждой свободной переменной исходной задачи базисную переменную двойственной в соответствии с порядком следования номеров, а каждой базисной переменной исходной задачи — свободную переменную двойственной задачи. Запишем это соответствие в таком виде:

$$\begin{array}{cccccccc} x_1, & x_2, & \dots, & x_n, & x_{n+1}, & x_{n+2}, & \dots, & x_{n+m} \\ \updownarrow & \updownarrow & & \updownarrow & \updownarrow & \updownarrow & & \updownarrow \\ y_{m+1}, & y_{m+2}, & \dots, & y_{m+n}, & y_1, & y_2, & \dots, & y_m. \end{array} \qquad (F)$$

Отметим, что это соответствие переменных устанавливает взаимосвязь базисных решений данной и двойственной задач. Если составлена таблица, аналогичная симплекс-таблице, по базисному решению данной задачи, то соответствующее базисное решение двойственной задачи прочитывается по последней строке этой симплекс-таблицы с использованием построенного выше соответствия переменных. А именно, значение y_i в базисном решении
двойственной задачи прочитывается в строке целевой функции симплекс-таблицы в столбце переменной x_j, соответствующей переменной y_i, по соответствию переменных (F). Эта взаимосвязь базисных решений сохраняется при всевозможных преобразованиях замещения данной таблицы в предположении, что в двойственной

задаче происходит переход к соответствующему новому базисному решению.

Пусть одна из задач решена симплекс-методом и получена последняя симплекс-таблица. Решение (точка оптимума) другой задачи может быть прочитано по последней симплекс-таблице с использованием соответствия переменных (F).

Пример 3.5. Рассматривается пара двойственных задач:

1) $z = -x_1 + x_2 \to \max;$

$$\begin{cases} x_1 - 2x_2 \le 1, \\ -2x_1 + x_2 \le 2, \\ 3x_1 + x_2 \le 3; \end{cases}$$

$x_1 \ge 0, x_2 \ge 0.$

2) $f = y_1 + 2y_2 + 3y_3 \to \min;$

$$\begin{cases} y_1 - 2y_2 + 3y_3 \ge -1, \\ -2y_1 + y_2 + y_3 \ge 1; \end{cases}$$

$y_1 \ge 0, y_2 \ge 0, y_3 \ge 0.$

После уравнивания неравенств задачи имеют следующий вид:

1) $z = -x_1 + x_2 \to \max;$

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

$x_1, x_2, \cdots, x_5 \ge 0.$

2) $f = y_1 + 2y_2 + 3y_3 \to \min;$

$$\begin{cases} y_1 - 2y_2 + 3y_3 - y_4 = -1, \\ -2y_1 + y_2 + y_3 - y_5 = 1; \end{cases}$$

$y_1, y_2, \cdots, y_5 \ge 0.$

Составим соответствие переменных:

$$\begin{array}{ccccc} x_1, & x_2, & x_3, & x_4, & x_5 \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow \\ y_4, & y_5, & y_1, & y_2, & y_3 \end{array}$$

Решая исходную задачу симплекс-методом, получаем следующую последнюю симплекс-таблицу:

Базисные переменные	Свободные члены	x_1	x_2	x_3	x_4	x_5
x_3	28/5	0	0	1	7/5	3/5
x_2	12/5	0	1	0	3/5	2/5
x_1	1/5	1	0	0	–1/5	1/5
z	11/5	0	0	0	4/5	1/5

По первой теореме двойственности и в соответствии со сказанным выше $f_{\min} = \frac{11}{5}$; точка максимума имеет координаты

$$y_1^0 = 0;\ y_2^0 = \frac{4}{5};\ y_3^0 = \frac{1}{5}.$$

Двойственный симплекс-метод можно распространить и на произвольные задачи, однако, при этом он заметно усложняется.

3.9. Метод последовательного уточнения оценок. Обобщенный двойственный симплекс метод

Метод последовательного уточнения оценок применяется к задаче линейного программирования в канонической форме

$$z = c_0 + (\vec{c}, \vec{x}) \to \max,$$

$$A\vec{x} = \vec{b},\ \vec{x} \geq \vec{0},$$

где A – $(m \times n)$ — матрица. Он так же, как и симплекс метод заключается в таком последовательном переходе от одного n-мерного вектора $\vec{x}$ к другому, при котором после конечного числа переходов либо получается оптимальное решение задачи, либо устанавливается ее неразрешимость. Однако в отличие от симплекс-метода эти последовательно получаемые векторы, вообще говоря, не являются допустимыми решениями системы ограничений (допустимыми планами), а являются в некотором смысле «почти» допустимыми, так называемыми *псевдопланами* данной задачи. Эти псевдопланы являются базисными решениями системы уравнений задачи, но не обязательно удовлетворяют условиям неотрицательности. Процесс перехода от одного псевдоплана к другому построен так, что как только очередной псевдоплан $\vec{x}$ оказывается неотрицательным, т.е. допустимым планом задачи, он является одновременно и оптимальным решением этой задачи. Каждому псевдоплану по его определению соответствует опорное решение системы ограничений двойственной задачи. При этом каждому следующему певдоплану соответствует лучшее, чем предыдущее, опорное решение двойственной задачи. Оптимальному решению данной задачи соответствует оптимальное решение двойственной. Таким образом, метод последовательного уточнения оценок — это применение обычного симплекс метода к двойственной задаче, дополненное на каждой итерации построением n-мерного вектора $\vec{x}$, являющегося псевдопланом данной задачи. Геометрически этот метод можно истолковать как построение последовательности псевдопланов $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_s$, являющихся точками n-мерного пространства R^n. находящимися вне допустимого многогранника. При этом каждый следующий вектор этой последовательности определяет гиперплоскость уровня целевой функции, которая находится ближе к области допустимых значений, чем предыдущая. Через конечное число шагов мы получим точку $\vec{x}_s$, принадлежащую допустимому многограннику, либо выяснится, что допустимый многогранник пуст. Перейдем к более детальному описанию метода. Предположим, что система ограничений задачи в канонической форме приведена к базисному виду, который, вообще говоря, не является допустимым и пусть из целевой функции исключены базисные переменные. Составим таблицу, аналогичную симплекс таблице. Отличие заключается лишь в том, что эта таблица составляется по, вообще говоря, недопустимому базисному виду системы уравнений. Эту таблицу будем называть обобщенной симплекс таблицей. Коэффициенты $\Delta_j = -\gamma_j$ при свободных переменных в последней строке этой таблицы назовем оценками свободных переменных для данного базисного решения.

Определение 3.1. *Базисное решение системы ограничений задачи линейного программирования в канонической форме называется псевдопланом этой задачи, если для него все оценки свободных переменных неотрицательны* ($\Delta_j \geq 0$).

Утверждение 3.1. *Если псевдоплан задачи линейного программирования в канонической форме является допустимым планом этой задачи, то он одновременно является оптимальным планом.*

Доказательство. Если псевдоплан является допустимым, то составленная нами таблица будет обычной симплекс таблицей, у которой все коэффициенты $\Delta_j = -\gamma_j$ в последней строке неотрицательны. Значит, получена последняя симплекс таблица и соответствующий план будет оптимальным, что и требовалось доказать.

Утверждение 3.2. *Если псевдоплан задачи линейного программирования в канонической форме имеет отрицательную компоненту $b_k<0$, а все коэффициенты k-ой строки таблицы неотрицательны $a_{kj}\geq 0$, то задача не имеет решения по причине отсутствия допустимых решений системы ограничений.*

Доказательство. k-ая строка таблицы отвечает уравнению

$$x_k = b_k - \sum_j a_{kj} x_j \ ,$$

которое при наших условиях не может иметь неотрицательного решения. Следовательно, система ограничений задачи допустимых решений не имеет. Утверждение доказано.

Рассмотрим некоторую процедуру перехода от одной обобщенной симплекс таблицы к другой, которая соответствует переходу от одного псевдоплана к другому. Если таблица соответствует псевдоплану, который не является допустимым планом задачи, то в столбце свободных членов среди отрицательных элементов выберем максимальное по модулю отрицательное число $b_k<0$ и в k-ой строке отметим все отрицательные числа $a_{kj}<0$. Если таковых не окажется, то в силу утверждения 2 задача является недопустимой и решения не имеет. Среди отмеченных элементов k-ой строки выберем элемент a_{kj_0} , на котором достигается $\min\limits_j \left(-\dfrac{\Delta_j}{a_{kj}} \right)$. Если таких элементов несколько, то выбираем любой из них. Этот выбранный элемент берем в качестве разрешающего и производим операцию замещения в нашей таблице, выводя переменную x_k из числа базисных и вводя в число базисных переменную x_{j_0} . Отметим, что новая таблица тоже будет отвечать псевдоплану задачи вследствие правила выбора разрешающего элемента. При этом в новой таблице значение целевой функции γ_0 будет меньше, чем в старой. Таким образом, справедливо

Утверждение 3.3. *При описанном переходе от обобщенной симплекс таблицы, отвечающей некоторому псевдоплану, либо выяснится, что исходная задача не имеет решения, либо получится новая таблица, отвечающая некоторому новому псевдоплану. При этом значение целевой функции на новом псевдоплане будет меньше, чем на старом.*

Описанные итерации повторяются до тех пор, пока очередной псевдоплан не станет допустимым планом задачи, который одновременно будет оптимальным планом. Важно заметить, что каждому псевдоплну однозначно сопоставляется некоторый опорный план двойственной задачи. Действительно, если система ограничений приведена к базисному виду, а из целевой функции исключены базисные переменные, то базисные переменные можно убрать, превращая уравнения в неравенства. Для полученной формы задачи построим двойственную задачу. Приведение данной и двойственной задач к каноническому виду дает пару задач с одинаковым числом переменных, которые можно привести во взаимно однозначное соответствие вида (F). С использованием этого соответствия и обобщенной симплекс таблицы мы можем, как и в предыдущем параграфе каждому псевдоплану сопоставить допустимый план двойственной задачи. Наши итерации, по сути, являются переходами от одного опорного решения двойственной задачи к другому с улучшением значения целевой функции. Поскольку координаты опорного решения двойственной задачи совпадают с оценками свободных переменных псевдоплана, то мы, по сути, последовательно улучшаем эти оценки. Отсюда происходит название рассмотренного метода.

Утверждение 3.4. *Если исходная задача не имеет решения по причине неограниченности целевой функции на области допустимых значений, то она не имеет ни одного псевдоплана.*

Доказательство. Предположим противное, т. е. что существует хотя бы один псевдоплан. Ему должен соответствовать допустимый план двойственной задачи. Но по первой теореме двойственности при наших условиях двойственная задача должна быть недопустимой, т. е. ее область допустимых значений пуста. Пришли к противоречию. Утверждение 4 доказано.

Пример 3.6. Решить задачу

$$z = 16 - x_1 - x_2 \to \max,$$

$$\begin{cases} x_1 + x_2 \le 8, \\ x_1 - x_2 \ge 4, \\ x_1 + 2x_2 \ge 6, \end{cases}$$

$$x_1,\ x_2 \ge 0.$$

Построим двойственную задачу

$$f = 16 + 8y_1 - 4y_2 - 6y_3 \to \min,$$

$$\begin{cases} y_1 - y_2 - y_3 \ge -1, \\ y_1 + y_2 - 2y_3 \ge -1, \end{cases}$$

$$y_1, y_2, y_3 \ge 0.$$

Приведем исходную и двойственную задачи к каноническому виду.

$$z = 16 - x_1 - x_2 \to \max,$$
$$\begin{cases} x_1 + x_2 + x_3 = 8, \\ -x_1 + x_2 + x_4 = -4, \\ -x_1 - 2x_2 + x_5 = -6, \end{cases}$$
$$x_1,\ x_2,\ x_3,\ x_5 \ge 0.$$

$$f = 16 + 8y_1 - 4y_2 - 6y_3 \to \min,$$
$$\begin{cases} -y_1 + y_2 + y_3 + y_4 = 1, \\ -y_1 - y_2 + 2y_3 + y_5 = 1, \end{cases}$$
$$y_1, y_2, y_3 \ge 0.$$

Составим обобщенную симплекс таблицу исходной задачи.

Таблица 1

Базисные переменные	Свободные члены	x_1	$\downarrow x_2$	x_3	x_4	x_5
x_3	8	1	1	1	0	0
x_4	–4	–1	1	0	1	0
$\leftarrow x_5$	–6	–1	[–2]	0	0	1
z	16	1	1	0	0	0

Таблица отвечает псевдоплану. Согласно соответствию переменных (F) этому псевдоплану соответствует допустимый план (0, 0, 0, 1, 1) двойственной задачи. По сформулированному выше правилу выбираем разрешающий элемент (–2) и перейдем к следующей таблице.

Таблица 2

Базисные переменные	Свободные члены	$\downarrow x_1$	x_2	x_3	x_4	x_5
x_3	5	1/2	0	1	0	1/2
$\leftarrow x_4$	–7	[–3/2]	0	0	1	1/2
x_2	3	1/2	1	0	0	–1/2
z	13	1/2	0	0	0	1/2

Эта таблица тоже соответствует псевдоплану, а также допустимому плану (0, 0, ½, ½, 0) двойственной задачи.

Базисные переменные	Свободные члены	x_1	x_2	x_3	x_4	x_5
x_3	8/3	0	0	1	1/3	2/3
x_1	14/3	1	0	0	–2/3	–1/3
x_2	2/3	0	1	0	1/3	–1/3
z	32/3	0	0	0	1/3	2/3

В последней симплекс таблице псевдоплан является допустимым планом, а, следовательно, оптимальным планом задачи. Получаем ответ в исходной задаче: z_{max}=32/3; точка максимума (14/3; 2/3). Решение двойственной задачи тоже прочитывается по последней симплекс таблице: f_{min}=32/3; точка минимума (0; 1/3; 2/3).

Отметим, что для получения решения исходной задачи методом последовательного уточнения оценок строить двойственную задачу, как в рассмотренном примере, совсем не обязательно. Рассмотренный пример показывает, по сути, равносильность нашего метода и двойственного симплекс метода, рассмотренного в предыдущем параграфе. Поэтому в дальнейшем метод последовательного уточнения оценок мы часто будем называть тоже *двойственным симплекс методом*. Этот метод удобен в случае, когда удалось выйти на псевдоплан. Такая ситуация встречается довольно часто (см., например, примеры 5.3 и 5.4). В общем же его применение может привести к сокращению количества итераций, необходимых в среднем для решения произвольной задачи линейного программирования в канонической форме по сравнению с применением простого симплекс метода или его модификаций типа метода искусственного базиса. Следующий составной алгоритм часто называют *обобщенным двойственным симплекс методом*.

1. Привести систему ограничений задачи к базисному виду и исключить базисные переменные из целевой функции.
2. Проверить, будет ли полученное базисное решение допустимым планом задачи. Если да, то решить задачу обычным симплекс методом и перейти к п. 5, иначе перейти к п. 3
3. Проверить, будет ли полученное базисное решение псевдопланом задачи. Если да, то решить задачу двойственным симплекс методом и перейти к п. 5, иначе перейти к п. 4.
4. Перейти к следующему базисному виду системы ограничений и перейти к п. 2.
5. Конец.

Контрольные вопросы и задачи для самостоятельного решения

1. Сформулируйте правило составления задачи, двойственной по отношению к данной задаче линейного программирования в стандартной форме. Какие пары задач называют симметричными взаимно двойственными?

2. В чем заключается экономический смысл пары симметрично двойственных задач?

3. Несимметрично двойственные задачи. В чем состоит общее правило построения двойственных задач?

4. Сформулируйте первую теорему двойственности. Что позволяет сказать эта теорема о задаче линейного программирования, если известно решение двойственной задачи?

5. Сформулируйте вторую теорему двойственности. Какие задачи позволяет решать эта теорема?

6. Третья теорема двойственности. Область устойчивости двойственных оценок и ее отыскание с помощью второй теоремы двойственности?

7. Что такое послеоптимизационный анализ?

8. В чем заключается двойственный симплекс-метод для пары симметрично двойственных задач.

9. Что называется псевдопланом задачи линейного программирования в канонической форме?

10. Опишите алгоритм последовательного уточнения оценок.

Сформулируйте двойственные задачи по отношению к задачам 3.1 — 3.4.

3.1.

$$z = 3x_1 + 3x_2 - 4x_3 \to \max;$$

$$\begin{cases} 2x_1 + x_2 - 3x_3 \ge 18, \\ 4x_1 - 5x_3 \le 12, \\ 3x_1 - 2x_2 + x_3 \ge 14, \end{cases}$$

$$x_1, x_2, x_3 \ge 0.$$

3.2.

$$z = 6x_1 - x_2 + 3x_3 \to \min;$$

$$\begin{cases} 4x_1 - 7x_2 + 5x_3 \le 15, \\ 2x_1 + 3x_2 - 4x_3 = 16, \\ 6x_1 + 5x_2 - 8x_3 \ge 12, \end{cases}$$

$$x_1, x_2, x_3 \ge 0.$$

3.3.

$$z = -2x_1 + 5x_2 - 4x_3 \to \max,$$

$$\begin{cases} 4x_1 + 2x_2 - 3x_3 \ge 9, \\ 3x_1 - 2x_2 + 5x_3 \le 8, \\ x_1 + 3x_2 + 4x_3 \ge 12, \end{cases}$$

$$x_1, x_2 \ge 0.$$

3.4.

$$z = -3x_1 + 4x_2 - 6x_3 \to \min;$$

$$\begin{cases} 2x_1 + 3x_2 - x_3 \ge 8, \\ -3x_1 + 2x_2 - 2x_3 = 10, \\ 5x_1 - 4x_2 + x_3 \ge 7, \end{cases}$$

$$x_1, x_2, x_3 \ge 0.$$

Сформулируйте двойственные задачи по отношению к задачам 3.5, 3.6 *и найдите их решения графически.*

3.5.

$z = 6x_1 + 8x_2 + x_3 \to \max;$

$$\begin{cases} x_1 + x_2 + x_3 \le 3, \\ x_1 + 2x_2 \le 4, \end{cases}$$

$x_1, x_2, x_3 \ge 0.$

Отв. $f_{\min} = 20$ *при* $y_1^0 = 4,\ y_2^0 = 2.$

3.6.

$z = 27x_1 + 10x_2 + 15x_3 + 28x_4 \to \max;$

$$\begin{cases} 3x_1 + 2x_2 + x_3 + 2x_4 \le 2, \\ 3x_1 + x_2 + 3x_3 + 4x_4 \le 5, \end{cases}$$

$x_1, \ldots, x_4 \ge 0.$

Отв. $f_{\min} = 29$ *при* $y_1^0 = 12,\ y_2^0 = 1.$

3.7, 3.8. *В задачах* 3.5, 3.6 *найти решение исходной задачи, используя решение двойственной и вторую теорему двойственности.*

3.9. *Для модели задачи* 1.2 *составить двойственную модель. Кроме того: а*) *найти оптимальный план двойственной задачи; б*) *определить область устойчивости двойственных оценок по отношению к изменениям количества сырья каждого типа; в*) *определить увеличение максимальной стоимости изготавливаемой продукции при увеличении количества сырья каждого типа соответственно на* 30, 40 *и* 50 *т.*

Найдите решения задач 3.10 —3.15, *используя двойственный симплекс-метод.*

3.10

$z = -4x_1 - 7x_2 - 8x_3 - 5x_4 \to \max;$

$$\begin{cases} x_1 + x_2 + 2x_4 \ge 4, \\ 2x_1 + x_2 + 2x_3 \ge 6, \end{cases}$$

$x_1, \ldots, x_4 \ge 0.$

Ответ: $z_{\max} = -29/2$, точка max (3; 0;0;1/2).

3.11.

$z = 5x_1 + 6x_2 + x_3 + x_4 \to \min;$

$$\begin{cases} 1{,}5x_1 + 3x_2 - x_3 + x_4 \ge 18, \\ 3x_1 + 2x_3 - 4x_4 \ge 24, \end{cases}$$

$x_1, \ldots, x_4 \ge 0.$

Ответ: $z_{\min}=52$, точка min (8;2;0;0).

3.12.

$z = x_1 + 3x_2 + 4x_3 + 2x_4 \to \min;$

$$\begin{cases} x_1 - x_2 + 4x_3 + 5x_4 \geq 27, \\ 2x_1 + 3x_2 - x_3 + 4x_4 \geq 24, \end{cases}$$

$x_1, \ldots, x_4 \geq 0.$

Ответ: $z_{\min}$=12, точка min (2;0;0;5).

3.13.

$z = -2x_1 - 8x_2 - x_3 - 5x_4 \to \max;$

$$\begin{cases} -2x_1 + x_2 - 3x_3 + x_4 \geq 18, \\ x_1 + 2x_2 + 4x_3 + 2x_4 \geq 24, \\ 3x_1 + 4x_2 + 2x_3 - 3x_4 \geq 30, \end{cases}$$

$x_1, \ldots, x_4 \geq 0.$

Ответ: $z_{\max}$=126, точка max (0; 12;0;6).

3.14.

$z = -x_1 - 7x_2 + 4x_3 - 9x_4 - 8x_5 + 3x_6 \to \max;$

$$\begin{cases} 3x_1 + 2x_2 + 3x_3 - 2x_4 + x_5 + x_6 = 18, \\ 2x_1 + x_2 - x_3 - 3x_4 + 2x_5 \geq 24, \end{cases}$$

$x_i \geq 0 \ (i = \overline{1,6}).$

Ответ: $z_{\max}$=–75, точка max (3; 0; 0; 0;9;0).

3.15.

$z = 2x_1 + 3x_2 + 5x_4 \to \max;$

$$\begin{cases} -2x_1 + x_2 - x_3 = 12, \\ x_1 + 2x_2 + x_4 = 10, \\ 3x_1 - 2x_2 - x_5 = 18, \end{cases}$$

$x_i \geq 0, \quad i = \overline{1,5}.$

Ответ: Задача не имеет решения.

4. ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОЙ ТЕОРИИ ИГР

Во многих видах человеческой деятельности, а особенно в экономике, часто встречаются ситуации, в которых интересы различных лиц, организаций и т.д. противоречат друг другу. Такие ситуации называют конфликтными. Например, при определении цен на товары интересы покупателя и продавца противоположны друг другу. Реальные конфликтные ситуации сложны для анализа, поэтому их нужно предварительно формализовать, отбросив все несущественные детали. Хорошей формой конфликтной ситуации, которая легко поддается формализации, является игра. Разрешение многих конфликтных ситуаций можно представить себе в виде игр, снабженных теми или иными правилами. Поэтому математическое моделирование игр является важной задачей.

Игры можно классифицировать по различным признакам, например, по количеству участников, по правилам распределения выигрышей и т.д. Мы рассмотрим простейшие игры с двумя участниками, в которых выигрыш одного из игроков является проигрышем другого. Такие игры называются играми двух игроков с нулевой суммой.

Во всех играх присутствует понятие стратегии игры, т.е. описание поведения игрока в зависимости от сложившейся ситуации. Игроки могут применять различные стратегии с тем или иным успехом.

4.1. Матричная игра двух игроков с нулевой суммой

Мы рассмотрим следующую формализацию игры, которая состоит из двух наборов стратегий первого и второго игроков. При этом конкретное содержание стратегий нас интересовать не будет. Обозначим возможные стратегии первого игрока через $A_1, A_2, \ldots, A_n$, а стратегии второго — $B_1, B_2, ..., B_m$. Игра является одноходовой: первый игрок применяет одну из своих возможных стратегий, а второй отвечает стратегией из своего набора. После этого происходит распределение выигрышей, которое задается числами a_{ij} — выигрышем первого игрока при условии, что он применяет стратегию A_i, а второй игрок отвечает стратегией B_j. При этом выигрыш первого игрока является проигрышем второго. Числа a_{ij} образуют матрицу, которую называют матрицей выигрышей первого игрока или платежной матрицей. Величины a_{ij} могут быть как положительными, так и отрицательными или равными нулю. Если $a_{ij} < 0$, то первый игрок проигрывает, а второй выигрывает сумму, равную $|a_{ij}|$.

Пример 4.1. Игра заключается в том, что первый игрок накрывает монету гербом или решкой вверх, а второй отгадывает. Если второй игрок угадывает, то получает выигрыш, равный 1. Если второй игрок не угадывает, то он проигрывает 1. Возможные стратегии первого игрока: A_1 — накрыть монету гербом вверх, A_2 — накрыть монету решкой вверх. У второго игрока тоже две стратегии: B_1 — назвать орла, B_2 — назвать решку. Платежная матрица имеет следующий вид:

	B_1	B_2
A_1	–1	1
A_2	1	–1

4.2. Анализ игры в чистых стратегиях

Основной задачей теории игр является выбор для игроков оптимальной в том или ином смысле стратегии. Платежную матрицу анализируют, стремясь найти такую стратегию, которая при любом поведении противника гарантировала бы достаточно высокий выигрыш. При этом считается, что противник ведет себя наиболее неблагоприятным образом.

Пример 4.2. Пусть матричная игра задается матрицей:

	B_1	B_2	B_3	B_4	α_i
A_1	10	7	4	1	$\boxed{1}$
A_2	5	–1	6	8	–1
A_3	2	–8	5	3	–8
β_j	10	7	$\boxed{6}$	8	1 6

При наиболее неблагоприятном поведении второго игрока стратегия A_1 первого игрока доставляет ему минимальный гарантированный выигрыш 1. Стратегии A_2 и A_3 в этом смысле хуже, чем A_1, так как при неблагоприятных условиях ведут к проигрышу. С точки зрения второго игрока наилучшей является стратегия B_3, так как ее применение может привести к проигрышу 6, а при остальных стратегиях можно проиграть больше. В этом смысле A_1 наилучшая стратегия первого, а B_3 — второго игрока.

В общем случае игроки оценивают свои стратегии следующим образом. Первый игрок, рассматривая свои стратегии, ищет $\min\limits_{j} a_{ij}$, а затем выбирает такую стратегию A_i, при которой эта величина наибольшая. При этом он вычисляет величину

$$\alpha = \max_{i} \min_{j} a_{i\,j}$$

Величина α называется *нижней ценой игры*, а правило выбора наилучшей стратегии называется *правилом максимина*. В нашем примере α=1. Второй игрок находит $\max\limits_{i} a_{ij}$, затем выбирает стратегию, при которой эта величина наименьшая, то есть он вычисляет величину $\beta = \min\limits_{i} \max\limits_{j} a_{i\,j}$. β называется *верхней ценой игры*, а правило, которым пользуется второй игрок для выбора своей наилучшей стратегии, — *правилом минимакса*. В нашем примере β = 6. Можно доказать, что для любой платежной матрицы $\alpha \le \beta$.

В рассмотренном примере большую роль играет информация о ходе противника. Если второй игрок знает, что первый применит стратегию A_1, он может, выбирая стратегию B_4 уменьшить свой проигрыш по сравнению со стратегией B_3. Если первый знает, что второй применит стратегию B_3, он имеет возможность увеличить свой выигрыш, применяя стратегию A_2. То есть приведенная выше оценка качества стратегий теряет смысл при наличии информации о поведении противника. Такая ситуация характерна для большинства игр. Информация о поведении противника — залог успеха в игре. Однако для некоторых платежных матриц это не так.

Пример 4.3. Пусть игра задаётся матрицей

	B_1	B_2	B_3	B_4	α_i
A_1	2	4	7	5	2
A_2	7	6	8	7	$\boxed{6}$
A_3	5	3	4	1	1
β_j	7	$\boxed{6}$	8	7	

В этом примере верхняя и нижняя цены игры совпадают $\alpha = \beta = 6$. Если первый игрок придерживается стратегии A_2, второму ничего не остается, как выбрать стратегию B_2, иначе он проиграет больше. Если второй игрок придерживается стратегии B_2, то первому нужно придерживаться стратегии A_2, иначе он выиграет меньше. Пара стратегий A_2, B_2 является парой оптимальных стратегий для обоих игроков, и информация о поведении противника здесь роли не играет.

В случае, когда верхняя и нижняя цены игры совпадают, говорят, что игра имеет седловую точку в *чистых стратегиях*. В этом случае всегда найдется такой элемент $a_{i_0 j_0}$ платежной матрицы, который удовлетворяет неравенствам

$$a_{i\,j_0} \le a_{i_0 j_0} \le a_{i_0 j}$$

при любых i, j. Этот элемент $a_{i_0 j_0}$ называется *седловой точкой игры*, а стратегии A_{i_0}, B_{j_0} называют *оптимальными стратегиями*, отвечающими седловой точке. Слова "чистая стратегия" означают просто стратегии первого и второго игроков. Существование седловой точки является исключением. Большинство игр не имеет седловой точки в чистых стратегиях. Если же игра имеет седловую точку, то она практически теряет смысл, так как ходы противников предрешены. Часто говорят, что в этом случае игра решается в чистых стратегиях, а пару оптимальных стратегий называют решением игры в чистых стратегиях.

4.3. Понятие смешанной стратегии. Седловая точка игры в смешанных стратегиях

Предположим, что игра повторяется много раз, то есть, проводится достаточно длинная серия партий, и каждый из игроков выбирает в каждой партии свою чистую стратегию случайным образом, в соответствии с некоторыми вероятностями выбора стратегий.

Определение 4.1. *Смешанной стратегией игрока называется совокупность вероятностей выбора им своих чистых стратегий.*

Обозначим через $p_1, p_2, \dots, p_n$ вероятности выбора чистых стратегий первым игроком в серии партий. Эта совокупность чисел и составляет смешанную стратегию первого игрока. Смешанная стратегия — это n-мерный вектор, компоненты которого удовлетворяют условиям

$$p_i \ge 0 \;\; (i = 1, 2, \dots, n); \;\; p_1 + p_2 + \dots + p_n = 1.$$

Под смешанной стратегией второго игрока будем понимать любой m-мерный вектор $\{q_1, q_2, \dots, q_m\}$ удовлетворяющий условиям

$$q_i \ge 0 \; (i = 1, 2, \dots, m); \;\; q_1 + q_2 + \dots + q_m = 1.$$

Определение 4.2. *Средним выигрышем первого игрока или платежной функцией игры называется функция* $n+m$ *переменных*

$M(\vec{p},\vec{q})=\sum\limits_{i,j} a_{ij}\, p_i q_j$, *определенная на смешанных стратегиях первого и второго игроков.*

При конкретном выборе смешанных стратегий $M(\vec{p},\vec{q})$ является математическим ожиданием выигрыша первого игрока.

Определение 4.3. *Седловой точкой игры в смешанных стратегиях называется такая пара смешанных стратегий первого и второго игроков*

$$\vec{p}^{\,0}=\left\{p_1^0,\, p_2^0,\dots,\, p_n^0\right\} \quad \text{и} \quad \vec{q}^{\,0}=\left\{q_1^0,\, q_2^0,\dots,\, q_m^0\right\},$$

для которой при любых смешанных стратегиях $\vec{p},\ \vec{q}$ *выполняются неравенства*

$$M\left(\vec{p},\vec{q}^{\,0}\right)\le M\left(\vec{p}^{\,0},\vec{q}^{\,0}\right)\le M\left(\vec{p}^{\,0},\vec{q}\right).$$

Седловая точка представляет собой наиболее выгодную пару смешанных стратегий для каждого игрока, так как отступление от нее невыгодно им обоим.

Отметим, что чистые стратегии можно рассматривать как частный случай смешанных. Например, стратегию A_1 первого игрока можно рассматривать как смешанную стратегию $\{1, 0, 0, \dots ,0\}$.

Пример 4.4. Составим платежную функцию игры в так называемое тюремное очко. Каждый из игроков выбрасывает определенное число пальцев (1, 2 или 3). Если сумма выброшенных пальцев четная, то выигрывает первый игрок и его выигрыш равен сумме выброшенных пальцев. Если сумма нечетная, выигрывает второй игрок и его выигрыш равен выброшенной сумме. Здесь каждый из игроков имеет по три стратегии и легко составить платежную матрицу

	B_1	B_2	B_3
A_1	2	–3	4
A_2	–3	4	–5
A_3	4	–5	6

Если игра повторяется много раз и чистые стратегии выбираются в соответствии с вероятностями $\{p_1, p_2, p_3\}$ для первого игрока и $\{q_1, q_2, q_3\}$ для второго, то платежная функция этой игры имеет вид

$$M(\vec{p},\vec{q}) = 2p_1q_1 - 3p_1q_2 + 4p_1q_3 - 3p_2q_1 + 4p_2q_2 -$$
$$- 5p_2q_3 + 4p_3q_1 - 5p_3q_2 + 6p_3q_3.$$

Посчитаем значение платежной функции при следующих смешанных стратегиях $\vec{p}^{\,0} = \left\{\frac{1}{4};\frac{1}{2};\frac{1}{4}\right\}$, $\vec{q}^{\,0} = \left\{\frac{1}{4};\frac{1}{2};\frac{1}{4}\right\}$. $M\left(\vec{p}^{\,0},\vec{q}^{\,0}\right) = 0.$

Можно показать, что пара стратегий $\vec{p}^{\,0}, \vec{q}^{\,0}$ в этом примере является седловой точкой.

4.4. Теорема Фон-Неймана. Нахождение седловой точки игры в смешанных стратегиях

Следующая теорема доказана американским математиком Фон-Нейманом (см., например, [24]).

Теорема 4.1. *Для каждой матричной игры двух игроков с нулевой суммой существует пара смешанных стратегий игроков, которая является седловой точкой игры в смешанных стратегиях.*

Теорема утверждает существование седловой точки, но вовсе не ее единственность. В некоторых случаях игра может иметь бесчисленное множество седловых точек в смешанных стратегиях.

Нахождение седловой точки в смешанных стратегиях называют *решением игры* в смешанных стратегиях. Решение игры сводится к решению пары двойственных задач линейного программирования. Сформулируем эту пару задач.

Для первого игрока смешанная стратегия в том и только в том случае отвечает седловой точке в смешанных стратегиях, когда она является точкой максимума задачи

$$A)\quad z = v \to \max;$$
$$\begin{cases} a_{11}p_1 + a_{21}p_2 + \dots + a_{n1}p_n \ge v, \\ a_{12}p_1 + a_{22}p_2 + \dots + a_{n2}p_n \ge v, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{1m}p_1 + a_{2m}p_2 + \dots + a_{nm}p_n \ge v; \\ p_1 + p_2 + \dots + p_n = 1; \end{cases}$$
$$p_i \ge 0 \quad \left(i = \overline{1,n}\right).$$

Здесь v — независимая переменная, которая, вообще говоря, не подчиняется условию неотрицательности.

Для второго игрока смешанная стратегия $q_1^0, q_2^0, \dots, q_m^0$ *в том и только в том случае отвечает седловой точке, когда она является точкой минимума задачи*

$B)\quad f = u \to \min;$

$$\begin{cases} a_{11}q_1 + a_{12}q_2 + \dots + a_{1m}q_m \le u, \\ a_{21}q_1 + a_{22}q_2 + \dots + a_{2m}q_m \le u, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots & q_i \ge 0\ \left(i = \overline{1, m}\right). \\ a_{n1}q_1 + a_{n2}q_2 + \dots + a_{nm}q_m \le u, \\ q_1 + q_2 + \dots + q_m = 1; \end{cases}$$

Здесь переменная u не подчинена условию неотрицательности.

Форму этих задач линейного программирования нетрудно объяснить. Пусть v — оптимальный выигрыш первого игрока. Оценивая свою смешанную стратегию $\{p_1, p_2, \dots, p_n\}$, первый игрок стремится, чтобы при каждой чистой стратегии второго игрока его средний выигрыш был не меньше оптимального. Если второй игрок применяет свою стратегию B_j, то первый должен выбирать смешанную стратегию, удовлетворяющей неравенству

$$a_{1j}p_1 + a_{2j}p_2 + \ldots + a_{nj}p_n \ge v, \quad j = 1,\ 2,\ \ldots,\ m.$$

Аналогично рассуждает и второй игрок. Если u — проигрыш второго игрока, который он стремится минимизировать, то оценивая свою смешанную стратегию $\{q_1, q_2, \dots, q_{m,}\}$ второй игрок должен стремиться к выполнению неравенств

$$a_{i1}q_1 + a_{i2}q_2 + \ldots + a_{im}q_m \le u, \quad i = 1, 2, \ldots, n\ .$$

Задачи А) и В) двойственны друг другу. Решение их в общем случае громоздко. Но для игр размера 2 × m и n × 2 одну из задач можно решить графически, а решение другой найти с помощью условий дополнительной нежесткости. Кроме того, в общем случае можно так преобразовать задачи А) и В), что они могут быть решены двойственным симплекс-методом в форме, которая рассматривалась выше.

4.5. Графическое решение игр размера $2\times m$ и $n\times 2$

Рассмотрим игру размера $2\times m$. Для оптимальной стратегии первого игрока имеем задачу

$$z = v \to \max;$$

$$\begin{cases} a_{11}p_1 + a_{21}p_2 \ge v, \\ a_{12}p_1 + a_{22}p_2 \ge v, \\ \dots\dots\dots\dots\dots & . \\ a_{1m}p_1 + a_{2m}p_2 \ge v, \\ p_1 + p_2 = 1;\ p_1, p_2 \ge 0 \end{cases}$$

Поскольку $p_2 = 1 - p_1$, то, подставляя это выражение вместо p_2 в остальные ограничения, получим задачу с двумя переменными p_1 и ν

$$z = v \to \max;$$

$$\begin{cases} (a_{11} - a_{21})\, p_1 + a_{21} - v \ge 0, \\ (a_{12} - a_{22})\, p_1 + a_{22} - v \ge 0, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ (a_{1m} - a_{2m})\, p_1 + a_{2m} - v \ge 0; \\ 0 \le p_1 \le 1. \end{cases}$$

Эту задачу можно решать графически. Для задач размером $n \times 2$ точно также преобразуется задача для второго игрока

$$f = u \to \min;$$

$$\begin{cases} a_{11}q_1 + a_{12}q_2 \le u, \\ a_{21}q_1 + a_{22}q_2 \le u, \\ \cdots\cdots\cdots\cdots\cdots \\ a_{n1}q_1 + a_{n2}q_2 \le u, \\ q_1 + q_2 = 1; \;\; q_1, q_2 \ge 0. \end{cases}$$

Пример 4.5. Решить игру с платежной матрицей

	$\boldsymbol{B}_1$	$\boldsymbol{B}_2$	$\boldsymbol{B}_3$	$\boldsymbol{B}_4$
$\boldsymbol{A}_1$	2	1	5	3
$\boldsymbol{A}_2$	1	3	4	0,5

Соответствующая задача линейного программирования имеет вид:

$$z = v \to \max;$$

$$\begin{array}{l} q_1 \\ q_2 \\ q_3 \\ q_4 \\ u \end{array} \begin{cases} 2p_1 + p_2 \ge v, \\ p_1 + 3p_2 \ge v, \\ 5p_1 + 4p_2 \ge v \\ 3p_1 + 0{,}5p_2 \ge v, \\ p_1 + p_2 = 1; \end{cases}$$

$$p_1, p_2 \ge 0.$$

Исключим переменную p_2; $p_2 = 1 - p_1$:

$$z = v \to \max;$$

$$\begin{cases} p_1 - v \ge -1, \\ -2p_1 - v \ge -3, \\ p_1 - v \ge -4 \\ 2{,}5p_1 - v \ge -0{,}5; \end{cases}$$

$$0 \le p_1 \le 1.$$

Полученную задачу решаем графически (рис. 4.1).

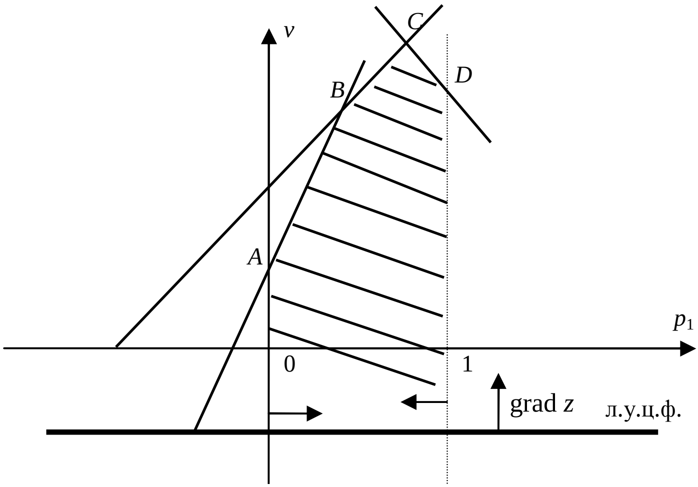


Рис. 4.1. Графическое решение задачи примера 4.5

Точкой максимума является точка C, удовлетворяющая системе уравнений

$$\begin{cases} p_1 - v = -1, \\ 2p_1 + v = 3. \end{cases}$$

Отсюда $p_1^0 = \dfrac{2}{3}$; $p_2^0 = \dfrac{1}{3}$ — смешанная стратегия первого игрока в седловой точке. Для нахождения смешанной стратегии второго игрока применим условия дополнительной нежесткости. Двойственная задача имеет вид

$$f = u \to \min;$$

$$\begin{array}{l} p_1 \\ p_2 \\ v \end{array} \begin{cases} 2q_1 + q_2 + 5q_3 + 3q_4 \le u, \\ q_1 + 3q_2 + 4q_3 + 0{,}5q_4 \le u, \\ q_1 + q_2 + q_3 + q_4 = 1; \end{cases}$$

$$q_1, q_2, q_3, q_4 \ge 0.$$

Условия дополнительной нежесткости запишутся так:

$$\begin{cases} p_1^0(2q_1^0 + q_2^0 + 5q_3^0 + 3q_4^0 - u^0) = 0, \\ p_2^0(q_1^0 + 3q_2^0 + 4q_3^0 + 0{,}5q_4^0 - u^0) = 0, \\ v^0(q_1^0 + q_2^0 + q_3^0 + q_4^0 - 1) = 0, \\ q_1^0(2p_1^0 + p_2^0 - v^0) = 0, \\ q_2^0(p_1^0 + 3p_2^0 - v^0) = 0, \\ q_3^0(5p_1^0 + 4p_2^0 - v^0) = 0, \\ q_4^0(3p_1^0 + 0{,}5p_2^0 - v^0) = 0, \\ u^0\left(p_1^0 + p_2^0 - 1\right) = 0. \end{cases}$$

Подставим найденные $p_1^0 = \frac{2}{3}$ и $p_2^0 = \frac{1}{3}$ в эти равенства, учитывая, что $v^0 = u^0 = \frac{5}{3}$. Четвертое, пятое и восьмое равенства будут тождествами. Из шестого и седьмого следует, что $q_3^0 = q_4^0 = 0$. Для определения q_1^0 и q_2^0 имеем

$$\begin{cases} 2q_1^0 + q_2^0 = \frac{5}{3}, \\ q_1^0 + 3q_2^0 = \frac{5}{3}. \end{cases}$$

Откуда следует $q_1^0 = \frac{2}{3}$; $q_2^0 = \frac{1}{3}$. Таким образом, седловая точка игры

$$\vec{p}^{\,0} = \left\{\frac{2}{3}; \frac{1}{3}\right\}; \ \vec{q}^{\,0} = \left\{\frac{2}{3}; \frac{1}{3}; 0; 0\right\}.$$

Отметим, что средний выигрыш в седловой точке называется обычно *ценой игры* в смешанных стратегиях. В нашей паре двойственных задач для первого и второго игроков цена игры равна $v^0 = u^0$. В примере 4.5 цена игры равна $5/3$.

4.6. Решение игры двойственным симплекс-методом

Преобразуем пару двойственных задач *A*) и *B*) так, чтобы удобно было применять двойственный симплекс-метод. Прежде всего, отметим, что седловая точка игры не изменится, если ко всем элементам платежной матрицы прибавить одну и ту же константу. Действительно, если ко всем элементам прибавить константу *C*, то мы получим новую платежную функцию

$$M_1(\vec{p},\vec{q}) = \sum_{i,j}(a_{ij}+C)p_i q_j =$$

$$\sum_{i,j} a_{ij} p_i q_j + C\left(\sum_{i,j} p_i q_j\right) = M(\vec{p},\vec{q}) + C.$$

Таким образом, новая платежная функция отличается от старой постоянным слагаемым *C*. Если $\{\vec{p}^0,\vec{q}^0\}$ — седловая точка старой платежной функции, то

$$M\left(\vec{p},\vec{q}^0\right) \le M\left(\vec{p}^0,\vec{q}^0\right) \le M\left(\vec{p}^0,\vec{q}\right).$$

Прибавляя к обеим частям каждого этих неравенств константу *C*, получим

$$M_1\left(\vec{p},\vec{q}^0\right) \le M_1\left(\vec{p}^0,\vec{q}^0\right) \le M_1\left(\vec{p}^0,\vec{q}\right).$$

То есть $\{\vec{p}^0,\vec{q}^0\}$ является седловой точкой и новой платежной функции. Прибавляя константу *C* ко всем элементам матрицы мы всегда можем добиться того, чтобы цена игры была положительна, а седловая точка осталась бы прежней. При этом, поскольку в решениях задач *A*) и *B*) u^0 и v^0 положительны, можно считать переменные подчиненными дополнительным условиям $v > 0$; $u > 0$, которые на точки оптимума задач не повлияют.

Рассмотрим ограничение $p_1 + p_2 + \cdots + p_n = 1$ и разделим обе его части на v, обозначив $\frac{p_i}{v} = x_i \; (i = 1, 2, \ldots, n)$. Поскольку в нашей задаче $v \to \max$, мы получим задачу на минимум для новой целевой функции

$$z_1 = \frac{1}{v} = x_1 + x_2 + \cdots + x_n \to \min.$$

Разделив все неравенства системы ограничений задачи *A*) на *v*, получим для переменных x_i условия

$$\begin{cases} a_{11}x_1 + a_{21}x_2 + \cdots + a_{n1}x_n \geq 1, \\ a_{12}x_1 + a_{22}x_2 + \cdots + a_{n2}x_n \geq 1, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{1m}x_1 + a_{2m}x_2 + \cdots + a_{nm}x_n \geq 1; \end{cases}$$

$$x_i \geq 0, \; i = \overline{1, n}.$$

Производя аналогичные преобразования задачи *B*) и вводя новые переменные $y_j = \dfrac{q_j}{u}$, получаем новую задачу:

$$f_1 = \frac{1}{u} = y_1 + y_2 + \cdots + y_m \to \max;$$

$$\begin{cases} a_{11}y_1 + a_{12}y_2 + \cdots + a_{1m}y_m \leq 1, \\ a_{21}y_1 + a_{22}y_2 + \cdots + a_{2m}y_m \leq 1, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1}y_1 + a_{n2}y_2 + \cdots + a_{nm}y_m \leq 1; \end{cases}$$

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

Мы получили пару симметрично-двойственных задач. Последнюю задачу можно решать симплекс-методом непосредственно после уравнивания неравенств, а решение другой можно будет прочесть по последней симплекс-таблице. Если $y_1^0, y_2^0, \ldots, y_m^0$ — точка максимума задачи для второго игрока, то вероятности чистых стратегий в седловой точке можно найти по формулам

$$q_1^0 = \frac{y_1^0}{f_{1\max}}; \; q_2^0 = \frac{y_2^0}{f_{1\max}}; \cdots; \; q_m^0 = \frac{y_m^0}{f_{1\max}}.$$

Если $x_1^0, x_2^0, \ldots, x_n^0$ — точка минимума задачи для первого игрока, то вероятности стратегий в седловой точке равны

$$p_1^0 = \frac{x_1^0}{z_{1\min}}; \; p_2^0 = \frac{x_2^0}{z_{1\min}}; \cdots; \; p_m^0 = \frac{x_m^0}{z_{1\min}}.$$

Отметим, что $f_{\min} = z_{\max} = A.$ Цена игры при этом равна $A - C$, где C — константа, которую мы прибавляли, гарантируя неравенства $v > 0$; $u > 0$.

Пример 4.6. Наметим нахождение седловой точки в смешанных стратегиях для игры примера 4.4. Прибавим ко всем элементам матрицы число 6. Получим матрицу

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \begin{pmatrix} 8 & 3 & 10 \\ 3 & 10 & 1 \\ 10 & 1 & 12 \end{pmatrix} \end{array}.$$

Построим пару двойственных задач в переменных x_1, x_2, x_3 и y_1, y_2, y_3.

1) $z_1 = x_1 + x_2 + x_3 \to \min;$

$$\begin{cases} 8x_1 + 3x_2 + 10x_3 \ge 1, \\ 3x_1 + 10x_2 + x_3 \ge 1, \\ 10x_1 + x_2 + 12x_3 \ge 1; \end{cases}$$

$x_1, x_2, x_3 \ge 0.$

2) $f_1 = y_1 + y_2 + y_3 \to \max;$

$$\begin{cases} 8y_1 + 3y_2 + 10y_3 \le 1, \\ 3y_1 + 10y_2 + y_3 \le 1, \\ 10y_1 + y_2 + 12y_3 \le 1; \end{cases}$$

$y_1, y_2, y_3, \ge 0.$

Вторую задачу можно решить симплекс-методом непосредственно после уравнивания неравенств системы ограничений.

4.7. Доминирование и дублирование стратегий. Упрощение игры

Если платежная матрица такова, что каждый элемент некоторой строки с номером i не меньше соответствующего элемента строки с номером k и, по меньшей мере, один ее элемент строго больше соответствующего элемента строки с номером k, то говорят, что стратегия A_i первого игрока доминирует над его стратегией A_k. Очевидно, что стратегия первого игрока, для которой есть доминирующая, не может быть оптимальной чистой стратегией для него, или входить в его оптимальную смешанную стратегию с ненулевой вероятностью. Таким образом, такую стратегию можно исключить из рассмотрения, вычеркнув из матрицы строку с номером k. Аналогично, если каждый элемент столбца с номером j платежной матрицы не больше соответствующего элемента столбца с номером r и, по меньшей мере, один его элемент строго меньше соответствующего элемента столбца с номером r, то говорят, что стратегия B_j второго игрока доминирует над его стратегией B_r. При поиске оптимальных стратегий столбец с номером r можно вычеркнуть.

Вообще говоря, в платежной матрице могут быть одинаковые строки или столбцы. Соответствующие стратегии называются дублирующими. Очевидно, что дублирующие стратегии целесообразно отбросить.

Выявление доминирующих и дублирующих стратегий позволяет произвести упрощение игры, то есть сократить размеры ее платежной матрицы, исходя из того очевидного факта, что стратегии, над которыми есть доминирующие, применяться в игре не будут, а избавиться от дублирования стратегий можно в виду их полной взаимозаменяемости. Упрощение игры обычно производят в следующем порядке. Вначале с точки зрения первого игрока выявляют его стратегии, над которыми есть доминирующие, а также его дублирующие стратегии. Соответствующие строки платежной матрицы вычеркивают. Для сокращенной матрицы проделывают то же самое, но с точки зрения второго игрока. Упрощение игры считается завершенным, если на очередном шаге в платежной матрице будут отсутствовать доминирующие и дублирующие стратегии для первого и второго игроков.

Замечание. *Если исходная платежная матрица имеет седловую точку в чистых стратегиях, то процесс упрощения игры приведет к матрице размера* 1×1, *состоящей из одного элемента* (*седловой точки*).

Пример 4.7. Рассмотрим игру с платежной матрицей

$$\begin{pmatrix} 2 & 4 & 7 & 5 \\ 7 & 6 & 8 & 7 \\ 5 & 3 & 4 & 1 \\ 7 & 6 & 0 & 9 \end{pmatrix}.$$

Нетрудно видеть, что стратегия первого игрока A_2 доминирует над стратегиями A_1 и A_3, которые можно отбросить. В оставшейся матрице с точки зрения второго игрока стратегия B_2 доминирует над стратегиями B_1 и B_4. Отбрасывая последние две стратегии, получим упрощенную игру

	B_2	B_3
A_2	6	8
A_4	6	0

Возвращаясь на точку зрения первого игрока, мы видим, что в последней игре стратегия A_2 доминирует над стратегией A_4. Отбрасывая последнюю стратегию и сравнивая оставшиеся стратегии второго игрока, приходим к выводу, что стратегия B_3 должна быть отброшена. Получилась игра размера 1×1 и седловая точка (A_2, B_2) в чистых стратегиях.

4.8. Типичный пример применения теории игр в экономике

Пусть предприятие может выпускать три вида продукции: A, B и C. Прибыль от реализации продукции зависит от уровня спроса, о котором ничего неизвестно, но он принимает три возможных значения. Прибыль от реализации каждого изделия вида A, B или C в зависимости от уровня спроса представлена в таблице:

Спрос / Продукц.	I	II	III
A	8	3	10
B	3	10	1
C	10	1	12

Эту ситуацию можно рассматривать как игру двух игроков, если предприятие будет считать, что спрос ведет себя самым невыгодным для него образом. Рассматривая соответствующую игру в смешанных стратегиях, нетрудно найти седловую точку, в которой вероятности стратегий p_1^0, p_2^0, p_3^0 первого игрока (предприятия) укажут удельные веса каждого вида продукции, которые обеспечат предприятию оптимальную гарантированную прибыль при самом неблагоприятном поведении спроса. Игры, подобные рассмотренной, носят название *игр с природой*. Решение в рамках матричной игры с нулевой суммой позволяет принимать решения, являющиеся наилучшими при самых неблагоприятных условиях. Однако при более благоприятных условиях это решение может оказаться далеко не лучшим.

4.9. Другие разновидности игр

В настоящее время теория игр является довольно развитым разделом математики, содержащим различные обобщения матричных игр с нулевой суммой. Рассмотрим кратко основные направления таких обобщений. Отказ от абсолютной противоположности интересов игроков (от антагонистичности игры) приводит к неантагонистическим играм, простейшими из которых являются *биматричные игры*. Биматричной игрой называется игра двух игроков с ненулевой суммой, в которой выигрыши задаются двумя матрицами отдельно для каждого из игроков. В каждой матрице строка соответствует стратегии первого игрока, а столбец — стратегии второго игрока. При этом на пересечении строки и столбца в первой матрице находится выигрыш первого игрока, а во второй — второго. Если считать, что первый игрок имеет n стратегий, i=1, 2, …, n, а у второго игрока имеется m стратегий, $j = 1$, 2, …, m, то выигрыши первого и второго игроков соответственно задаются матрицами

$$A = \begin{pmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1} & \dots & a_{ij} & \dots & a_{im} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{nj} & \dots & a_{nm} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & \dots & b_{1j} & \dots & b_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ b_{i1} & \dots & b_{ij} & \dots & b_{im} \\ \dots & \dots & \dots & \dots & \dots \\ b_{n1} & \dots & b_{nj} & \dots & b_{nm} \end{pmatrix}.$$

Обычная матричная игра является частным случаем биматричной при $a_{ij}= -b_{ij}$. Будем по-прежнему называть полный набор вероятностей $\vec{p} =\{p_1, p_2, \dots , p_n\}$ применения первым игроком своих чистых стратегий смешанной стратегией первого игрока, и $\vec{q} =\{q_1,\ q_2,\ \dots\ ,\ q_m\}$ — смешанной стратегией второго игрока. Тогда средние выигрыши первого и второго игроков соответственно равны

$$M_1(\vec{p},\vec{q}) = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} p_i q_j, \quad M_2(\vec{p},\vec{q}) = \sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij} p_i q_j.$$

Вместо понятия седловой точки здесь вводится понятие *точки равновесия биматричной игры*. Это пара смешанных стратегий первого и второго игроков $\vec{p}^{\,0} = \left\{p_1^0, p_2^0, \dots, p_n^0\right\}$ и $\vec{q}^{\,0} = \left\{q_1^0, q_2^0, \dots, q_m^0\right\}$, отступление от которых не выгодно обоим игрокам, то есть, для которых выполнены неравенства

$$M_1(\vec{p}^{\,0},\vec{q}^{\,0}) \ge M_1(\vec{p},\vec{q}^{\,0}); \quad M_2(\vec{p}^{\,0},\vec{q}^{\,0}) \ge M_2(\vec{p}^{\,0},\vec{q})$$

при любых смешанных стратегиях $\vec{p}$ и $\vec{q}$.

Теорема 4.2. *Всякая биматричная игра имеет, хотя бы одну, точку равновесия.*

Существуют способы нахождения точек равновесия, однако эти способы значительно сложнее, чем для матричных игр с нулевой суммой.

Другим естественным направлением обобщения матричных игр являются *бесконечные игры*, то есть игры, в которых хотя бы один из игроков имеет бесконечное количество возможных стратегий. При формализации таких игр будем обозначать их через $\Gamma(X, Y, a_1, a_2)$, где X, Y — множества возможных стратегий первого и второго игроков соответственно, а $a_i(x,y)$ — выигрыш i-го игрока. Часто каждую стратегию из множеств X и Y можно взаимно однозначно сопоставить определенному числу из единичного интервала (0,1). В этом случае функции $a_i(x,y)$ являются функциями двух аргументов, определенных на внутренности единичного квадрата, а игру называют игрой на единичном квадрате.

Бесконечная игра называется антагонистической, если $a_2(x,y)=-a_1(x,y)=-a(x,y)$. Такую игру обозначим $\Gamma(X, Y, a)$. *Бесконечные антагонистические игры* изучены лучше остальных и во многом аналогичны матричным играм с нулевой суммой, хотя и имеют свои особенности. Их можно анализировать как в чистых, так и в смешанных стратегиях. Назовем чистой нижней ценой игры величину

$$\alpha = \max_{x\in X} \min_{y\in Y} a(x, y),$$

а чистой верхней ценой игры величину

$$\beta = \min_{y\in Y} \max_{x\in X} a(x, y).$$

Для матричных игр величины α и β всегда существуют, а для бесконечных игр они могу не существовать. Если $\alpha = \beta = V$, то такая игра имеет решение в чистых стратегиях, т. е. оптимальной стратегией первого игрока является выбор числа $x_0\in X$ и второго игрока — числа $y_0\in Y$, при которых $a(x_0,y_0) = V$. В этом случае V называется ценой игры, а (x_0,y_0) — седловой точкой игры в чистых стратегиях.

Если игра $\Gamma(X, Y, a)$ не имеет седловой точки в чистых стратегиях, то оптимальные стратегии можно искать среди смешанных стратегий. Рассмотрим игру на единичном квадрате. Пусть $F(x)$ — функция распределения вероятностей применения чистых стратегий первым игроком, т. е., если ξ — чистая стратегия первого игрока, то

$$F(x) = P\{\xi \le x\},$$

где $P\{\xi\le x\}$ означает вероятность того, что случайно выбранная чистая стратегия ξ не будет превосходить x. Аналогично рассматривается функция распределения вероятностей применения чистых стратегий η вторым игроком

$$G(y)=P\{\eta \le y\}.$$

Функции $F(x)$ и $G(y)$ называются смешанными стратегиями соответственно первого и второго игроков. Определим средний выигрыш первого игрока, как математическое ожидание случайной величины $a(\xi,\eta)$

$$E(F,\ G)=M(a(\xi,\eta))=\int_0^1\int_0^1 a(x,y)dF(x)dG(y)\ .$$

В антагонистической бесконечной игре $\Gamma(X,\ Y,\ a)$ пара смешанных стратегий $F^*(x)$ и $G^*(y)$ соответственно для первого и второго игроков образует седловую точку в смешанных стратегиях, если для любых смешанных стратегий $F(x)$ и $G(y)$ справедливы соотношения

$$E(F,\ G^*)\leq E(F^*,\ G^*)\leq E(F^*,\ G).$$

В отличие от матричных игр с нулевой суммой, для бесконечных антагонистических игр седловая точка в смешанных стратегиях существует не всегда.

Теорема 4.3. *Всякая бесконечная антагонистическая игра двух игроков* $\Gamma(X,\ Y,\ a)$ *с непрерывной функцией выигрышей* $a(x,y)$ *на единичном квадрате имеет решение* (*игроки имеют оптимальные смешанные стратегии*, *отвечающие седловой точке в смешанных стратегиях*).

Коснемся еще одного направления обобщения понятия игры. Реальные конфликтные ситуации часто приводят к формализации в виде игры с количеством игроков больше двух, например, n игроков. Такие игры называются *играми* n *участников* (игроков). Поскольку в них участвуют не менее трех игроков, то возможны два варианта правил:

1) игрокам не разрешается вступать в соглашения,
2) игрокам разрешается вступать в соглашения.

В первом случае каждый игрок должен самостоятельно и независимо от желаний других выбирать свои стратегии с целью максимального увеличения своего выигрыша, т. е. игрокам не разрешается вступать к
коалиции. Поэтому такая игра называется *бескоалиционной*. Во втором случае некоторые игроки могут по соглашению объединяться (кооперироваться) в действиях против других игроков, образовывая коалиции с целью максимизации выигрыша коалиции. Такие игры называются коалиционными или (в некоторых случаях) кооперативными. Методы исследования игр n участников существенно зависят от возможности образования или запрета коалиций.

Подробнее с математической теорией игр можно ознакомиться по книгам [19], [24].

4.10. Задачи теории статистических решений

Близкой по идеям и методам к теории игр является теория статистических решений. В отличие от теории игр второй игрок здесь вовсе не имеет никаких интересов, не пытается противодействовать, но выбор им своей стратегии заранее неизвестен. Как правило, в роли этого второго игрока выступает комплекс объективных условий, который обычно называют *природой*. С такой ситуацией мы уже встречались. Для принятия решения здесь нужно иметь какую-нибудь информацию о поведении природы. Если предположить, что она ведет себя самым неблагоприятным образом, то решать задачу можно так же, как в теории игр. Но обычно такое решение в итоге оказывается не самым выгодным. Анализ матрицы выигрышей здесь часто заменяют анализом матрицы рисков.

Под риском для стратегии A_i при данном комплексе условий Π_j понимают величину $r_{ij} = \max\limits_k \left(a_{kj}\right) - a_{ij}$. Например, при игре с природой, имеющей матрицу выигрышей

	$П_1$	$П_2$	$П_3$	$П_4$
A_1	1	4	5	9
A_2	3	8	4	3
A_3	4	6	6	2
$\max\limits_k a_{kj}$	4	8	6	9

соответствующая матрица рисков имеет такой вид:

	$П_1$	$П_2$	$П_3$	$П_4$
A_1	3	4	1	0
A_2	1	0	2	6
A_3	0	2	0	7

Риск—это плата за отсутствие информации. В нашем примере $r_{21} = 1$; $r_{24} = 6$, тогда как выигрыши a_{ij} в обоих случаях одинаковы. Если бы мы узнали, что природа находится в состоянии $П_1$, то наш максимальный выигрыш получим, применив стратегию A_3, и этот выигрыш равен 4.

Если информации нет, и мы применяем стратегию A_2, то рискуем недополучить сумму $r_{21} = 1$. При состоянии природы $П_4$ и применении стратегии A_2 риск больше ($r_{24} = 6$). При выборе стратегий в игре с природой естественно стремиться или к получению максимального выигрыша, или к минимизации риска, сопровождающего выбор решения.

Пусть о состоянии природы имеется информация в виде вероятностей $q_1, q_2, \ldots, q_m$ соответствующих комплексов условий $П_1$, $П_2$, ... , $П_m$,. Если нам нужно выбрать решение в чистых стратегиях, то естественно выбирать ту стратегию A_i для которой среднее значение выигрыша максимально. Причём выбрать i так, чтобы

$$a_i = \sum_{j=1}^{m} a_{ij} q_j \to \max.$$

В другом случае своя стратегия A_i выбирается из соображений минимальности среднего риска $r_i = \sum\limits_{j=1}^{m} r_{ij} q_j \to \min$ Можно показать, что оба способа выбора решения приводят к одинаковому результату.

Задача выбора решения ставится и в смешанных стратегиях. В этом случае приходится решать простые задачи линейного программирования. Если $\{p_1, p_2, \ldots, p_n\}$ — искомая смешанная стратегия, то она должна быть или точкой максимума задачи

$$z = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} a_{ij} q_j \right) p_i \to \max,$$

$$p_1 + p_2 + \cdots + p_n = 1,$$

$$p_i \geq 0, \ i = 1, 2, \ldots, n;$$

или точкой минимума другой задачи:

$$z_r = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} r_{ij} q_j \right) p_i \to \min,$$

$$p_1 + p_2 + \cdots + p_n = 1,$$

$$p_i \geq 0, \ i = 1, 2, \ldots, n.$$

Очень часто вероятности q_j в принципе существуют, но неизвестны. Иногда при этом пользуются принципом недостаточного основания Лапласа, который состоит в том, что $q_1 = q_2 = \cdots = q_m = \frac{1}{m}$. Однако чаще всего предпринимают меры для определения хотя бы ориентировочных значений этих вероятностей. При этом можно воспользоваться, например, методом экспертных оценок. В некоторых случаях для определения статистических данных о состоянии природы производят специальные эксперименты, используют анализ уже имеющегося статистического материала.

В случае, когда вероятности состояний природы либо вообще не существуют, либо не поддаются оценке даже приближенно, объективный выбор решения становится невозможным. Здесь все зависит от точки зрения на ситуацию, от позиции принимающего решение лица. Опишем несколько возникающих при этом критериев при принятии решений в чистых стратегиях.

Максиминнный критерий Вальда. Природа здесь рассматривается как разумный агрессивный противник, стремящийся создать самые неблагоприятные условия. Выбор оптимальной стратегии производится так же, как и при анализе игры в чистых стратегиях, то есть выбирается стратегия, гарантирующая в любом случае выигрыш не меньший, чем $\alpha = \max_i \min_j a_{ij}$. Принимающее решение лицо здесь находится на точке зрения "крайнего пессимизма".

Критерий минимаксного риска Сэвиджа. Здесь в качестве оптимальной стратегии выбирается стратегия, при которой величина риска в наихудших условиях минимальна. При этом находится величина $S = \min_i \max_j r_{ij}$.

Оценка стратегий производится тоже с точки зрения "крайнего пессимизма", но "пессимизм" понимается по-иному.

Критерий пессимизма-оптимизма Гурвица. Выбор стратегии производится в процессе вычисления величины

$$H = \max_i \left[\theta \min_j a_{ij} + (1-\theta) \max_j a_{ij} \right],$$

где θ — "коэффициент пессимизма", выбираемый между нулем и единицей. При $\theta = 1$ критерий Гурвица превращается в критерий Вальда; при $\theta = 0$ — в критерий “крайнего оптимизма", выбирающий стратегию, для которой самый большой выигрыш в строке максимален.

Контрольные вопросы и задачи для самостоятельного решения

1. Что обычно называют конфликтной ситуацией? Как строится простейшая модель конфликтной ситуации в виде матричной игры двух игроков с нулевой суммой?
2. Как игроки оценивают свои стратегии в процессе анализа игры в чистых стратегиях? Что такое нижняя и верхняя цены игры в чистых стратегиях?
3. Что такое седловая точка игры в чистых стратегиях?
4. Что такое смешанная стратегия игрока? Дайте определение платежной функции игры.
5. Что такое седловая точка игры в смешанных стратегиях? Сформулируйте теорему фон Неймана о существовании седловой точки игры в смешанных стратегиях.
6. Как строится пара двойственных задач для определения седловой точки в смешанных стратегиях?
7. В чем состоит графический метод решения игр размера $2 \times m$ и $n \times 2$?
8. Как решить игру в смешанных стратегиях двойственным симплекс-методом?
9. Дайте определения биматричной игры и ее точки равновесия.
10. Что такое бесконечная антагонистическая игра, ее чистые и смешанные стратегии?
11. Что вы знаете об играх n лиц? Когда игра называется бескоалиционной, а когда — коалиционной или кооперативной?
12. Что обычно называют игрой с природой? Постановка задачи различные подходы к ее решению. Что такое матрица рисков?
13. В чем состоит принцип недостаточного основания Лапласа?
14. Что представляют собой критерии Вальда, Сэвиджа и Гурвица?

В задачах 4.1 — 4.4 *для заданной платежной матрицы произвести анализ игры в чистых стратегиях.*

4.1. $\begin{pmatrix} 2 & -5 & 3 & -1 \\ 4 & 7 & -6 & 8 \\ -9 & 11 & 2 & 3 \\ -4 & 2 & -7 & 12 \end{pmatrix}$.

4.2. $\begin{pmatrix} 2 & 4 & -9 & -4 \\ -5 & 7 & 11 & 2 \\ 3 & -6 & 2 & -7 \\ -1 & 8 & 3 & 12 \end{pmatrix}$.

4.3. $\begin{pmatrix} 1 & -2 & 3 & -4 & 5 \\ 6 & -7 & 8 & -9 & 10 \\ 5 & -4 & 3 & -2 & 1 \end{pmatrix}$.

4.4. $\begin{pmatrix} 4 & -2 & 3 \\ -5 & 6 & -7 \\ 8 & -9 & 12 \end{pmatrix}$.

В задачах 4.5 — 4.8 *найти седловую точку игры в смешанных стратегиях, используя графический метод.*

4.5.

$$\begin{pmatrix} 7 & 10 \\ 9 & 6 \\ 8 & 9 \end{pmatrix}$$

Ответ: $\begin{cases} \vec{p}^{(0)} = (0;1/4;3/4), \\ \vec{q}^{(0)} = (3/4;1/4). \end{cases}$

4.6.

$$\begin{pmatrix} 6 & 5 \\ 4 & 6 \\ 2 & 7 \\ 1 & 8 \end{pmatrix}$$

Ответ: $\begin{cases} \vec{p}^{(0)} = (7/8;0;0;1/8), \\ \vec{q}^{(0)} = (3/8;5/8). \end{cases}$

4.7.

$$\begin{pmatrix} 3 & 5 & 1 & 4 & 6 \\ 7 & 4 & 9 & 5 & 3 \end{pmatrix}$$

Ответ: $\begin{cases} \vec{p}^{(0)} = (1/2;1/2), \\ \vec{q}^{(0)} = (0;0;0;3/4;1/4). \end{cases}$

4.8.

$$\begin{pmatrix} 3 & 9 \\ 5 & 4 \\ 1 & 7 \\ 4 & 5 \\ 6 & 3 \end{pmatrix}$$

Ответ: $\begin{cases} \vec{p}^{(0)} = (1/3;0;0;0;2/3), \\ \vec{q}^{(0)} = (2/3;1/3). \end{cases}$

В задачах 4.9 — 4.12 *найти решения игр сведением к парам двойственных задач линейного программирования.*

4.9.

$$\begin{pmatrix} 3 & 4 & 5 \\ 7 & 6 & 4 \end{pmatrix}$$

4.10.

$$\begin{pmatrix} 4 & 7 \\ 9 & 5 \\ 5 & 9 \\ 6 & 9 \end{pmatrix}$$

Ответ: $\begin{cases} \vec{p}^{(0)} = (3/5;2/5), \\ \vec{q}^{(0)} = (1/5;0;4/5). \end{cases}$

Ответ: $\begin{cases} \vec{p}^{(0)} = (0;3/7;0;4/7), \\ \vec{q}^{(0)} = (4/7;3/7). \end{cases}$

4.11.

$$\begin{pmatrix} 8 & 4 & 7 \\ 6 & 5 & 9 \\ 7 & 7 & 8 \end{pmatrix}$$

Ответ: $\begin{cases} \vec{p}^{(0)} = (0;0;1), \\ \vec{q}^{(0)} = (0;1;0). \end{cases}$

4.12.

$$\begin{pmatrix} 7 & 6 & 7 & 5 \\ 6 & 7 & 9 & 8 \\ 5 & 8 & 4 & 6 \end{pmatrix}$$

Ответ: $\begin{cases} \vec{p}^{(0)} = (1/2;1/2;0), \\ \vec{q}^{(0)} = (3/4;0;0;1/4). \end{cases}$

5. ЭЛЕМЕНТЫ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ

5.1. Некоторые задачи, приводящие к требованию целочисленности. Постановка задач дискретного программирования

Среди практически важных задач отыскания условного экстремума линейной функции особое место занимают задачи с требованием целочисленности всех или части переменных. Они получили название задач целочисленного (частично целочисленного) программирования. Исторически первой задачей целочисленного типа явилась, опубликованная венгерским математиком Е. Эгервари в 1932 году, задача о назначении персонала.

Задача о назначениях. Пусть требуется выполнить n видов различных работ и имеется n исполнителей (машин или людей) для их выполнения. Каждый исполнитель может использоваться на любой работе. Производительности исполнителей на различных работах, вообще говоря, различны. Обозначим через c_{ij} производительность i-го исполнителя на j-ой работе. Задача заключается в таком распределении исполнителей по работам, при котором суммарная производительность максимальна.

Построим математическую модель этой задачи. Для описания каждого варианта распределения исполнителей по работам введем переменные x_{ij}, относительно которых условимся, что x_{ij}=1, если i-й исполнитель назначен на j-ю работу, и x_{ij}=0 – в противном случае. Выбор варианта назначения исполнителей на работы равносилен приданию переменным x_{ij} значений 0 или 1. При этом должны выполняться условия

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n\,,$$

которые означают, что каждый исполнитель назначается на работу, а также условия

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n\,,$$

которые означают, что каждая работа обеспечивается исполнителем. Суммарная производительность при данном варианте назначений выразится суммой

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\,x_{ij}\,.$$

Таким образом, получаем следующую математическую модель задачи

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij} \to \max;$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n, \ \sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n,$$

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}, \quad i = 1,2,\ldots,n, \quad j = 1,2,\ldots,n.$$

Последние из условий носят название условий булевости переменных x_{ij}, а задачи математического программирования с такими условиями называются задачами с булевыми переменными. Отметим, что замена условий булевости условиями неотрицательности переменных превращает нашу задачу в обычную закрытую транспортную задачу, если ее сформулировать как задачу на минимум. Если эту задачу решить методом потенциалов, то полученное решение автоматически будет удовлетворять условию булевости. Таким образом, в задаче о назначениях условия булевости переменных можно заменить условиями их неотрицательности. Отметим, что возможен открытый вариант задачи о назначениях, в котором количество исполнителей и видов работ не совпадают. С помощью введения фиктивных исполнителей или фиктивных видов работ эта задача сводится к такой же закрытой задаче, в которой производительности фиктивных пунктов полагаются равными нулю. При решении задачи некоторые исполнители могут оказаться без работы или некоторые виды работ — без исполнителей. При этом по-прежнему условия булевости можно заменить условиями неотрицательности.

Пример 5.1. Фирма объявила набор работников на две появившиеся вакансии. Были поданы заявления от трех претендентов, готовых занять каждую из двух вакансий. Предварительные испытания определили, что производительности претендентов на работе по каждой из двух вакансий задаются матрицей

$$\begin{pmatrix} 4 & 7 & 1 \\ 5 & 3 & 6 \end{pmatrix}.$$

Требуется так заполнить вакансии имеющимися претендентами, чтобы суммарная производительность принятых работников была максимальной. Предполагается, что каждая вакансия заполняется лишь одним претендентом, а каждый претендент может занять не более одной вакансии.

Для составления математической модели задачи введем такие переменные x_{ij}, что x_{ij}=1, если i-й исполнитель назначен на j-ю работу, и x_{ij}=0 — в противном случае. Получаем следующую задачу на максимум

$$\begin{cases} x_{11} + x_{12} + x_{13} = 1, \\ x_{21} + x_{22} + x_{23} = 1; \end{cases} \qquad \begin{cases} x_{11} + x_{21} \le 1, \\ x_{12} + x_{22} \le 1, \\ x_{13} + x_{23} \le 1; \end{cases}$$

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}, \quad i = 1,2, \quad j = 1,2,3.$$

После введения фиктивной вакансии мы получаем задачу, в которой условия булевости можно заменить условиями неотрицательности переменных. Задачу можно рассматривать как транспортную задачу на минимум целевой функции $z = -4x_{11} - 7x_{12} - x_{13} - 5x_{21} - x_{22} - 6x_{23} \to \max$, которую можно записать в виде следующей таблицы данных

		v_1	v_2	v_3
	Запасы	Потребности		
		1	1	1
u_1	1	$\boxed{-4}$	$\boxed{-7}$ 1	$\boxed{-1}$ 0
u_2	1	$\boxed{-5}$ 1	$\boxed{-1}$ 0	$\boxed{-6}$ 0
u_3	1	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$ 1

Находим первое опорное решение методом наименьшей стоимости. Запишем и решим систему уравнений для потенциалов.

$$\begin{cases} u_1 + v_2 = -7, \\ u_1 + v_3 = -1, \\ u_2 + v_1 = -5, \\ u_2 + v_3 = -6, \\ u_3 + v_3 = 0. \end{cases} \qquad \begin{cases} u_1 = 0, \quad v_1 = 0, \\ u_2 = -5, \quad v_2 = -7, \\ u_3 = 1, \quad v_3 = -1. \end{cases}$$

Вычислим коэффициенты γ_{ij} для свободных клеток. γ_{11}=–4, γ_{22}=11, γ_{31}=–1, γ_{32}=6. Составим цикл пересчета для клетки x_{31} и произведем сдвиг по этому циклу на единицу. Получим следующую таблицу

		v_1	v_2	v_3
	Запасы	Потребности		
		1	1	1
u_1	1	$\boxed{-4}$ 0	$\boxed{-7}$ 1	$\boxed{-1}$
u_2	1	$\boxed{-5}$ 1	$\boxed{-1}$	$\boxed{-6}$
u_3	1	$\boxed{0}$ 0	$\boxed{0}$	$\boxed{0}$ 1

Снова составим и решим систему уравнений для потенциалов.

$$\begin{cases} u_1 + v_1 = -4, \\ u_1 + v_2 = -7, \\ u_2 + v_1 = -5, \\ u_3 + v_1 = 0, \\ u_3 + v_3 = 0. \end{cases} \qquad \begin{cases} u_1 = 0, & v_1 = -4, \\ u_2 = -1, & v_2 = -7, \\ u_3 = 4, & v_3 = -4. \end{cases}$$

Следовательно, γ_{13}=1, γ_{22}=7, γ_{23}=–1, γ_{32}=3. Составим цикл пересчета для клетки x_{23} и произведем сдвиг по этому циклу. Получим таблицу

		v_1	v_2	v_3
	Запасы	Потребности		
		1	1	1
u_1	1	$\boxed{-4}$ 1	$\boxed{-7}$ 0	$\boxed{-1}$
u_2	1	$\boxed{-5}$	$\boxed{-1}$ 1	$\boxed{-6}$
u_3	1	$\boxed{0}$ 0	$\boxed{0}$	$\boxed{0}$ 1

$$\begin{cases} u_1 + v_1 = -4, \\ u_1 + v_2 = -7, \\ u_2 + v_2 = -1, \\ u_3 + v_1 = 0, \\ u_3 + v_3 = 0. \end{cases} \qquad \begin{cases} u_1 = 0, & v_1 = -4, \\ u_2 = 6, & v_2 = -7, \\ u_3 = 4, & v_3 = -4. \end{cases}$$

$\gamma_{13}=3$, $\gamma_{21}=-7$, $\gamma_{23}=-8$, $\gamma_{32}=3$. Производим сдвиг по циклу пересчета для клетки x_{21}.

		v_1	v_2	v_3
	Запасы	Потребности		
		1	1	1
u_1	1	$\boxed{-4}$ 1	$\boxed{-7}$ 0	$\boxed{-1}$
u_2	1	$\boxed{-5}$	$\boxed{-1}$ 1	$\boxed{-6}$
u_3	1	$\boxed{0}$ 0	$\boxed{0}$	$\boxed{0}$ 1

$$\begin{cases} u_1 + v_1 = -4, \\ u_1 + v_2 = -7, \\ u_2 + v_1 = -5, \\ u_3 + v_1 = 0, \\ u_3 + v_3 = 0. \end{cases} \qquad \begin{cases} u_1 = 0, & v_1 = -4, \\ u_2 = -1, & v_2 = -7, \\ u_3 = 4, & v_3 = -4. \end{cases}$$

$\gamma_{13}=3$, $\gamma_{22}=7$, $\gamma_{23}=-1$, $\gamma_{32}=3$. Преобразуем таблицу с помощью цикла пересчета для клетки x_{23}.

		v_1	v_2	v_3
	Запасы	Потребности		
		1	1	1
u_1	1	$\boxed{-4}$ 0	$\boxed{-7}$ 1	$\boxed{-1}$
u_2	1	$\boxed{-5}$	$\boxed{-1}$	$\boxed{-6}$ 1
u_3	1	$\boxed{0}$ 1	$\boxed{0}$	$\boxed{0}$ 0

$$\begin{cases} u_1 + v_1 = -4, \\ u_1 + v_2 = -7, \\ u_2 + v_3 = -6, \\ u_3 + v_1 = 0, \\ u_3 + v_3 = 0. \end{cases} \qquad \begin{cases} u_1 = 0, & v_1 = -4, \\ u_2 = -2, & v_2 = -7, \\ u_3 = 4, & v_3 = -4. \end{cases}$$

$\gamma_{13} = 3$, $\gamma_{21} = 1$, $\gamma_{22} = 8$, $\gamma_{32} = 3$. Получена последняя таблица, содержащая решение задачи. Второй претендент получает первую вакансию, третий — вторую, а первый остается безработным.

Не во всякой задаче с условиями целочисленности эти последние условия можно игнорировать. В большинстве случаев они играют очень существенную роль.

Задача о ранце. Имеется вектор ограниченных ресурсов $\vec{b} = \{b_1, b_2, \ldots, b_m\}$, которые можно использовать для перевозки различных по своим характеристикам грузов. Каждый из этих n видов грузов имеет следующие свойства:

1) неделимость, т.е. для транспортировки груз с номером j может выбираться в количестве кратном единице;
2) определена полезность (или стоимость) c_j единицы груза;
3) известным является расход i-го ресурса для перевозки единицы j-го груза a_{ij}, $i = 1,2,\ldots,m$, $j = 1,2,\ldots,n$.

Требуется определить такой набор груза различных видов, при котором общая полезность рейса максимальна. Под полезностью рейса понимается суммарная стоимость перевезенного за рейс груза.

Построим математическую модель задачи. Пусть x_j — количество выбранных для перевозки предметов j-го вида. Требованию неделимости соответствует условие

$$x_j \geq 0,\ x_j \text{ — целые числа},\ j = 1,2,\ldots,n.$$

Сопоставляя расход ресурсов при транспортировке с их наличным количеством, получаем ограничения

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

Общая полезность рейса определяется значением целевой функции

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

Частным случаем этой задачи является классическая задача о ранце, в которой переменные x_j подчинены условию булевости. Смысл этих условий заключается в том, что любой из заданного набора предметов (грузов) может быть выбран для перевозки или нет, т.е. x_j может принимать значения 0 или 1.

Часто возникают и другие подобные задачи, которые можно сформулировать в виде следующей задачи на максимум

$$z = \sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \{\le = \ge\} b_i, \quad i = 1,2,\dots,m \tag{P}$$

$$x_j \ge 0, \qquad j = 1,2,\dots,n,$$

$$x_j \text{ – целые числа}, \qquad j = 1,2,\dots,k.$$

Задачу (P) при $k<n$ будем называть задачей частично целочисленного программирования. При $k=n$ сформулированная задача называется задачей целочисленного программирования. Более общими являются задачи дискретного или частично дискретного программирования. В этих задачах все или часть переменных принимают значения из заранее заданного дискретного множества. Ниже мы в основном будем рассматривать целочисленные задачи.

Основные понятия обычного линейного программирования переносятся и на дискретный случай. Вектор $X = \{x_1, x_2, \dots, x_n\}$, удовлетворяющий системе ограничений задачи, называется допустимым решением (планом) задачи. Допустимое решение, доставляющее максимум целевой функции, называют оптимальным решением (планом). Процесс решения задачи дискретного программирования состоит в нахождении оптимального плана.

Казалось бы, естественный путь решения целочисленной задачи состоит в решении соответствующей обычной линейной задачи с последующим округлением компонент оптимального плана $X = \{x_1, x_2, \dots, x_n\}$ 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 5.2. Рассмотрим задачу целочисленного программирования с двумя переменными

$$z = x_1 + x_2 \to \max,$$

$$\begin{cases} x_1 + x_2 \le 7, \\ 4x_1 - 5x_2 \le 5, \\ 2x_1 + 11x_2 \le 38; \end{cases}$$

$$x_1 \ge 0,\ x_2 \ge 0,\ x_1,\ x_2 \text{ — целые.}$$

Отбросим условия целочисленности и решим задачу графически. Экстремум достигается в точках $A(\frac{13}{3}, \frac{8}{3})$ и $B(\frac{40}{9}, \frac{23}{9})$, а также в любой точке отрезка AB, и равен 7. Округляя значения координат A, получим точку $A'(4,3)$, которая не принадлежит области поиска. Можно показать, что максимум достигается в точках $N(3,2)$ и $M(2,3)$ и равен 5.

Рассмотренный пример показывает, что для задач дискретного программирования необходимо применять особые методы оптимизации.

5.2. Методы отсечения. Первый алгоритм Гомори

Методы отсечения сводятся к решению некоторой последовательности специально построенных задач линейного программирования без условий целочисленности. Каждая последующая задача получается из предыдущей добавлением дополнительного линейного ограничения (неравенства), называемого сечением. Если обозначить через $З_0$, $З_1$, $З_2$, …, $З_l$, … задачи указанной последовательности, то l-ым сечением называется линейное ограничение, вводимое в задачу $З_{l-1}$ для образования задачи $З_l$, и удовлетворяющее двум условиям:

a) любое целочисленное решение системы ограничений задачи $З_{l-1}$ ему удовлетворяет;

b) найденное нецелочисленное решение задачи $З_{l-1}$ ему не удовлетворяет ("отсекается").

Рассмотрим задачу целочисленного программирования, которая имеет следующий вид

$$z = c_0 + (\vec{c}, \vec{x}) \to \max,$$
$$A\vec{x} = \vec{b}, \ \ \vec{x} \ge \vec{0}, \qquad (P_1)$$
$$x_j - \textit{целые} \ (j = 1,2,\dots,n),$$

то есть является задачей с ограничениями равенствами. Частным случаем такой задачи можно считать задачу вида

$$z = \sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad i = 1,2,\dots,m, \qquad (P_2)$$

$$x_j \ge 0, \quad x_j - \text{целые числа}, \qquad j = 1,2,\dots,n \ ,$$

в которой предполагается, что величины c_j, b_i и a_{ij} также являются целочисленными. Задачу (P_2) легко свести к задаче вида (P_1) с помощью уравнивания неравенств. Задачу вида (P_1) (или (P_2)) будем называть канонически целочисленной.

Существует несколько методов построения сечений для целочисленных задач. В этом параграфе мы рассмотрим наиболее простой первый алгоритм Р. Гомори, который применяется к задаче вида (P_1), т.е. к канонически целочисленной задаче.

Пусть задача З$_{l\text{-}1}$ уже решена симплекс методом и ее решение $\overline{X}_{l-1}$ не удовлетворяет условию целочисленности. Обозначим через $\{a\}$ дробную часть числа a, $\{a\}=a-[a]$, где $[a]$–целая часть числа a. Пусть k — индекс свободных переменных в последней симплексной таблице. Обозначим через s — номер строки в этой таблице с *наибольшим* значением $\{b_s^*\}$ для свободного члена b_s^*, отвечающего базисной переменной. Тогда сечение Гомори запишется в виде

$$\{b_s^*\}-\sum_k \{a_{sk}^*\}x_k \le 0. \qquad (S_1)$$

Не удовлетворяющее условию целочисленности решение $\overline{X}_{l-1}$ задачи З$_{l\text{-}1}$ условию (S_1) не удовлетворяет. Действительно, в оптимальном опорном решении свободные переменные x_k=0, а поскольку $\{b_s^*\}>0$, то (S_1) на $\overline{X}_{l-1}$ не удовлетворяется. Следовательно, ограничение (S_1) удовлетворяет условию b), упомянутому выше. Покажем, что условие (S_1) удовлетворяет и условию a). Сначала мы установим, что оно удовлетворяется для любого целочисленного решения системы ограничений задачи (P_1). Рассмотрим i-ое ограничение этой задачи $\sum_{j=1}^{n} a_{ij}x_j = b_i$, и возьмем дробную часть от обеих частей равенства. Поскольку дробная часть суммы не превосходит суммы дробных частей слагаемых ($\{a+b\}\le\{a\}+\{b\}$), то справедливо неравенство $\sum_{j=1}^{n}\{a_{ij}x_j\}\ge\{b_i\}$. Поскольку x_j — целое неотрицательное число, то $\{a_{ij}x_j\}\le\{a_{ij}\}x_j$. Таким образом, для любого целочисленного решения выполнено неравенство $\sum_{j=1}^{n}\{a_{ij}\}x_j\ge\{b_i\}$. Это означает, что условие а) выполнено для сечения (S_1), построенного после решения задачи З$_0$, при построении задачи З$_1$. Для решения задачи З$_1$ симплекс методом неравенство, задающее сечение, нужно уровнять введением новой переменной. Можно показать, что эту новую переменную можно вывести из числа базисных переменных. В процессе решения задачи З$_1$ эта переменная в число базисных вернуться не может, так как она не входит в выражение для целевой функции. Поэтому мы можем рассматривать задачу З$_1$ в качестве задачи целочисленного программирования с увеличенным на единицу количеством переменных. В силу сказанного выше, построенное после решения задачи З$_1$ сечение будет обладать свойствами *a*), и *b*). То же можно сказать и о задаче З$_{l-1}$ при любом l.

Алгоритм решения канонически целочисленной задачи состоит из последовательности итераций, каждая из которых включает следующие пункты:

1) решается задача З$_{l-1}$ и находится ее оптимальное решение $\overline{X}_{l-1}$;
2) если решение этой задачи удовлетворяет условию целочисленности, то процесс заканчивается, и мы получаем оптимальный план. В противном случае переходим к пункту 3).
3) На основании последней симплекс-таблицы задачи З$_{l-1}$ записываем сечение Гомори (S_1).
4) Добавление ограничения из предыдущего пункта к условиям задачи З$_{l-1}$ приводит к задаче З$_l$, после чего снова возвращаемся к пункту 1) с увеличенным на единицу l.

Отметим, что при некотором l задача З$_l$ может оказаться не имеющей решения. Очевидно, что в случае недопустимости задачи З$_l$ и исходная задача целочисленного программирования тоже является недопустимой. Нетрудно показать, что если целевая функция задачи З$_l$ неограниченна на области допустимых решений, то это же справедливо и для исходной задачи целочисленного программирования.

Можно доказать, что в результате конечного числа итераций рассмотренного алгоритма мы придем к задаче З$_l$, которая либо имеет целочисленное решение, либо не имеет решений. В первом случае мы получаем решение исходной задачи целочисленного программирования, а во втором — устанавливаем, что эта задача не имеет решения.

Пример 5.3. Требуется найти оптимальный план задачи

$$z = x_1 + 4x_2 \to \max ,$$

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

$$x_i \ge 0, \quad x_i - \text{целые числа } (i = 1{,}2).$$

Отбрасывая условие целочисленности, решаем симплекс-методом задачу З$_0$:

$$z = x_1 + 4x_2 \to \max ,$$

$$\begin{cases} -x_1+2x_2+x_3=2, \\ 3\text{x}_1+2x_2+x_4=6; \end{cases}$$
$$x_i \geq 0, \text{i}=1{,}2.$$

Таблица 1

Базисн. перем.	Своб. члены	x_1	$\downarrow x_2$	x_3	x_4
$\leftarrow x_3$	2	–1	2	1	0
x_4	6	3	2	0	1
z	0	–1	–4	0	0

Таблица 2

Базисн. перем.	Своб. члены	$\downarrow x_1$	x_2	x_3	x_4
x_2	1	–1/2	1	1/2	0
$\leftarrow x_4$	4	4	0	–1	1
z	4	–3	0	2	0

Таблица 3

Базисн. перем.	Своб. члены	x_1	x_2	x_3	x_4
x_2	3/2	0	1	3/8	1/8
x_1	1	1	0	–1/4	1/4
z	7	0	0	5/4	3/4

Из последней симплекс-таблицы получаем оптимальное решение задачи З$_0$: $\overline{X}_0=\{1; 3/2; 0; 0\}$. Это решение не является целочисленным. Построим задачу З$_1$. Единственная строка с нецелочисленным значением b_s — первая строка. Запишем сечение Гомори (S_1)

$$\{3/2\}–(\{3/8\}x_3+\{1/8\}x_4)\leq 0,$$

то есть.

$$1/2–(3/8)x_3–(1/8)x_4\leq 0.$$

Преобразуя и уравнивая неравенство, получим

$$–(3/8)x_3–(1/8)x_4+u_1=–1/2.$$

Добавляя это ограничение к ограничениям задачи З$_0$, получим задачу З$_1$. Решение этой последней задачи удобно начинать, добавив к табл. 3 еще одну строку. Получена таблица, отвечающая псевдоплану. Применим двойственный симплекс метод (метод последовательного уточнения оценок).

Таблица 1

Базисн. перем.	Своб. члены	x_1	x_2	x_3	x_4	u_1
x_2	3/2	0	1	3/8	1/8	0
x_1	1	1	0	1/4	1/4	0
u_1	–1/2	0	0	[–3/8]	–1/8	1
z	7	0	0	5/4	3/4	0

По правилу выбора разрешающего элемента в двойственном симплекс методе выбираем элемент (–3/8) и переходим к следующей таблице

Таблица 2

Базисн. перем.	Своб. члены	x_1	x_2	x_3	x_4	u_1
x_2	1	0	1	0	0	1
x_1	4/3	1	0	0	1/3	–2/3
x_3	4/3	0	0	1	1/3	–8/3
z	16/3	0	0	0	1/3	10/3

Полученный псевдоплан оказывается одновременно оптимальным решением задачи З$_1$. Оно не является целочисленным. Строим соответствующее сечение Гомори

$$\{4/3\}-(\{1/3\}x_4+\{-2/3\}u_1)\leq 0.$$

Поскольку $\{4/3\}=1/3$; $\{1/3\}=1/3$; $\{-2/3\}=1/3$, после преобразований и уравнивания неравенства получаем ограничение

$$-(1/3)x_4-(1/3)u_1+u_2=-1/3.$$

Добавляя это ограничение, получаем задачу З$_2$, которую можно записать в виде следующей обобщенной симплекс таблицы, отвечающей псевдоплану.

Таблица 1

Базисн. перем.	Своб. члены	x_1	x_2	x_3	x_4	u_1	u_2
x_2	1	0	1	0	0	1	0
x_1	4/3	1	0	0	1/3	–2/3	0
x_3	4/3	0	0	1	1/3	–8/3	0
u_2	–1/3	0	0	0	[–1/3]	–1/3	1
z	16/3	0	0	0	1/3	10/3	0

Выбирая разрешающий элемент, переходим к следующей таблице.

Таблица 2

Базисн. перем.	Своб. члены	x_1	x_2	x_3	x_4	u_1	u_2
x_2	1	0	1	0	0	1	0
x_1	1	1	0	0	0	–1	1
x_3	1	0	0	1	0	–3	1
x_4	1	0	0	0	1	1	–3
z	5	0	0	0	0	3	1

Получили псевдоплан, который является допустимым и одновременно оптимальным и целочисленным. Таким образом, получаем решение исходной задачи целочисленного программирования: $z_{\max}=5$; точка максимума: $x_1 = x_2 = 1$.

5.3. Второй алгоритм Гомори

Рассмотренный первый алгоритм Гомори применим не к любым задачам целочисленного программирования, а лишь к канонически целочисленным. Существуют методы отсечения применимые как для целочисленных (вообще говоря, не канонически целочисленных), так и для частично целочисленных задач. Самым распространенным из этих методов является второй алгоритм Гомори, который мы рассмотрим без обоснования. Как и ранее строим последовательность задач линейного программирования без условий целочисленности $З_0$, $З_1$, $З_2$, …, $З_l$, … . Каждая последующая получается из последней симплекс таблицы предыдущей задачи с помощью добавления еще одного условия (сечения Гомори второго рода). Если в последней симплекс таблице задачи $З_{l-1}$ решение не удовлетворяет условиям целочисленности, т.е. в столбце свободных членов есть дробные числа b_i^*, которые по условию должны быть целыми, то выбрав строку с максимальным $\{b_i^*\}$ ($\max\limits_i \{b_i^*\} = \{b_s^*\}$), построим дополнительное ограничение вида

$$\{b_s^*\} - \sum_k \gamma_{sk} x_k \le 0. \qquad (S_2)$$

где γ_{sk} определяются следующим образом:

1) для x_k, которые могут принимать нецелочисленные значения,

$$\gamma_{sk} = \begin{cases} a_{sk}^* & \text{при } a_{sk}^* \ge 0, \\ \dfrac{\{b_s^*\}}{1-\{b_s^*\}} \left| a_{sk}^* \right| & \text{при } a_{sk}^* < 0; \end{cases}$$

2) для x_k, которые могут принимать только целочисленные значения,

$$\gamma_{sk} = \begin{cases} \{a_{sk}^*\} & \text{при } \{a_{sk}^*\} \le \{b_s^*\}, \\ \dfrac{\{b_s^*\}}{1-\{b_s^*\}} \left| 1-\{a_{sk}^*\} \right| & \text{при } \{a_{sk}^*\} > \{b_s^*\}. \end{cases}$$

После добавления ограничения (S_2) к системе ограничений задачи $З_{l-1}$ получаем задачу $З_l$, которую снова решаем симплекс методом. Итерации повторяются до тех пор, пока решение очередной задачи не будет удовлетворять условиям целочисленности или не обнаружится, что очередная задача не имеет решения. Как и в случае первого алгоритма Гомори, можно показать, что при применении сечения Гомори второго рода задача решается за конечное число итераций.

Пример 5.4. Пусть требуется найти решение следующей задачи

$$z = x_1 + 4x_2 \to \max,$$

$$\begin{cases} 2x_1 + x_2 \le \dfrac{19}{3}, \\ x_1 + 3x_2 \le 4; \end{cases}$$

$$x_i \ge 0, \quad x_i - \text{целые числа} \ (i = 1{,}2),$$

которая не является канонически целочисленной, т. к. после уравнивания неравенств получаем систему ограничений

$$\begin{cases} 2x_1 + x_2 + x_3 = \dfrac{19}{3}, \\ x_1 + 3x_2 + x_4 = 4, \end{cases}$$

где переменная x_3 может принимать нецелочисленные значения.

Отбрасывая условия целочисленности, получаем следующую симплекс таблицу

Таблица 1

Базисн. перем.	Своб. члены	x_1	$\downarrow x_2$	x_3	x_4
x_3	19/3	2	1	1	0
$\leftarrow x_4$	4	1	3	0	1
z	0	–1	–4	0	0

Таблица 2

Базисн. перем.	Своб. члены	x_1	x_2	x_3	x_4
x_3	5	5/3	0	1	–1/3
x_2	4/3	1/3	1	0	1/3
z	16/3	1/3	0	0	4/3

Получили последнюю симплекс таблицу задачи $З_0$. Переменная x_2 должна быть целочисленной, но в точке максимума задачи $З_0$ принимает дробное значение. Построим сечение (S_2)

$$1/3 \le (1/3)\, x_1 + (1/3)\, x_4 \text{ или } -(1/3)\, x_1 - (1/3)\, x_4 + u_1 = -1/3.$$

Добавим в последнюю симплекс таблицу строку, отвечающую построенному сечению. Получим таблицу, отвечающую псевдоплану.

Таблица 1

Базисн. перем.	Своб. члены	x_1	x_2	x_3	x_4	u_1
x_3	5	5/3	0	1	–1/3	0
x_2	4/3	1/3	1	0	1/3	0
u_1	–1/3	[–1/3]	0	0	–1/3	1
z	16/3	1/3	0	0	4/3	0

Выбираем разрешающий элемент и переходим к новой таблице.

Таблица 2

Базисн. перем.	Своб. члены	x_1	x_2	x_3	x_4	u_1
x_3	10/3	0	0	1	–2	5
x_2	1	0	1	0	0	1
x_1	1	1	0	0	1	–3
z	5	0	0	0	1	1

Эта таблица является и последней для задачи З$_1$. Она дает решение и исходной задачи целочисленного программирования, т. к. в этой задаче допускается дробное значение переменной x_3. Таким образом, для исходной задачи точка максимума $x_1 = x_2 = 1$, $z_{\max} = 5$.

5.4. Метод ветвей и границ

Этот метод часто называют методом последовательного анализа вариантов. В этом методе также решается ряд задач без условия целочисленности, но эти задачи не образуют единой последовательности.

Рассмотрим задачу вида (P). Начнем опять с задачи З$_0$, которая получается из исходной отбрасыванием условий целочисленности. Пусть переменная x_k, которая должна принимать целочисленное значение, принимает в точке максимума задачи З$_0$ дробное значение x_{k_0}. Нетрудно найти такое целое n, что выполнены неравенства $n < x_{k_0} < n+1$. Рассмотрим две задачи З$_1$ и З$_2$, полученные из З$_0$ добавлением условий

$$1)\ x_k \geq n + 1,\ \text{и}\ \ 2)\ x_k \leq n$$

соответственно. Для каждой из этих задач найдем решение. Если решение одной из задач будет удовлетворять условиям целочисленности задачи (P), то эту задачу будем называть прозондированной. Одна из задач может оказаться и не имеющей решения. В этом случае задачу тоже будем называть прозондированной. Если же задача имеет решение, не удовлетворяющее условиям целочисленности, то ее «разветвляют», как и раньше, на две задачи по какой-нибудь переменной, которая должна быть целочисленной, но принимает в полученном решении дробное значение. Прозондированные задачи далее уже не ветвятся. По ходу решения задач определяются оптимальные значения целевой функции. Все сказанное можно изобразить в виде следующей схемы

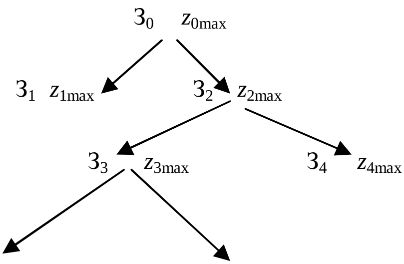


Рис. 5.1. Схема метода ветвей и границ

Вычисления заканчиваются после того, когда все очередные задачи окажутся прозондированными. Если решается задача на максимум, то среди всех разрешимых прозондированных задач выбирается та, у которой самое большое значение $z_{\max}$. Эта задача и дает точку максимума исходной задачи.

Метод ветвей и границ является универсальным методом, применимым как к полностью целочисленным задачам, так и к частично целочисленным, и даже к произвольным задачам дискретного программирования. Коммерческие задачи обычно решаются этим методом. В некоторых случаях, однако, этот метод трудно реализовать из-за большого объема машинного счета.

Следует отметить, что большинство типов задач дискретного программирования имеют экспоненциальную сложность (см. п. 1.7). Исключение составляют задача о назначениях, задача о ранце с булевыми переменными и некоторые другие, имеющие полиномиальную сложность. Таким образом, при больших размерностях большинство задач дискретного программирования за разумное время решить нельзя.

Контрольные вопросы и задачи для самостоятельного решения

1. Какие задачи называют задачами линейного целочисленного (частично целочисленного, дискретного, частично дискретного) программирования?

2. Сформулируйте задачу о назначениях. В чем заключается связь между задачей о назначениях и транспортной задачей?

3. Сформулируйте задачу о ранце.

4. В чем заключается основная идея методов отсечений? Опишите первый алгоритм Гомори для полностью целочисленных задач.

5. Как строится сечение Гомори второго рода?

6. Какова роль двойственного симплекс метода (метода последовательного уточнения оценок) при применении сечений Гомори первого и второго рода?

7. В чем заключается метод ветвей и границ?

8. Что можно сказать о сложности задач дискретного программирования?

Решить задачи о назначениях 5.1, 5.2.

5.1. Имеется пять механизмов и пять видов работ, которые этими механизмами могут выполняться. Матрица эффективностей имеет вид

$$\begin{pmatrix} 2 & 3 & 3 & 5 & 4 \\ 4 & 2 & 4 & 6 & 2 \\ 2 & 2 & 2 & 4 & 3 \\ 4 & 3 & 4 & 3 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{pmatrix}.$$

Найти оптимальное распределение механизмов по видам работ.

Ответ: Первый механизм назначается на пятую работу, второй — на четвертую, третий — на первую, четвертый — на третью, пятый — на вторую.

5.2. Для выполнения четырех видов землеройных работ используются четыре экскаватора четырех различных типов. Производительность экскаватора i-ого типа при выполнении j-ой работы задается матрицей

$$\begin{pmatrix} 0{,}9 & 0{,}6 & 0{,}7 & 0{,}9 \\ 0{,}7 & 0{,}8 & 0{,}9 & 0{,}8 \\ 0{,}8 & 0{,}6 & 0{,}8 & 0{,}9 \\ 0{,}8 & 0{,}7 & 0{,}9 & 0{,}7 \end{pmatrix}.$$

Найти такое распределение экскаваторов по видам работ, которое обеспечивает максимальную производительность.

Ответ: Экскаватор первого типа следует назначить на первый вид работ, второго типа — на второй, третьего типа — на четвертый и четвертого типа — на третий вид работ.

С помощью графического истолкования найти решения задач целочисленного программирования 5.3 — 5.5.

5.3.

$$z = x_1 + x_2 \to \max,$$

$$\begin{cases} x_1 + x_2 \le 7, \\ 4\,x_1 - 5x_2 \le 5, \\ 2\,x_1 + 11x_2 \le 38; \end{cases}$$

$$x_1 \ge 0,\ x_2 \ge 0, \quad x_1, x_2 \text{ — целые.}$$

Ответ: z_{max}=5; точек максимума две: M(3,2) и N(2,3).

5.4.

$$z = 3x_1 + x_2 \to \min;$$

$$\begin{cases} -4x_1 + x_2 \le 29, \\ 3x_1 - x_2 \le 15, \\ 5x_1 + 2x_2 \ge 38, \end{cases}$$

$$x_1, x_2 \ge 0,\ x_1, x_2 - \textit{целые.}$$

Ответ: z_{min}=19; точка минимума (0;19).

5.5.

$$z = 5x_1 + 7x_2 \to \min;$$

$$\begin{cases} -3x_1 + 14x_2 \le 78, \\ 5x_1 - 6x_2 \le 26, \\ x_1 + 4x_2 \ge 25, \end{cases}$$

$$x_1, x_2 \ge 0,\ x_1, x_2 - \textit{целые.}$$

Ответ: z_{min}=52; точка минимума (2; 6).

Пользуясь алгоритмами Гомори найти решения задач 5.6 — 5.10.

5.6.

$$z = 2x_1 + x_2 \to \max;$$

$$\begin{cases} 6x_1 + 4x_2 + x_3 = 24, \\ 3x_1 - 3x_2 + x_4 = 9, \\ -x_1 + 3x_2 + x_5 = 3, \end{cases}$$

$$x_i \geq 0,\ x_i - \textit{целые},\ (i = \overline{1,5}).$$

Ответ: $z_{\max} = 7$; точка максимума (3; 1; 2; 3; 3).

5.7.

$$z = 3x_1 + 2x_2 \to \max;$$

$$\begin{cases} x_1 + x_2 + x_3 = 13, \\ x_1 - x_2 + x_4 = 6, \\ -3x_1 + x_2 + x_5 = 9, \end{cases}$$

$$x_i \geq 0,\ x_i - \textit{целые}\ (i = \overline{1,5}).$$

Ответ: $z_{\max}$=35; точка максимума (9; 4; 0; 1; 32).

5.8.

$$z = 7x_1 + x_2 \to \max;$$

$$\begin{cases} 9x_1 + 4x_2 + x_3 = 110, \\ 11x_1 - 3x_2 - x_4 = 24, \\ 2x_1 - 7x_2 - x_5 = 15, \end{cases}$$

$$x_i \geq 0,\ x_i - \textit{целые}\ (i = \overline{1,5}).$$

Ответ: $z_{\max}$=84; точка максимума (12; 0; 2; 108; 9).

5.9.

$$z = 60x_1 + 70x_2 + 120{,}4x_3 + 130x_4 \to \max,$$

$$\begin{cases} x_1 + x_2 + x_3 + x_4 = 16, \\ x_1 + 1{,}85x_2 + x_3 + x_4 \leq 16, \\ 4x_1 + 6{,}9x_2 \quad 10x_3 + 13x_4 \leq 100, \\ 6{,}3x_1 + 5x_2 + 4x_3 + 3x_4 \leq 100, \\ x_1 + x_2 + x_3 + x_4 \geq 10, \end{cases}$$

$$x_i \geq 0,\ \ x_i - \textit{целые}\ \ (i = 1,\ 4).$$

Ответ: $z_{\max}$=1322,4; точка максимума (10; 0; 6; 0).

5.10.

$z = 5x_1 + 2x_2 - 3x_3 + 2x_4 + 3x_5 - 3x_6 \to \max,$

$$\begin{cases} 5x_1 + 6x_2 + 4x_3 + 2x_4 - 3x_5 + 5x_6 = 11, \\ 5x_1 + 5x_2 + 7x_3 + 3x_5 + 5x_6 = 10, \\ 2x_1 + 2x_2 + 2x_3 + 3x_5 = 4, \end{cases}$$

$x_i \geq 0,\ x_i - \text{\textit{целые}}\ (i = \overline{1,6}).$

Ответ: $z_{\max}$=7; точка максимума (1; 1; 0; 0; 0; 0).

6. НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

6.1. Постановка задачи и общие понятия

1. ***Пример возникновения задачи нелинейного программирования в экономике.*** Пусть предприятие изготавливает два вида продукции P_1 и P_2. Если обозначить прибыль от реализации единицы продукции P_1 через c_1, а P_2 — через c_2, то суммарную прибыль предприятия можно записать в виде

$$z = c_1 x_1 + c_2 x_2,$$

где x_1 и x_2-плановые количества продукции P_1 и P_2.

Предприятие стремится к максимуму функции z при ограничениях, определяемых условиями производства. Ранее мы считали величины c_1 и c_2 константами, независящими от x_1 и x_2. Однако такое предположение оправдано лишь при изменении x_1 и x_2 в некоторых пределах, да и то приближенно. Величины c_1 и c_2 зависят от x_1 и x_2. Чем больше продукции попадает на рынок, тем меньше цены на эту продукцию. Предположим, что истинная прибыль от реализации единицы продукции P_1 равна $c_1 - \dfrac{x_1}{b_1}$, а продукции P_2 — $c_2 - \dfrac{x_2}{b_2}$, где c_1, b_1 и c_2, b_2 —некоторые константы. Общая прибыль определяется формулой $z = \left(c_1 - \dfrac{x_1}{b_1}\right)x_1 + \left(c_2 - \dfrac{x_2}{b_2}\right)x_2$.

Эта целевая функция уже не является линейной, т.к. в ее выражении присутствуют квадраты переменных. Задача на максимум для такой функции уже не может быть решена методами линейного программирования.

2. ***Общая формулировка задачи нелинейного программирования и связанные с ней общие понятия.*** Задачей нелинейного программирования мы будем называть задачу

$$z = f(x_1, x_2, \ldots, x_n) \to \max(\min);$$

$$\begin{cases} \varphi_1(x_1, x_2, \ldots, x_n) = 0, \\ \varphi_2(x_1, x_2, \ldots, x_n) = 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \varphi_k(x_1, x_2, \ldots, x_n) = 0, \\ \varphi_{k+1}(x_1, x_2, \ldots, x_n) \le 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \varphi_m(x_1, x_2, \ldots, x_n) \le 0. \end{cases}$$

В этой задаче часть ограничений является равенствами, а часть-неравенствами. Все входящие в формулировку функции могут быть нелинейными. В дальнейшем часто совокупность значений переменных x_1, x_2, ... ,x_n мы будем рассматривать как точку $\vec{x}$. в n-мерном евклидовом пространстве $R^n \left(\vec{x} \in R^n\right)$. Множество точек R^n, которые удовлетворяют системе ограничений задачи, обозначим через М. Сокращенно задачу можно записать так:

$$f(\vec{x}) \to \max(\min);$$

$$\vec{x} \in \mathrm{M} \subset R^n.$$

Отметим, что конкретный вид системы ограничений можно изменять, не изменяя множества М, т.е. формулировать задачу в различных формах. Каждое равенство можно заменить системой неравенств

$$\begin{cases} \varphi_i(\vec{x}) \le 0, \\ -\varphi_i(\vec{x}) \le 0. \end{cases}$$

Поэтому можно считать, что система ограничений состоит только из неравенств. С другой стороны, каждое неравенство $\varphi_i(\vec{x}) \le 0$ можно заменить эквивалентным равенством $\varphi_i(\vec{x}) + x_{n+1}^2 = 0$, вводя новую переменную x_{n+1}. Отметим, что в нелинейном программировании условие неотрицательности переменной не играет особой роли и включается (если оно есть) в систему ограничений "на общих основаниях". Количество ограничений тоже можно изменить, не меняя, по сути, задачу. Например, всегда можно систему ограничений записать в виде одного равенства. Для этого вначале преобразуем систему ограничений в систему равенств

$$\begin{cases} \varphi_1(x_1, x_2, \ldots, x_n) = 0, \\ \varphi_2(x_1, x_2, \ldots, x_n) = 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \varphi_m(x_1, x_2, \ldots, x_n) = 0. \end{cases}$$

Эта же система эквивалентна одному уравнению

$$\Phi(x_1,\ x_2,\ \ldots\ ,\ x_n) = \sum_{i=1}^{m} \varphi_i^2(x_1,\ x_2,\ \ldots\ ,\ x_n) = 0.$$

Здесь, однако, следует заметить, что такое уменьшение количества ограничений часто не упрощает, а усложняет решение задачи.

Определение 6.1. *Локальным минимумом задачи называется такая точка* $\vec{x}^{(0)}$, *удовлетворяющая системе ограничений* ($\vec{x}^{(0)} \in \mathrm{M}$), *для которой выполнено неравенство* $f\left(\vec{x}^{(0)}\right) \le f\left(\vec{x}\right)$. *Здесь* $\vec{x}$ — *произвольная точка множества* $\mathrm{M} \cap K_{\varepsilon}(\vec{x}^{(0)})$, $K_{\varepsilon}(\vec{x}^{(0)})$ — ε-*окрестность точки* $\vec{x}^{(0)}$.

Под ε-окрестностью точки $\vec{x}^{(0)} = \left\{x_1{}^0, x_2{}^0, \ldots, x_n{}^0\right\}$ понимается множество точек $\vec{x} = \left\{x_1, x_2, \ldots, x_n\right\}$, для которых

$$\sqrt{\left(x_1 - x_1^0\right)^2 + \left(x_2 - x_2^0\right)^2 + \cdots + \left(x_n - x_n^0\right)^2} < \varepsilon.$$

Другими словами, в точке локального минимума значение целевой функции не больше, чем во всех точках множества М, достаточно близких к точке $\vec{x}^{(0)}$.

Аналогично определяется понятие локального максимума, т.е. точки $\vec{x}^{(0)}$, для которой выполняется неравенство

$$f(\vec{x}^0) \ge f(\vec{x}) \text{ при } \vec{x} \in \mathrm{M} \cap K_{\varepsilon}(\vec{x}^{(0)}).$$

Определение 6.2. *Глобальным минимумом* (*максимумом*) *задачи называется точка* $\vec{x}^{(0)} \in \mathrm{M}$, *для которой* $f\left(\vec{x}^{(0)}\right) \le f\left(\vec{x}\right)\left(f\left(\vec{x}^{(0)}\right) \ge f\left(\vec{x}\right)\right)$ *для любой точки* $\vec{x} \in \mathrm{M}$.

Решением задачи нелинейного программирования является точка глобального минимума или максимума (экстремума). В настоящее время регулярных общих методов нахождения глобальных экстремумов не существует. Правда, существуют методы случайного и эволюционного поиска (генетические алгоритмы), которые дают, вообще говоря, приемлемое, а не оптимальное решение. Здесь мы рассматриваем методы нахождения точек локального экстремума. Эти методы особенно важны для так называемых *одноэкстремальных задач*.

Определение 6.3. *Задача на минимум нелинейного программирования называется одноэкстремальной, если каждый ее локальный минимум одновременно является и глобальным. Аналогично понимается одноэкстремальность задачи на максимум.*

Всякая задача линейного программирования является одноэкстремальной.

6.2. Метод множителей Лагранжа

1. ***Задача на условный экстремум. Необходимые условия экстремума.*** Рассмотрим задачу нелинейного программирования с ограничениями равенствами:

$$z = f(x_1, x_2, \dots, x_n) \to \min(\max);$$

$$\begin{cases} \varphi_1(x_1, x_2, \dots, x_n) = 0, \\ \varphi_2(x_1, x_2, \dots, x_n) = 0, \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ \varphi_m(x_1, x_2, \dots, x_n) = 0. \end{cases}$$

Будем рассматривать в каждой точке $\vec{x} \in \mathrm{M}$ m+1 вектор градиентов функций $f(x)$, $\varphi_1(\vec{x})$, $\varphi_2(\vec{x})$, …, $\varphi_m(\vec{x})$. Обозначим

$$\nabla f = grad\ f, \nabla\varphi_1 = grad\,\varphi_1, \dots, \nabla\varphi_m = grad\,\varphi_m.$$

Справедливо следующее необходимое условие того, что точка $\vec{x}^{(0)} = \{x_1^0, x_2^0, \dots, x_n^0\}$ является точкой локального экстремума задачи.

Теорема 6.1. *Если точка* $\vec{x}^{(0)} \in \mathrm{M}$ *является точкой локального экстремума, то в этой точке система из* $m+1$ *векторов градиентов* $\nabla f, \nabla\varphi_1, \dots, \nabla\varphi_m$ *является линейно зависимой системой, то есть найдутся такие* $m+1$ *чисел,* $\lambda_0, \lambda_1, \dots, \lambda_m$, *не все равные нулю, для которых в точке* $\vec{x}^{(0)}$ *выполнено равенство*

$$\lambda_0 \nabla f(\vec{x}^{(0)}) + \sum_{i=1}^{m} \lambda_i \nabla\varphi_i(\vec{x}^{(0)}) = 0.$$

Для большинства задач в каждой точке множества M система векторов $\nabla\varphi_1, \nabla\varphi_2, \dots, \nabla\varphi_m$ линейно независима. Последнее условие назовем "*условие A*".

Функцией Лагранжа задачи называется функция

$$L(\vec{x}, \vec{\lambda}) = f(\vec{x}) + \sum_{i=1}^{m} \lambda_i \varphi_i(\vec{x}),$$

которая является функцией $m + n$ переменных

$$\lambda_1, \lambda_2, \dots, \lambda_m, x_1, x_2, \dots, x_n.$$

Теорема 6.2. *Пусть выполнено условие A. Для того чтобы точка* $\vec{x}^{(0)}$ *была точкой локального экстремума, необходимо, чтобы существовали такие числа* $\{\lambda_1^0, \lambda_2^0, \dots, \lambda_m^0\} = \vec{\lambda}^{(0)}$ *при которых все* *частные производные функции Лагранжа по всем ее* $n + m$ *переменным равнялись бы нулю*:

$$\frac{\partial L}{\partial x_i}\left(\vec{x}^0, \vec{\lambda}^0\right) = 0 \left(i = 1, 2, \ldots, n\right); \quad \frac{\partial L}{\partial \lambda_j}\left(\vec{x}^0, \cdots \vec{\lambda}^0\right) = 0 \ \left(j = 1, 2, \ldots, m\right).$$

Эта теорема дает необходимые условия локального экстремума. Если систему равенств теоремы 2 рассматривать, как систему уравнений относительно неизвестных $x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2, \ldots, \lambda_m$, то мы можем выделить точки, подозреваемые на экстремум.

Пример 6.1. Исходная задача:

$$z = 2x_1^2 + 3x_2^2 + x_3^2 \to \max;$$

$$\begin{cases} x_1 - x_2 + x_3 + 5 = 0, \\ x_1 + 4x_2 - x_3 - 1 = 0. \end{cases}$$

Составим функцию Лагранжа:

$$L\left(x_1, x_2, x_3, \lambda_1, \lambda_2\right) = 2x_1^2 + 3x_2^2 + x_3^2 + \\ + \lambda_1\left(x_1 - x_2 + x_3 + 5\right) + \lambda_2\left(x_1 + 4x_2 - x_3 - 1\right).$$

Необходимые условия имеют вид

$$\begin{cases} \dfrac{\partial L}{\partial x_1} = 4x_1 + \lambda_1 + \lambda_2 = 0, \\ \dfrac{\partial L}{\partial x_2} = 6x_2 - \lambda_1 + 4\lambda_2 = 0, \\ \dfrac{\partial L}{\partial x_3} = 2x_3 + \lambda_1 - \lambda_2 = 0, \\ \dfrac{\partial L}{\partial \lambda_1} = x_1 - x_2 + x_3 + 5 = 0, \\ \dfrac{\partial L}{\partial \lambda_2} = x_1 + 4x_2 - x_3 - 1 = 0. \end{cases}$$

Исключая λ_1 и λ_2 из первых трех уравнений, получим

$$\begin{cases} 6x_1 - 9x_2 - 2x_3 = 0, \\ x_1 - x_2 + x_3 + 5 = 0, \\ x_1 + 4x_2 - x_3 - 1 = 0. \end{cases}$$

Решим систему методом Гаусса:

$$\left(\begin{array}{ccc|c} 6 & -9 & -2 & 0 \\ 1 & -1 & 1 & -5 \\ 1 & 4 & -1 & 1 \end{array}\right) \Leftrightarrow \left(\begin{array}{ccc|c} 0 & -9 & -8 & 30 \\ 1 & -1 & 1 & -5 \\ 0 & 5 & -2 & 6 \end{array}\right) \Leftrightarrow \left(\begin{array}{ccc|c} 0 & 1 & \frac{8}{3} & -10 \\ 1 & -1 & 1 & -5 \\ 0 & 5 & -2 & 6 \end{array}\right) \Leftrightarrow$$

$$\Leftrightarrow \left(\begin{array}{ccc|c} 0 & 1 & \frac{8}{3} & -10 \\ 1 & 0 & \frac{11}{3} & -15 \\ 0 & 0 & -\frac{46}{3} & -44 \end{array}\right) \Leftrightarrow \left(\begin{array}{ccc|c} 0 & 1 & \frac{8}{3} & -10 \\ 1 & 0 & \frac{11}{3} & -15 \\ 0 & 0 & 1 & \frac{66}{23} \end{array}\right) \Leftrightarrow \left(\begin{array}{ccc|c} 0 & 1 & 0 & -\frac{406}{23} \\ 1 & 0 & 0 & -\frac{587}{23} \\ 0 & 0 & 1 & \frac{66}{23} \end{array}\right).$$

Координаты точки, подозреваемой на экстремум:

$$x_1 = -\frac{587}{23};\ x_2 = -\frac{406}{23};\ x_3 = \frac{66}{23}.$$

2. ***Достаточные условия условного экстремума.*** Пусть в некоторой задаче необходимые условия дают точку ($x_1^0, x_2^0, \ldots, x_n^0, \lambda_1^0, \lambda_2^0, \ldots, \lambda_m^0$). Построим матрицу вторых частных производных функции Лагранжа по переменным $x_1, x_2, \ldots, x_n$,

$$H = \begin{pmatrix} \frac{\partial^2 L}{\partial x_1^2} & \frac{\partial^2 L}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 L}{\partial x_1 \partial x_n} \\ \frac{\partial^2 L}{\partial x_2 \partial x_1} & \frac{\partial^2 L}{\partial x_2^2} & \cdots & \frac{\partial^2 L}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 L}{\partial x_n \partial x_1} & \frac{\partial^2 L}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 L}{\partial x_n^2} \end{pmatrix},$$

которая называется матрицей Гесса функции Лагранжа. В точке ($x_1^0, x_2^0, \ldots, x_n^0, \lambda_1^0, \lambda_2^0, \ldots, \lambda_m^0$) матрица Гесса — это обычная числовая матрица размера $n \times n$. Перейдем из стационарной точки M($x_1^0, x_2^0, \ldots, x_n^0, \lambda_1^0, \lambda_2^0, \ldots, \lambda_m^0$) в точку M_1 с координатами $x_1^0 + \Delta x_1,\ x_2^0 + \Delta x_2,\ \ldots,\ x_n^0 + \Delta x_n,\ \lambda_1^0,\ \lambda_2^0,\ \ldots,\ \lambda_m^0$. По формуле Тейлора приращение функции Лагранжа

$$\Delta L = L(M_1) - L(M) = \frac{1}{2}\sum_{i,j} \left.\frac{\partial^2 L}{\partial x_i \partial x_j}\right|_M \Delta x_i \Delta x_j + \mathrm{o}(|\Delta \vec{x}|^2) = \frac{1}{2} d^2 L + \mathrm{o}(|\Delta \vec{x}|^2)$$

имеет главную часть, которая является квадратичной формой с матрицей, совпадающей с матрицей Гесса функции Лагранжа H. Если Δx_i (i=1,2,…,n) выбираются так, что точка M_1 тоже удовлетворяет системе условий задачи, то $\Delta L=\Delta f$ и знак величины Δf (при достаточно малых Δx_i) определяется знаком этой же квадратичной формы. Таким образом, справедлива следующая

Теорема 6.3. *Если в исследуемой точке все собственные числа матрицы Гесса функции Лагранжа положительны, то в точке* $\{x_1^{\,0}, x_2^{\,0}, \ldots, x_n^{\,0}\}$ *находится локальный минимум. Если же эти собственные числа отрицательны, то* $\vec{x}^{(0)}=\left\{x_1^0, x_2^0, \ldots, x_n^0\right\}$ *является точкой локального максимума.*

Продолжим рассмотрение примера 1. Матрица Гесса здесь легко вычисляется, во всех точках:

$$H=\begin{pmatrix} 4 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 2 \end{pmatrix}.$$

Поскольку собственные числа диагональной матрицы равны диагональным элементам, эти собственные числа положительны, т.е. в точке $\left\{-\frac{587}{23};-\frac{406}{23};\frac{66}{23}\right\}$ локальный минимум. Можно показать, что она является и точкой глобального минимума.

Приведенные достаточные условия являются довольно грубыми. Их можно уточнить следующим образом. При смещении из точки области допустимых значений в другую ее точку приращения Δx_i не являются независимыми. Взаимосвязь между приращениями Δx_i задается системой уравнений

$$\begin{cases} \frac{\partial \varphi_1}{\partial x_1}\Delta x_1 + \frac{\partial \varphi_1}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial \varphi_1}{\partial x_n}\Delta x_n = 0, \\ \frac{\partial \varphi_2}{\partial x_1}\Delta x_1 + \frac{\partial \varphi_2}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial \varphi_2}{\partial x_n}\Delta x_n = 0, \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ \frac{\partial \varphi_m}{\partial x_1}\Delta x_1 + \frac{\partial \varphi_m}{\partial x_2}\Delta x_2 + \ldots + \frac{\partial \varphi_m}{\partial x_n}\Delta x_n = 0. \end{cases} \qquad (*)$$

Поскольку условие A считается выполненным, ранг матрицы этой системы равен m (обычно $m<n$). Поэтому из переменных Δx_i ($i=1,2,\ldots,n$) можно выбрать m базисных, которые выражаются через $n-m$ свободных переменных. Если в выражении для квадратичной формы в формуле для ΔL базисные переменные выразить через свободные

переменные, мы получим новую квадратичную форму с матрицей H' порядка $n–m$.

Теорема 6.4. *Если в исследуемой точке все собственные числа матрицы* H' *положительны, то в точке* $\{x_1^0, x_2^0, \ldots, x_n^0\}$ *находится локальный минимум. Если же эти собственные числа отрицательны, то* $\{x_1^0, x_2^0, \ldots, x_n^0\}$ *является точкой локального максимума. Если у матрицы* H' *имеется хотя бы два собственных числа разных знаков, то точка* $\{x_1^0, x_2^0, \ldots, x_n^0\}$ *не является ни точкой минимума, ни точкой максимума.*

Последняя теорема не охватывает случая, когда у матрицы H' нет собственных чисел разных знаков, но есть равные нулю собственные числа. В этом случае нужны дополнительные исследования с помощью дифференциалов высших порядков.

Пример 6.2. Рассмотрим простейшую задачу

$$f=x_1^2-x_2^2\to\min,$$
$$3x_1+4x_2-7=0.$$

Построим функцию Лагранжа

$$L(x_1, x_2, \lambda)=x_1^2-x_2^2+\lambda(3x_1+4x_2-7).$$

Применим необходимые условия условного экстремума.

$$\begin{cases} \dfrac{\partial L}{\partial x_1} = 2x_1 + 3\lambda = 0, \\ \dfrac{\partial L}{\partial x_2} = -2x_2 + 4\lambda = 0, \\ \dfrac{\partial L}{\partial \lambda} = 3x_1 + 4x_2 - 7 = 0. \end{cases}$$

Получим стационарную точку $M(–3, 4, 2)$. Матрица Гесса функции Лагранжа в этой точке

$$H = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$$

не позволяет применить теорему 3. Переменные Δx_1, Δx_2 удовлетворяют системе (*), состоящей из одного уравнения $3\Delta x_1+4\Delta x_2=0$, или $\Delta x_2=-3/4\ \Delta x_1$. Квадратичная форма $(1/2)d^2L=\Delta x_1^2-\Delta x_2^2=\Delta x_1^2-(-3/4\ \Delta x_1)^2=7/16\Delta x_1^2$. Матрица H' имеет порядок, равный единице. H'=(7/16), поэтому по теореме 4 заключаем, что точка (–3, 4) является точкой минимума нашей задачи.

3. ***Критерий Сильвестра.*** Вычисление собственных чисел матрицы является довольно громоздкой задачей. Однако для применения теорем 3 и 4 сами эти числа не нужны. Матрица квадратичной формы является симметрической и все ее собственные числа вещественны. Для определения знаков собственных чисел симметрической матрицы можно использовать теорему, принадлежащую американскому математику Сильвестру. Рассмотрим симметрическую матрицу

$$A=\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}.$$

Назовем угловыми минорами этой матрицы следующие определители

$$\Delta_1 = a_{11}, \quad \Delta_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \dots, \Delta_n = \det A.$$

Теорема 6.5. 1^0. *Для положительности всех собственных чисел симметрической матрицы необходимо и достаточно, чтобы все угловые миноры этой матрицы были положительны.*

2^0. *Для отрицательности собственных чисел необходимо и достаточно, чтобы все угловые миноры матрицы были отличны от нуля, их знаки чередовались, и выполнялось условие* $\Delta_1 < 0$.

Из этой теоремы и теоремы 6.4 следует, что из положительности угловых миноров матрицы H' вытекает, что в исследуемой стационарной точке локальный минимум, а из чередования знаков этих миноров при $\Delta_1 < 0$ вытекает, что в стационарной точке функции Лагранжа локальный максимум. Если учесть, что *произведение всех собственных чисел симметрической матрицы равняется ее определителю,* получаем, что при $\Delta_{n-m} \neq 0$ отсутствие положительности угловых миноров матрицы H' или указанного выше их знакочередования означает, что в исследуемой точке нет локального экстремума. Случай $\Delta_{n-m} = 0$ следует рассматривать особо. В этом случае возможно существование пары собственных чисел разных знаков, что гарантирует отсутствие локального экстремума. Возможно также, что нет собственных чисел разных знаков. Поскольку при этом есть нулевые собственные числа, то для установления характера стационарной точки в этом случае следует привлекать дифференциалы функции Лагранжа более высокого порядка.

Пример 6.3. Найти точки условного экстремума функции $z = x_1 x_2^2 x_3^3$ при условии $x_1+x_2+x_3=12$, лежащие в области $\{\, \vec{x} \in R^3 : x_1, x_2, x_3 > 0 \,\}$, и исследовать характер этих точек.

Построим функцию Лагранжа

$$L(\vec{x}, \lambda) = x_1 x_2^2 x_3^3 + \lambda(x_1 + x_2 + x_3 - 12)$$

и найдем ее частные производные

$$\frac{\partial L}{\partial x_1} = x_2^2 x_3^3 + \lambda; \quad \frac{\partial L}{\partial x_2} = 2x_1 x_2 x_3^3 + \lambda; \quad \frac{\partial L}{\partial x_3} = 3x_1 x_2^2 x_3^2 + \lambda;$$

$$\frac{\partial L}{\partial \lambda} = x_1 + x_2 + x_3 - 12.$$

Приравнивая эти частные производные нулю, получаем стационарную точку $M_0(2,\ 4,\ 6)$; $\lambda=-3456$. Найдем матрицу Гесса в этой точке.

$$H\big|_{M_0} = \left.\begin{pmatrix} 0 & 2x_2 x_3^3 & 3x_2^2 x_3^2 \\ 2x_2 x_3^3 & 2x_1 x_3^3 & 6x_1 x_2 x_3^2 \\ 3x_2^2 x_3^2 & 6x_1 x_2 x_3^2 & 6x_1 x_2^2 x_3 \end{pmatrix}\right|_{M_0} = 576\begin{pmatrix} 0 & 3 & 3 \\ 3 & 1 & 3 \\ 3 & 3 & 2 \end{pmatrix}.$$

Теорема 6.3 здесь ничего не дает. Второй дифференциал является квадратичной формой

$$576(\Delta x_2^2 + 2\Delta x_3^2 + 6\Delta x_1 \Delta x_2 + 6\Delta x_1 \Delta x_3 + 6\Delta x_2 \Delta x_3).$$

Приращения Δx_i в силу соотношений (*) удовлетворяют равенству $\Delta x_1 = -\Delta x_2 - \Delta x_3$. Поэтому квадратичная форма преобразуется к виду

$$576(-5\Delta x_2^2 - 4\Delta x_3^2 - 6\Delta x_2 \Delta x_3).$$

Матрица этой квадратичной формы имеет вид

$$H' = 576\begin{pmatrix} -5 & -3 \\ -3 & -4 \end{pmatrix}.$$

Очевидно, что здесь $\Delta_1<0$, $\Delta_2>0$. Таким образом, в рассматриваемой области находится только одна точка экстремума, которая является точкой максимума.

6.3. Задачи выпуклого программирования

Определение 6.4. *Множество* $M \subset R^n$ *называется выпуклым, если вместе с любыми двумя его точками* $\vec{x}_1$ *и* $\vec{x}_2$ *множеству* M *принадлежит всякий вектор, имеющий вид* $(1-\alpha)\vec{x}_1+\alpha\vec{x}_2, (0\le\alpha\le 1)$.

При n = 2 или 3 векторы вида $(1-\alpha)\vec{x}_1+\alpha\vec{x}_2, (0\le\alpha\le 1)$ лежат на отрезке, соединяющем точки $\vec{x}_1$ и $\vec{x}_2$.

Определение 6.5. *Функцию* $f(\vec{x})$ *будем называть выпуклой, если она определена на выпуклом множестве* $\Omega\subseteq R^n$ *и для любых* $\vec{x}_1, \vec{x}_2 \in \Omega$, $\alpha\in(0,1)$ *выполнено неравенство*

$$f((1-\alpha)\vec{x}_1+\alpha\vec{x}_2)\le(1-\alpha)f(\vec{x}_1)+\alpha f(\vec{x}_2).$$

Нетрудно видеть, что для выпуклой функции, определенной на всем R^n множество точек, удовлетворяющих неравенству $f(\vec{x})\le C$ с любой константой C, либо выпукло, либо пусто. То же самое можно сказать и относительно системы таких неравенств.

Всякую функцию вида $g(\vec{x})=-f(\vec{x})$, где $f(\vec{x})$ — выпуклая функция, обычно называют *вогнутой функцией*. Каждая линейная функция, очевидно, выпукла (и одновременно вогнута).

Задача выпуклого программирования формулируется следующим образом:

$$z = f(x_1,x_2,\ldots,x_n)\to \min;$$

$$\begin{cases}\varphi_1(x_1,x_2,\ldots,x_n)\le 0,\\ \varphi_2(x_1,x_2,\ldots,x_n)\le 0,\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ \varphi_m(x_1,x_2,\ldots,x_n)\le 0,\end{cases}$$

$$\vec{x}=\{x_1,x_2,\ldots,x_n\}\in\Omega.$$

где функции $f(\vec{x}), \varphi_1(\vec{x}), \ldots, \varphi_m(\vec{x})$ являются выпуклыми, а Ω — выпуклое множество, которое может в конкретном случае совпадать с R^n.

Задача линейного программирования является частным случаем задачи выпуклого программирования. Нетрудно показать, что *всякая задача выпуклого программирования одноэкстремальна*. Действительно, если $\vec{x}^0 \in M$ — точка локального минимума, то найдется такая окрестность $U(\vec{x}^0)$, что $f(\vec{x}) \geq f(\vec{x}^0)$ при всех $\vec{x} \in M \cap U(\vec{x}^0)$. Предположим, что $\vec{x}^0$ не является точкой глобального минимума. Тогда найдется такая точка $\vec{x}^1 \in M$, что $f(\vec{x}^1) < f(\vec{x}^0)$. При любом $\alpha \in (0,1)$ получим

$$f((1-\alpha)\vec{x}^0 + \alpha\vec{x}^1) \leq (1-\alpha) f(\vec{x}^0) + \alpha f(\vec{x}^1) < f(\vec{x}^0).$$

Однако при достаточно малом α выполнено включение $(1-\alpha)\vec{x}^{\,0} + \alpha\vec{x}^{\,1} \in M \cap U(\vec{x}^{\,0})$. Получаем противоречие. Следовательно, в точке $\vec{x}^{\,0}$ достигается глобальный минимум, что и требовалось доказать.

Функцией Лагранжа задачи выпуклого программирования называется функция

$$L_B\left(\vec{x}, \vec{\lambda}\right) = f\left(\vec{x}\right) + \sum_{i+1}^{m} \lambda_i \varphi_i\left(\vec{x}\right).$$

Задача выпуклого программирования называется *удовлетворяющей условию Слэйтера*, если существует хотя бы одна точка $\vec{x} \in \Omega$, в которой все неравенства системы ограничений выполняются и являются строгими неравенствами.

Введем множество $E_m = \{\vec{\lambda} \in R^m : \lambda_i \geq 0, (i = \overline{1,m})\}$.

Теорема 6.6. (*теорема Куна-Таккера*). *Пусть задача выпуклого программирования удовлетворяет условию Слэйтера. Тогда точка* $\vec{x}^{\,0} \in \Omega$ *в том и только в том случае является точкой минимума задачи, если существуют такие неотрицательные числа* $\lambda_1^0, \lambda_2^0, \ldots, \lambda_m^0 \geq 0$, $(\vec{\lambda} \in E_m)$ *что* $\left\{x_1^0, x_2^0, \ldots, x_n^0, \lambda_1^0, \lambda_2^0, \ldots, \lambda_m^0\right\} \in R^{n+m}$ *является седловой точкой функции Лагранжа задачи, то есть для любых* $\vec{x} \in \Omega$ *и* $\vec{\lambda} \in E_m$ *выполняются неравенства*

$$L_B\left(\vec{x}^{\,0}, \vec{\lambda}\right) \leq L_B\left(\vec{x}^{\,0}, \vec{\lambda}^0\right) \leq L_B\left(\vec{x}, \vec{\lambda}^0\right).$$

Таким образом, решение задачи выпуклого программирования сводится к нахождению седловых точек функции Лагранжа. Правда, задача нахождения седловых точек далеко не всегда допускает непосредственное решение. Теорема Куна-Таккера является одной из основ теории двойственности выпуклого программирования, которая выходит за пределы настоящего пособия. Приведем лишь обобщение второй теоремы двойственности, которое будет полезно в дальнейшем. Рассмотрим задачу выпуклого программирования с множеством $\Omega = E_n = \{\vec{x} \in R^n : x_i \geq 0, (i = \overline{1,n})\}$.

Теорема 6.7. *Если в задаче выпуклого программирования целевая функция* $f(\vec{x})$ *и функции* $\varphi_1(\vec{x}),\ldots,\varphi_m(\vec{x})$ *являются непрерывно дифференцируемыми, а множество* $\Omega = E_n$, *то точка* $(\vec{x}^0, \vec{\lambda}^0)$ *тогда и только тогда является седловой точкой функции Лагранжа задачи, когда в этой точке выполнены следующие соотношения*

$$\left.\frac{\partial L_B}{\partial x_j}\right|_{(\vec{x}^0,\vec{\lambda}^0)} \geq 0; \quad x_j^0 \left.\frac{\partial L_B}{\partial x_j}\right|_{(\vec{x}^0,\vec{\lambda}^0)} = 0; \tag{I}$$

$$\left.\frac{\partial L_B}{\partial \lambda_i}\right|_{(\vec{x}^0,\vec{\lambda}^0)} \leq 0; \quad \lambda_i^0 \left.\frac{\partial L_B}{\partial \lambda_i}\right|_{(\vec{x}^0,\vec{\lambda}^0)} = 0; \tag{II}$$

$$x_j^0 \geq 0; \quad j = \overline{1,n}; \quad \lambda_i^0 \geq 0; \quad i = \overline{1,m}. \tag{III}$$

В случае задачи линейного программирования в стандартной форме первые неравенства в (I) и (II) фактически совпадают с системами ограничений соответственно двойственной и заданной задач, а равенства представляют собой условия дополнительной нежесткости.

6.4. Задачи квадратичного программирования

Одним из наиболее простых и важных классов задач нелинейного программирования являются задачи квадратичного программирования, которые состоят в минимизации (или максимизации) квадратичной функции при линейных ограничениях. Поскольку задача максимизации изменением знака целевой функции сводится к задаче минимизации, мы будем считать, что общая задача квадратичного программирования в векторно-матричной форме записывается в следующем виде

$$f(\vec{x}) = (D\vec{x}, \vec{x}) + (\vec{c}, \vec{x}) \to \min,$$

$$A\vec{x} \leq \vec{b}, \ \vec{x} \geq \vec{0},$$

где $\vec{x}$ — неизвестный n-мерный вектор, D — симметричная матрица квадратичной формы размера $n\times n$, A — матрица ограничений размера $m\times n$, $\vec{C}$ — n-мерный вектор, $\vec{b}$ — m-мерный вектор.

Способы решения такой задачи определяются свойствами матрицы D. Если все собственные числа матрицы D неотрицательны, то это задача выпуклого программирования и любой локальный минимум дает ее решение. Если все собственные числа матрицы D неположительны, то мы имеем задачу вогнутого программирования, у которой может быть большое число неэквивалентных локальных минимумов, но глобальный минимум (если он существует) достигается в одной из вершин многогранного множества допустимых значений. Еще более сложный случай, когда собственные числа матрицы D имеют разные знаки. Задача здесь также является многоэкстремальной, но глобальный минимум может достигаться в точке, которая не является вершиной множества допустимых значений. Из сказанного видно, что из всех задач квадратичного программирования простейшими являются задачи *выпуклого квадратичного программирования*. Для них существует ряд способов решения. Мы коснемся одного из них, который основан на теоремах 6.6, 6.7.

Функция Лагранжа задачи выпуклого квадратичного программирования имеет вид

$$L_B\left(\vec{x},\vec{\lambda}\right)=(D\vec{x},\vec{x})+(\vec{c},\vec{x})+(\vec{\lambda},A\vec{x}-\vec{b}).$$

Если функция L_B имеет седловую точку $\left\{x_1^0, x_2^0, \ldots, x_n^0, \lambda_1^0, \lambda_2^0, \ldots, \lambda_m^0\right\}\in R^{n+m}$, то в этой точке выполняются соотношения (I), (II) и (III) теоремы 6.7. Введем уравнивающие неотрицательные переменные u_j и v_i ($j=\overline{1,n}$; $i=\overline{1,m}$), обращающие неравенства в (I), (II) в равенства. Тогда соотношения (I) — (III) запишутся в виде

$$\left.\frac{\partial L_B}{\partial x_j}\right|_{(\vec{x}^0,\vec{\lambda}^0)}-u_j=0 \quad j=(\overline{1,n}); \tag{I$_1$}$$

$$\left.\frac{\partial L_B}{\partial \lambda_i}\right|_{(\vec{x}^0,\vec{\lambda}^0)}+v_i=0 \quad i=(\overline{1,m}); \tag{II$_1$}$$

$$x_j^0 u_j=0\,(j=\overline{1,n});\quad \lambda_i^0 v_i=0\,(i=\overline{1,m}); \tag{I$_2$}$$

$$x_j^0 \geq 0, u_j \geq 0\ (j = \overline{1,n}); \quad \lambda_i^0 \geq 0, v_i \geq 0\ (i = \overline{1,m}); \qquad (III_1)$$

Таким образом, для нахождения решения задачи выпуклого квадратичного программирования нужно определить неотрицательное решение системы линейных уравнений (I_1), (II_1), удовлетворяющее условиям (I_2). Система (I_1), (II_1) имеет базисный вид, который, однако, не дает неотрицательного базисного решения, поскольку является недопустимым. Переход к допустимому базисному виду можно произвести с помощью метода искусственного базиса. Вводя искусственные переменные y_k в систему (I_1), (II_1) и решая задачу на максимум функции $z = -\sum\limits_k y_k$ с системой ограничений (I_1), (II_1), получим допустимый базисный вид системы (I_1), (II_1) и соответствующее базисное решение будет неотрицательным решением этой системы, вообще говоря, не удовлетворяющим некоторым из условий (I_2). Можно показать, что в случае разрешимости исходной задачи с помощью конечного числа операций замещения можно прийти к допустимому базисному решению системы (I_1), (II_1), удовлетворяющему всем условиям (I_2). При этом отсутствие среди допустимых базисных решений системы (I_1), (II_1) такого, которое удовлетворяет условиям (I_2), означает неразрешимость исходной задачи.

Пример 6.4. Найти минимальное значение функции

$$f = x_1^2 + 2x_2^2 - 2x_1 - 4x_2$$

При условиях

$$\begin{cases} x_1 + 2x_2 \leq 8, \\ 2x_1 - x_2 \leq 12, \end{cases}$$

$$x_1, x_2 \geq 0.$$

Функция f является выпуклой, так как собственные числа матрицы D положительны. Система ограничений состоит из линейных неравенств. Мы имеем задачу выпуклого квадратичного программирования. Составим функцию Лагранжа задачи

$$L_B = x_1^2 + 2x_2^2 - 2x_1 - 4x_2 + \lambda_1(x_1 + 2x_2 - 8) + \lambda_2(2x_1 - x_2 - 12)\ .$$

Запишем систему уравнений (I_1), (II_1), и условия (III_1), (I_2)

$$\begin{cases} 2x_1 + \lambda_1 + 2\lambda_2 - u_1 = 2, \\ 4x_2 + 2\lambda_1 - \lambda_2 - u_2 = 4, \\ x_1 + 2x_2 + v_1 = 8, \\ 2x_1 - x_2 + v_2 = 12, \end{cases}$$

$$x_1, x_2, \lambda_1, \lambda_2, u_1, u_2, v_1, v_2 \geq 0.$$

$$u_1 x_1 = u_2 x_2 = v_1 \lambda_1 = v_2 \lambda_2 = 0\,. \qquad (\mathrm{I}_3)$$

Для нахождения допустимого базисного решения построенной системы линейных уравнений применим метод искусственного базиса, т.е. решим следующую задачу линейного программирования

$$z = -y_1 - y_2 \to \max,$$

$$\begin{cases} 2x_1 + \lambda_1 + 2\lambda_2 - u_1 + y_1 = 2, \\ 4x_2 + 2\lambda_1 - \lambda_2 - u_2 + y_2 = 4, \\ x_1 + 2x_2 + v_1 = 8, \\ 2x_1 - x_2 + v_2 = 12, \end{cases}$$

$$x_1, x_2, \lambda_1, \lambda_2, u_1, u_2, v_1, v_2, y_1, y_2 \geq 0.$$

Здесь y_1, y_2 — искусственные переменные. Выразив искусственные переменные через свободные и подставив полученные выражения в целевую функцию, получим следующую симплекс-таблицу

Таблица 1

Баз. пер.	Св. чл.	x_1	$\downarrow x_2$	λ_1	λ_2	u_1	u_2	v_1	v_2	y_1	y_2
y_1	2	2	0	1	2	–1	0	0	0	1	0
$\leftarrow y_2$	4	0	4	2	–1	0	–1	0	0	0	1
v_1	8	1	2	0	0	0	0	1	0	0	0
v_2	12	2	–1	0	0	0	0	0	1	0	0
z	–6	–2	–4	–3	–1	1	1	0	0	0	0

Таблица 2

Баз. пер.	Св. чл.	$\downarrow x_1$	x_2	λ_1	λ_2	u_1	u_2	v_1	v_2	y_1	y_2
$\leftarrow y_1$	2	2	0	1	2	–1	0	0	0	1	0
x_2	1	0	1	1/2	–1/4	0	–1/4	0	0	0	1/4
v_1	6	1	0	–2	–1/2	0	1/2	1	0	0	–1/2
v_2	13	2	0	1/2	–1/4	0	–1/4	0	1	0	1/4
z	–2	–2	0	–1	–2	1	0	0	0	0	1

Таблица 3

Баз. пер.	Св. чл.	x_1	x_2	λ_1	λ_2	u_1	u_2	v_1	v_2	y_1	y_2
x_1	1	1	0	1/2	1	–1/2	0	0	0	1/2	0
x_2	1	0	1	1/2	–1/4	0	–1/4	0	0	0	1/4
v_1	5	0	0	–5/2	–3/2	1/2	1/2	1	0	–1/2	–1/2
v_2	11	0	0	–1/2	–9/4	1	–1/4	0	1	–1	1/4
z	0	0	0	0	0	0	0	0	0	1	1

Последняя симплекс таблица доставляет допустимое базисное решение системы, удовлетворяющее условиям (I_3). Поэтому точка $(\vec{x}^0, \vec{\lambda}^0)$=(1; 1; 0; 0) является седловой точкой функции Лагранжа исходной задачи, а $\vec{x}^0$=(1; 1) — оптимальным планом исходной задачи. При этом $f_{min}= -3$.

6.5. Задачи дробно-линейного программирования

Задача дробно-линейного программирования (ДЛП) может быть сформулирована в векторной форме в следующем виде

$$z = \frac{(\vec{c}, \vec{x}) + \alpha}{(\vec{d}, \vec{x}) + \beta} \to \max\ (\min),$$

где α и β — скалярные константы, постоянные векторы $\vec{c}$ и $\vec{d} \in R^n$, $\vec{x} \in R^n$ — вектор искомых переменных. Предполагается, что $\vec{x} \in$ $\mathrm{M} \subset R^n$. При этом

$$\mathrm{M}=\{\vec{x} \in R^n \colon A\vec{x} = \vec{b},\ \vec{x} \geq 0,\ \vec{b} \in R^m\}.$$

Таким образом, в качестве целевой функции используется отношение двух линейных функций, условия-ограничения задачи являются линейными равенствами и неравенствами.

Обычно предполагают также, что знаменатель целевой функции не обращается в нуль на области допустимых значений М. Не ограничивая общности, можно считать, что знаменатель положителен на множестве М.

Задачи дробно-линейного программирования решают в тех приложениях, когда оптимизируются относительные показатели, например, рентабельность, производительность и т. д.. Особенно часто такие задачи встречаются в области финансовой деятельности: планировании доходов корпораций, управлении статьями банковского баланса и т. п.

Задачи ДЛП имеют одну важную общую черту с задачами линейного программирования. Поверхности уровня целевой функции этих задач определяются линейными уравнениями, т. е. являются гиперплоскостями. Действительно, равенство $z=C$ при произвольной константе C можно записать в виде $(\vec{c}, \vec{x}) + \alpha = C((\vec{d}, \vec{x}) + \beta)$, а это уравнение задает некоторую гиперплоскость. При неопределенном C мы получаем пучок гиперплоскостей, пересекающихся по линейному многообразию размерности $n-2$, часто называемому множеством вращения. Множество вращения является множеством точек, удовлетворяющих системе уравнений

$$\begin{cases} (\vec{c}, \vec{x}) & = & -\alpha, \\ (\vec{d}, \vec{x}) & = & -\beta. \end{cases}$$

Мы будем считать, что эта система совместна, т. к. в противном случае задачу ДЛП легко свести к обычной задаче линейного программирования. В случае $n=2$ гиперповерхности уровня будут прямыми линиями, а множество вращения состоит из одной точки. Вращая линию уровня целевой функции вокруг этой точки, мы будем увеличивать или уменьшать значение целевой функции. На этом основан графический метод решения задачи ДЛП. По крайней мере, одна из угловых точек допустимого многоугольника будет оптимальной.

В общем случае обычно применяют сведение задачи ДЛП к задаче линейного программирования с помощью приема преобразования переменных. Еще раз отметим, что знаменатель целевой функции можно считать положительным на множестве M, т.к. в противном случае можно умножить числитель и знаменатель на (–1). Введем следующие переменные $y_0 = ((\vec{d}, \vec{x}) + \beta)^{-1}$ и $y_i = y_0 x_i$ при $i=1, 2, \ldots, n$. Задача ДЛП может быть записана в виде следующей задачи линейного программирования

$$z = (\vec{c}, \vec{y}) + \alpha\, y_0 \to \max(\min),$$

$$\begin{cases} A\vec{y} & -y_0 \vec{b} & = 0, \\ (\vec{d}, \vec{y}) & +\beta\, y_0 & = 1, \end{cases}$$

$$0 \le \vec{y} \in R^n, \quad 0 \le y_0 \in R^1.$$

Количество переменных в этой задаче равно $n+1$. После решения этой задачи линейного программирования решение исходной задачи легко найти по формулам $x_i = y_i / y_0$ при $i=1, 2, \ldots, n$.

Пример 6.5. Найти решение следующей задачи

$$z=\frac{2x_1+x_2}{x_1+x_2}\to \min,$$

$$\begin{cases} 2x_1 & +8x_2 & \le 26, \\ x_1 & +x_2 & \le 4, \\ 12x_1 & +3x_2 & \le 39, \end{cases}$$

$$x_1, \quad x_2 \ge 0.$$

Для нахождения решения вначале уравняем неравенства системы ограничений с помощью введения неотрицательных переменных.

$$z=\frac{2x_1+x_2}{x_1+x_2}\to \min,$$

$$\begin{cases} 2x_1 & +8x_2 & +x_3 & =26, \\ x_1 & +x_2 & +x_4 & =4, \\ 12x_1 & +3x_2 & +x_5 & =39, \end{cases}$$

$$x_1, x_2, x_3, x_4, x_5 \ge 0.$$

Введем новые переменные $y_0=(x_1+x_2)^{-1}$, $y_i=y_0x_i$ $(i=\overline{1,2})$. Получим задачу линейного программирования

$$z=2y_1+y_2\to \min,$$

$$\begin{cases} 2y_1 & +8y_2 & +y_3 & -26y_0 & =0, \\ y_1 & +y_2 & +y_4 & -4y_0 & =0, \\ 12y_1 & +3y_2 & +y_5 & -39y_0 & =0, \\ y_1 & + & y_2 & & =1, \end{cases}$$

$$y_i \ge 0 \ (i=\overline{0,5}).$$

Воспользуемся модификацией симплекс-метода, изложенном в 1.6. Для этого перейдем к задаче на максимум, введем искусственную переменную u в последнее равенство системы ограничений, составим М-задачу при М=200 и исключим искусственную переменную из целевой функции. Получим задачу

$$z_1=-200+198y_1+199y_2\to \max,$$

$$\begin{cases} 2y_1 & +8y_2 & +y_3 & -26y_0 & =0, \\ y_1 & +y_2 & +y_4 & -4y_0 & =0, \\ 12y_1 & +3y_2 & +y_5 & -39y_0 & =0, \\ y_1 & + & y_2 & +u & =1, \end{cases}$$

$$y_i \ge 0 \ (i=\overline{0,5}).$$

Составим симплекс-таблицу

Таблица 1

Баз. пер.	Св. чл.	y_1	$\downarrow y_2$	y_3	y_4	y_5	y_0	u
y_3	0	2	8	1	0	0	–26	0
$\leftarrow y_4$	0	1	1	0	1	0	–4	0
y_5	0	12	3	0	0	1	–39	0
u	1	1	1	0	0	0	0	1
z_1	–200	–198	–199	0	0	0	0	0

Таблица 2

Баз. пер.	Св. чл.	y_1	y_2	y_3	y_4	y_5	$\downarrow y_0$	u
$\leftarrow y_3$	0	–6	0	1	–8	0	6	0
y_2	0	1	1	0	1	0	–4	0
y_5	0	9	0	0	–3	1	–27	0
u	1	0	0	0	–1	0	4	1
z_1	–200	1	0	0	199	0	–796	0

Таблица 3

Баз. пер.	Св. чл.	y_1	y_2	y_3	$\downarrow y_4$	y_5	y_0	u
y_0	0	–1	0	1/6	–4/3	0	1	0
y_2	0	–3	1	2/3	–13/3	0	0	0
y_5	0	–18	0	9/2	–39	1	0	0
$\leftarrow u$	1	4	0	–2/3	13/3	0	0	1
z_1	–200	–795	0	398/3	–2587/3	0	0	0

Таблица 4

Баз. пер.	Св. чл.	y_1	y_2	y_3	y_4	y_5	y_0	u
y_0	4/13				0			
y_2	1				0			
y_5	9				0			
y_4	3/13	12/13	0	–2/13	1	0	0	3/13
z_1	–1	1	0	0	0	0	0	3/13

Таким образом, ответ в исходной задаче: $z_{min} = -z_{1max} = 1$, точка минимума $x_1 = 0$, $x_2 = 13/4$.

6.6. Численные методы решения задач нелинейного программирования

1. ***Методы безусловной оптимизации. Метод градиетного спуска и метод Ньютона-Рафсона.*** Рассматриваемые ниже методы являются способами нахождения точек локального экстремума для задач без ограничений и поэтому называются методами безусловной оптимизации. В приложениях чаще всего встречаются задачи с ограничениями, которые можно свести к задачам на безусловный экстремум с помощью метода *штрафных функций*.

Рассмотрим задачу без ограничений

$$z = f(x_1, x_2, \ldots, x_n) \to \min .$$

В любой точке $\vec{x} \in R^n$ можно построить вектор градиента

$$\nabla f = \left\{ \frac{\partial f}{\partial x_1}; \frac{\partial f}{\partial x_2}; \ldots; \frac{\partial f}{\partial x_n} \right\}.$$

Этот вектор указывает направление наискорейшего возрастания функции $f(\vec{x})$. Противоположный вектор $(-\nabla f)$ называется вектором антиградиента и указывает направление наискорейшего убывания функции $f(\vec{x})$ в данной точке.

Идея ***метода градиентного спуска*** заключается в том, что мы из произвольно взятой точки в R^n сдвигаемся по направлению антиградиента в некоторую близкую точку. При этом сдвиг выбирается таким, чтобы в новой точке значение функции было меньше, чем в старой. Затем в этой новой точке определяется антиградиент. В его направлении делается новый шаг в некоторую новую точку с уменьшением значения целевой функции $f(\vec{x})$. В результате ряда шагов мы можем сколь угодно близко подойти к точке локального минимума функции $f(\vec{x})$. Если функция $f(\vec{x})$ имеет лишь один экстремум, который является и глобальным минимумом, то мы получаем приближенное решение задачи. Если же у функции $f(\vec{x})$ несколько локальных минимумов, то мы "скатимся" в результате ряда шагов в окрестность одного из локальных минимумов. Причем в зависимости от начальной точки можно попасть в окрестность того или иного локального минимума.

Расчетные формулы метода градиентного спуска имеют следующий вид:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - h_k \nabla f_k ,$$

где $\vec{x}^{(k+1)}$ — вектор координат точки, получаемой на $k+1$ шаге вычислительного процесса; $\vec{x}^{(k)}$ — точка , полученная на к-том шаге; ∇f_k — градиент функции в точке $\vec{x}^{(k)}$; h_k — неотрицательное число, определяющее шаг в направлении антиградиента. В покоординатной записи расчетные формулы имеют вид

$$x_i^{(k+1)} = x_i^{(k)} - h_k \left(\frac{\partial f}{\partial x_i} \right)_k, \; i = 1, 2, \ldots, n.$$

При организации вычислительного процесса требуется на каждом шаге определять величину h_k и сформулировать критерий окончания счета. Поскольку в самой точке минимума все частные производные равны нулю, то в достаточно малой ее окрестности они должны быть достаточно малы. Поэтому счет заканчивается, когда длина градиента $|\nabla f|$ в очередной точке становится меньше некоторого наперед заданного числа.

Выбор величины h_k можно производить по-разному. Если функция $f(\vec{x})$ не очень сложна, то на каждом шаге h_k можно выбрать оптимальным образом. Для этого построим функцию

$$\varphi_k(h) = f\left(\vec{x}^{(k)} - h(\nabla f)_k\right).$$

Найдем минимум этой функции по переменной h. Для этого обычно решают уравнение

$$\varphi_k'(h) = 0.$$

Если это уравнение удается решить, то мы и получаем искомое число h_k. Вычислительный метод, включающий такое определение h_k, называют *методом наискорейшего градиентного спуска*. В некоторых случаях выбор оптимального шага затруднен. Иногда, решая задачу на ЭВМ, программируют эмпирический подбор шага, т.е. подбирают h_k так, чтобы выполнялось неравенство $f\left(\vec{x}^{(k+1)}\right) \le f\left(\vec{x}^{(k)}\right)$.

В ***методе Ньютона-Рафсона*** приближенно решается система уравнений

$$\frac{\partial f}{\partial x_i} = 0, \; i = 1,2,\ldots,n.$$

Так же, как и в методе градиентного спуска, этот метод можно применять в случае одноэкстремальных задач. Расчетные формулы метода Ньютона-Рафсона имеют вид:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - H_k^{-1} \cdot (\nabla f)_k,$$

где $\vec{x}^{(k)}, (\nabla f)_k, \vec{x}^{(k+1)}$ рассматриваются как матрицы-столбцы размера $n\times 1$, а H_k^{-1} — матрица, обратная по отношению к матрице Гесса для функции $f(\vec{x})$ в точке $\vec{x}^{(k)}$,

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

Вторые частные производные берутся в точке $\vec{x}^{(k)}$. Окончание счета, как и в методе градиентного спуска, производится по критерию малости $|\nabla f|$. Метод Ньютона-Рафсона сходится быстрее к точке минимума, чем метод градиентного спуска в случае задач выпуклого программирования ($f(\vec{x})$ — выпуклая функция). В частности, для квадратичной функции $f(\vec{x})$ этот метод приводит к точке минимума за один шаг. Применение метода Ньютона-Рафсона требует вычисления обратной матрицы, что при большом числе переменных является очень громоздкой задачей.

Оба описанных метода часто применяют для решения произвольных задач нелинейного программирования. Здесь, однако, отсутствуют теоретические обоснования сходимости методов и скорости их сходимости.

2. Метод штрафных функций. Рассмотрим задачу выпуклого программирования:

$$z = f(x_1, x_2, \ldots, x_n) \to \min;$$

$$\begin{cases} \varphi_1(x_1, x_2, \ldots, x_n) \le 0, \\ \varphi_2(x_1, x_2, \ldots, x_n) \le 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \varphi_m(x_1, x_2, \ldots, x_n) \le 0. \end{cases}$$

Здесь функции $f, \varphi_1, \varphi_2, \ldots, \varphi_m$ — выпуклые, а множество $\Omega = R^n$.

Метод штрафных функций заключается в составлении по данной задаче некоторой задачи на безусловный экстремум, точка минимума которой совпадает с точкой минимума исходной задачи. Построенную задачу можно решать приближенно с помощью описанных выше методов.

Вспомогательная задача на безусловный экстремум строится с помощью введения в целевую функцию дополнительных штрафных слагаемых, которые очень велики в тех точках R^n, которые не принадлежат области допустимых значений M. При минимизации функции точка минимума вынуждена находиться в области допустимых значений исходной задачи.

Рассмотрим функцию

$$\mu(y) = \begin{cases} y, & y > 0; \\ 0, & y \le 0. \end{cases}$$

Построим по нашей задаче функцию

$$\Phi_R(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n) + R\sum_{j=1}^{m} \mu\big(\varphi_j(x_1, x_2, \ldots, x_n)\big),$$

где $R > 0$ — достаточно большая константа.

Теорема 6.8. *Если* $\vec{x}^{(0)}$ *является точкой минимума задачи выпуклого программирования, то существует такое число* $R_0 > 0$, *что при* $R > R_0$ *точка* $\vec{x}^{(0)}$ *будет точкой безусловного минимума функции* $\Phi_R(\vec{x})$. *Если для всех* $R > R_0$ $\vec{x}^{(0)}$ *является точкой безусловного экстремума функции* $\Phi_R(\vec{x})$, *то эта точка является и точкой минимума задачи выпуклого программирования.*

Отметим, что функция $\Phi_R(\vec{x})$ сама является выпуклой, поэтому задача $\Phi_R(\vec{x}) \to \min$ одноэкстремальна. Для решения можно применить метод градиентного спуска. Компоненты градиента здесь являются разрывными функциями, поэтому используют обобщенный градиент, имеющий вид

$$\nabla \Phi_R = \nabla f + R \sum_{j=0}^{m} \mu(\operatorname{sign} \varphi_i) \nabla \varphi_j,$$

$$\text{где} \quad \operatorname{sign} \varphi_j = \begin{cases} 1, \text{ при } \varphi_j > 0, \\ 0, \text{ при } \varphi_j = 0, \\ -1, \text{ при } \varphi_j < 0. \end{cases}$$

Организация наискорейшего спуска встречает трудности из-за сложности функции $\Phi_R(\vec{x})$. Поэтому применяют градиентный спуск с эмпирическим выбором шага. Подбор величины R часто тоже производится эмпирически. Величина R достаточно велика, если ее увеличение не приводит к изменению точки минимума $\Phi_R(\vec{x})$. Существуют точные оценки для R, при выполнении которых точка минимума функции $\Phi_R(\vec{x})$ совпадает с решением соответствующей задачи выпуклого программирования (см. [21], стр. 335).

Иногда, чтобы избежать трудностей, связанных с разрывностью градиента, вместо функции $\Phi_R(\vec{x})$ используют функцию

$$S_R(x_1, x_2, \dots, x_n) = f(x_1, x_2, \dots, x_n) + R\sum_{j=1}^{m} \mu^2(\varphi_j(x_1, x_2, \dots, x_n)),$$

с квадратичной функцией штрафа. При гладкости функций, входящих в формулировку исходной задачи, функция $S_R(\vec{x})$ гладкая, и для нахождения ее точки минимума можно использовать любой из градиентных методов. Недостатком функции $S_R(\vec{x})$ является то, что ее точка минимума удовлетворяет системе ограничений лишь приближенно, причем тем точнее, чем больше R. При этом скорость сходимости градиентных методов при больших R медленная.
На практике метод штрафных функций применяется и при решении одноэкстремальных задач, не являющихся задачами выпуклого программирования.

6.7. Многокритериальные задачи

В рассмотренных до сих пор задачах имелся лишь один целевой параметр, который мы выражали через контролируемые параметры и получали одну целевую функцию. В экономике такие задачи возникают при рассмотрении небольших по масштабу и скромных по значению мероприятий. Когда речь идет о крупномасштабных операциях, затрагивающих разнообразные интересы, их эффективность не может быть выражена с помощью одного единственного показателя.

При организации работы промышленного предприятия естественно стремиться к максимуму валового объема продукции V, к максимуму чистого дохода D, к минимуму себестоимости продукции S, а также к максимуму производительности труда P. При более глубоком анализе может появиться еще ряд дополнительных критериев. Такая множественность показателей эффективности, из которых одни желательно обратить в максимум, а другие — в минимум, характерна для любой достаточно сложной задачи исследования операций. Такие задачи называются *многокритериальными*.

Если все целевые показатели удается выразить через контролируемые параметры x_1, x_2 , ... ,x_n, мы получаем задачу на отыскание точки $\vec{x}^{(0)} \in \mathrm{M}$, в которой достигался бы глобальный экстремум (можно считать минимум) сразу нескольких функций $f_1(\vec{x})$, $f_2(\vec{x})$, …, $f_k(\vec{x})$. В общем случае такая задача решения не имеет. При принятии решения приходится выбирать компромиссный вариант, приемлемый сразу по всем критериям. Для такого выбора, кроме самих функций $f_i(\vec{x})$ и ограничений, определяющих множество M, требуются дополнительные (часто субъективные) соображения. Объективно в множестве M можно выделить подмножество $\wp$ так называемых *паретовских решений*, из которых и производится выбор компромиссного варианта.

Определение 6.6. *Паретовским решением многокритериальной задачи на минимум называется такая точка* $\vec{x}^{(0)} \in \mathrm{M}$, , *для которой нельзя найти «лучшую» по любому данному критерию точку, «не ухудшающую» значения других критериев. Таким образом,* $\vec{x}^{(0)}$ *характеризуется следующим свойством. Если для некоторого* $\vec{x} \in \mathrm{M}$ *выполнено неравенство* $f_i\left(\vec{x}^{(0)}\right) > f_i\left(\vec{x}\ \right)$, *то найдется* j $(1 \le j \le k)$, *для которого выполняется неравенство* $f_j\left(\vec{x}^{(0)}\right) < f_j\left(\vec{x}\ \right)$.

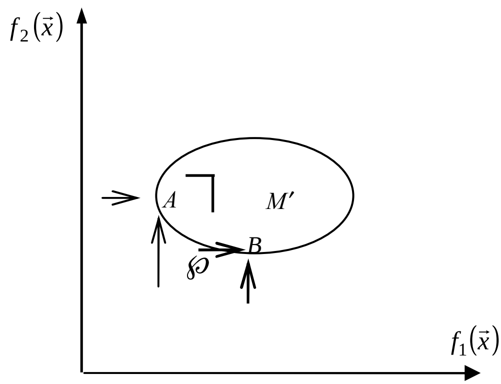


Рис. 6.1. Образ множества Парето в плоскости критериев f_1, f_2

Можно сказать, что паретовское решение $\vec{x}^{(0)}$ — это такое решение системы ограничений, для которого нет решений этой системы, доминирующих над $\vec{x}^{(0)}$ по всем критериям. Чаще всего множество $\wp$ гораздо уже множества М. Это видно на следующем примере. Пусть рассматривается задача с двумя критериями $f_1(\vec{x})$ и $f_2(\vec{x})$. Для каждого значения $\vec{x} \in M$ мы имеем точку на плоскости с координатами $(f_1(\vec{x}),\ f_2(\vec{x}))$. Множество всех таких точек обозначим M' (рис. 6.1). Паретовским решениям отвечают точки дуги AB границы множества M'.

Непосредственное вычисление множества Парето $\wp$ довольно сложно. Часто применяют *эвристические приемы решения многокритериальных задач*. Рассмотрим несколько таких приемов.

1. ***Способ агрегированного критерия.*** Рассмотрим задачу

$$\sum_{i=1}^{k} a_i f_i(\vec{x}) \to \min, \vec{x} \in \text{M},$$

где a_i — некоторые неотрицательные числа, называемые весовыми коэффициентами. Они характеризуют относительную важность критериев. Чем больше число a_i, тем большую важность придают критерию $f_i(\vec{x})$. Если все функции $f_i(\vec{x})$ выпуклые и множество М — тоже, то у этой задачи есть точка минимума, и она обязательно принадлежит множеству $\wp$. При этом в качестве точки минимума можно получить любую точку множества $\wp$ надлежащим выбором весовых коэффициентов. Выбор этих коэффициентов обычно производится эвристически, например, группой специальных экспертов.

2. ***Способ главного показателя***. Здесь эвристически выбирается некоторый главный показатель, например, $f_1(\vec{x})$. Его стремятся минимизировать при условии, что $\vec{x} \in \text{M}$; кроме того, остальные показатели остаются не больше некоторых приемлемых значений $b_2, b_3, \ldots, b_k$ $(f_i(\vec{x}) \le b_i,\ i = 2, 3, \ldots, k)$. При таком подходе все показатели, кроме главного, переводятся в разряд заданных ограничений.

3. ***Способ последовательных уступок***. Этот способ построения компромиссного решения предполагает предварительное расположение целевых функций в порядке убывающей важности: $f_1(\vec{x}),\ f_2(\vec{x}), \ldots,\ f_k(\vec{x})$. Сначала находится решение, обращающее в минимум важнейший показатель $f_1(\vec{x})$. Пусть $f_{1\min}=b_1$. Затем назначается некоторая "уступка" Δb_1, которую мы согласны сделать для того, чтобы минимизировать второй показатель $f_2(\vec{x})$. К первоначальной системе ограничений добавляется условие $f_1(\vec{x}) \le b_1 + \Delta b_1$ и с новой системой ограничений находится $\min f_2(\vec{x}) = b_2$. Затем назначается новая уступка Δb_2 и к системе ограничений первого шага добавляется условие $f_2(\vec{x}) \le b_2 + \Delta b_2$. С новой системой ограничений решается задача на минимум функции $f_3(\vec{x})$ и т.д. На последнем шаге к системе ограничений добавляется условие $f_{k-1}(\vec{x}). \le b_{k-1} + \Delta b_{k-1}$ и находится $\min f_k(\vec{x})$. Последняя точка минимума $\vec{x}^{(0)}$ считается решением задачи. Такой способ построения компромиссного решения хорош тем, что здесь видно, ценой какой "уступки" в одном показателе приобретается выигрыш в другом, и какова величина этого выигрыша. Отметим, что последний способ требует эвристического упорядочивания показателей и такого же назначения последовательных уступок.

При любом способе постановки многокритериальной задачи для получения ее конкретного решения требуется вмешательство человеческого сознания для проведения неформализуемых оценок.

Контрольные вопросы и задачи для самостоятельного решения

1. Как формулируется общая задача нелинейного программирования? Как можно менять формулировку задачи, оставляя задачу эквивалентной исходной?

2. Дайте определение локального экстремума задачи нелинейного программирования. Что такое глобальный экстремум? Какие задачи называются одноэкстремальными?

3. Когда задача нелинейного программирования называется задачей на условный экстремум? Что такое функция Лагранжа задачи на условный экстремум? В чем заключаются необходимые условия условного экстремума?

4. Сформулируйте достаточные условия условного экстремума. Что такое критерий Сильвестра и как он используется при выяснении характера стационарной точки функции Лагранжа?

5. Дайте определение выпуклого множества в R^n и выпуклой функции. Что такое задача выпуклого программирования? Является ли такая задача одноэкстремальной?

6. Что такое функция Лагранжа задачи выпуклого программирования и ее седловая точка?

7. Сформулируйте теорему Куна-Таккера.

8. Как формулируются необходимые и достаточные условия Седловой точки?

9. Что такое задача квадратичного программирования? Когда она является задачей выпуклого программирования?

10. Как решить задачу выпуклого квадратичного программирования сведением к вспомогательной задаче линейного программирования?

11. Как формулируется задача дробно-линейного программирования? Как истолковать эту задачу геометрически в случае двух переменных?

12. Как сводится задача дробно-линейного программирования к задаче линейного программирования с помощью введения новых переменных?

13. Опишите метод наискорейшего градиентного спуска для задач на безусловный экстремум. В чем состоит расчетная схема метода Ньютона-Рафсона?

14. В чем состоит основная идея метода штрафных функций? Линейная и квадратичная функции штрафа. В чем заключаются преимущества и недостатки каждой из них?

15. Что такое паретовское решение многокритериальной задачи? Основные подходы, используемые для решения многокритериальных задач. Что такое способ агрегированного критерия?

16. Опишите суть способов главного показателя и последовательных уступок.

Используя геометрическое истолкование задачи нелинейного программирования решить задачи 6.1 — 6.4.

6.1.

$$z = x_1 x_2 \to \max,$$

$$\begin{cases} 6x_1 + 4x_2 \geq 12, \\ 2x_1 + 3x_2 \leq 24, \\ -3x_1 + 4x_2 \leq 12, \\ x_1, \ x_2 \geq 0. \end{cases}$$

Ответ: z_{max}=24, точка max (6; 4).

6.2.

$$z = 9(x_1 - 5)^2 + 4(x_2 - 6)^2 \to \min,$$

$$\begin{cases} 3x_1 + 2x_2 \geq 12, \\ x_1 - x_2 \leq 6, \\ x_2 \leq 4, \\ x_1, \ x_2 \geq 0. \end{cases}$$

Ответ: z_{min}=16, точка min (5;4).

6.3.

$z = 4x_1 + 3x_2 \to \max;$

$$\begin{cases} x_1^2 - 2x_1 + x_2^2 - 2x_2 - 34 \le 0, \\ x_1 \ge 1, \\ x_2 \ge 1. \end{cases}$$

Ответ: $z_{\max}$=37, точка max (5,8;4,6).

6.4.

$z = x_1 x_2 \to \max;$

$x_1^2 + 2x_1 + x_2^2 - 2x_2 - 14 \ge 0,$

$2x_1 + x_2 \le 10,$

$x_1,\ x_2 \ge 0.$

Ответ: $z_{\max}$=12,5, точка max(2,5; 5).

В задачах 6.5 — 6.8 *найти условные экстремумы указанных функций и определить их характер.*

6.5.

$z = x_1^2 + x_2^2 + x_3$

при условиях

$$\begin{cases} x_1 + x_2 + x_3 = 4, \\ 2x_1 - 3x_2 = 12. \end{cases}$$

Ответ: $\begin{cases} z_{\min} = 16\dfrac{53}{64} \quad \textit{в точке} \\ (27/8; -7/4;\ 19/8). \end{cases}$

6.6.

$z = x_1 x_2 x_3$

при условиях

$$\begin{cases} 2x_1 x_2 + x_2 x_3 = 4, \\ 2x_1 - 3x_2 = 12. \end{cases}$$

Ответ: $\begin{cases} z_{\min} = -56/27 \ \textit{в точке} \\ (-1{,}3;\ -26/3;\ -28/39), \\ z_{\max} = 72 \ \textit{в точке} \\ (3;\ -2;\ -12). \end{cases}$

6.7.

$z = x_1 x_2 + x_2 x_3$

при условиях

$$\begin{cases} x_1 + x_2 = 4, \\ x_2 + x_3 = 4. \end{cases}$$

Ответ: $\begin{cases} z_{\max} = 8 \ \textit{в точке} \\ (2; 2; 2). \end{cases}$

6.8.

$z = 3x_1^2 + 2x_1 + 2x_2^2 + 4x_2 x_3$

при условиях

$$\begin{cases} x_1^2 + 2x_2^2 = 19, \\ x_1 + 2x_2 x_3 = 11. \end{cases}$$

Отв. $\begin{cases} z_{\min} = 43 \ \textit{в точках} \\ (-1; 3; 2) \ \textit{и} \ (-1; -3; -2). \end{cases}$

Решить задачи выпуклого квадратичного программирования 6.9 — 6.12.

6.9.

$f = 2x_1^2 + 2x_2^2 - x_1 x_2 - x_1 - 4x_2 \to \min;$

$$\begin{cases} x_1 + 2x_2 \le 12, \\ 3x_1 + x_2 \le 15, \end{cases}$$

$x_1,\quad x_2 \ge 0.$

6.10.

$f = x_1^2 + x_2^2 - x_1 - 8x_2 \to \min;$

$$\begin{cases} x_1 + x_2 \le 7, \\ x_2 \le 5, \end{cases}$$

$x_1,\quad x_2 \ge 0.$

Ответ: $f_{\min} = -\dfrac{38}{15}$ *при* $x_1 = \dfrac{8}{15}$, $x_2 = \dfrac{17}{15}$. **Ответ:** $f_{\min} = -\dfrac{65}{4}$ *при* $x_1 = \dfrac{1}{4}$, $x_2 = 4$.

6.11.

$$f = x_1^2 + x_2^2 + 2x_1 - 8x_2 \to \min;$$
$$\begin{cases} x_1 + 2x_2 \le 12, \\ -x_1 + x_2 \ge -8, \end{cases}$$
$$x_1,\ x_2 \ge 0.$$

6.12.

$$f = x_1^2 + x_2^2 + 2x_3^2 - 2x_2 - 3x_3 \to \min;$$
$$\begin{cases} x_1 + x_2 + x_3 \le 18, \\ x_2 \le 12, \\ x_1 + 2x_3 \le 14, \end{cases}$$
$$x_1,\quad x_2,\quad x_3 \ge 0.$$

Ответ: $f_{\min} = -16$ *при* $x_1 = 0$, $x_2 = 4$. **Ответ:** $f_{\min} = -\dfrac{17}{8}$ *при* $x_1 = 0$, $x_2 = 1$, $x_3 = \dfrac{3}{4}$.

Решить задачи дробно-линейного программирования 6.13 — 6.17.

6.13.

$$z = \frac{-5x_1 + 2x_2}{3x_1 + 4x_2} \to \max,$$
$$\begin{cases} x_1 + 4x_2 + x_3 = 16, \\ 2x_1 + 3x_2 - x_4 = 12, \\ 3x_1 - 2x_2 + x_5 = 18, \end{cases}$$
$$x_i \ge 0\ (i = \overline{1,5}).$$

Ответ: z_{max}=1/2 в точке (0; 4; 0; 0; 26).

6.14.

$$z = \frac{3x_1 - x_2}{x_1 + x_2} \to \max,$$
$$\begin{cases} x_1 + x_2 - x_3 = 5, \\ -x_1 + 3x_2 + x_4 = 7, \\ 3x_1 - x_2 + x_5 = 11, \end{cases}$$
$$x_i \ge 0\ (i = \overline{1,5}).$$

Ответ: z_{max}=2,2 в точке (4; 1; 0; 8; 0).

6.15.

$$z = \frac{8x_1 + 3x_2 + 2x_3 + x_4}{x_1 + x_2 + x_3 + x_4} \to \max,$$

$$\begin{cases} 2x_1 + x_2 + x_3 + 3x_4 \le 300, \\ x_1 + 2x_3 + x_4 \le 70, \\ x_1 + 2x_2 + x_3 \le 340, \end{cases}$$

$x_i \ge 0$ $(i = \overline{1, 4})$.

Ответ: z_{max}=8 в точке (70; 0; 0; 0).

6.16.

$$z = \frac{5x_1 - x_2 + 8x_3 + 10x_4 - 5x_5 + x_6}{x_1 + x_2 + x_3 + x_4 + x_5 + x_6} \to \max,$$

$$\begin{cases} 2x_1 - x_2 + 3x_4 + x_5 - x_6 = 40, \\ -x_1 + 2x_2 + x_3 + 2x_4 + 2x_6 = 20, \\ 3x_1 - x_2 + 2x_3 - x_4 + 3x_5 + x_6 = 30, \end{cases}$$

$x_i \ge 0$ $(i = \overline{1, 6})$.

Ответ: z_{max}=489/62 в точке (6,8; 0; 9,2; 8,8; 0; 0).

6.17.

$$z = \frac{2x_1 - 3x_2 + 4x_3 + 5x_4 - x_5 + 8x_6}{x_1 + x_2 + x_3 + x_4 + x_5 + x_6} \to \max,$$

$$\begin{cases} x_1 + 5x_2 - 3x_3 - 4x_4 + 2x_5 + x_6 = 120, \\ 2x_1 + 9x_2 - 5x_3 - 7x_4 + 4x_5 + 2x_6 = 320, \end{cases}$$

$x_i \ge 0$ $(i = \overline{1, 6})$.

Ответ: z_{max}=98/13 в точке (0; 0; 0; 80; 0; 440).

7. ЭЛЕМЕНТЫ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Под динамическим программированием понимается метод оптимизации для операций, которые можно разбить на ряд шагов (этапов). Выбор решений на отдельных этапах можно рассматривать как управление реализацией данной операции. Для точной постановки задачи требуется введение некоторых новых понятий.

7.1. Прикладной пример и основные понятия

На ферме имеется стадо скота. Ежегодно часть стада отправляется на мясозаготовки, а остальная часть остается на ферме для воспроизводства. Доход от продажи скота выражается функцией $\varphi(u)$, где u — количество проданного скота. Функция $\varphi(u)$ может иметь, например, следующий вид (рис. 7.1)

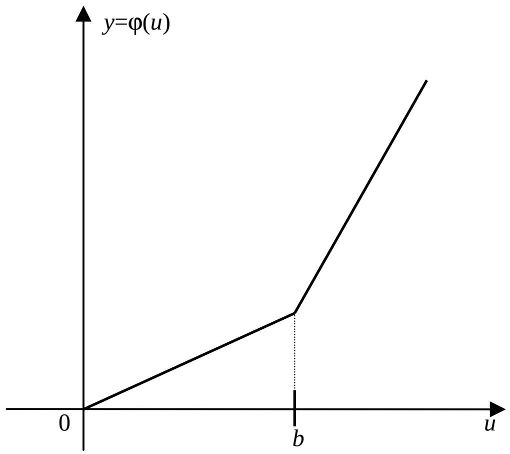


Рис. 7.1. Зависимость цены от количества проданного скота

Здесь поставки мяса сверх планового задания b оплачиваются по более высокой цене. Количество скота, оставленного на ферме для воспроизводства, в следующем году до начала мясозаготовок увеличивается в a раз, где $a \geq 1$. Требуется таким образом спланировать мясозаготовки на N лет, чтобы в итоге получить максимальный доход при условии, что минимальные ежегодные мясозаготовки составляют b.

Обозначим через $x(0)$ начальное количество скота на ферме, а через $x(t)$ — количество скота, оставленного на ферме к концу t-го года, $t=1,2,...,N$.

Количество голов скота, проданного для мясозаготовок в t-м году, обозначим через $u(t)$, $t=1,2,...,N$. В $t-1$ году, было оставлено для воспроизводства количество скота, равное $x(t-1)$. Следовательно, в t-м году перед мясозаготовками количество скота будет $ax(t-1)$. Из этого количества $u(t)$ будет продано для мясозаготовок, а остальная часть, т.е. $ax(t-1)-u(t)$, останется на ферме для воспроизводства. Доход фермы за N лет составит:

$$I = \varphi(u(1)) + \varphi(u(2)) + ... + \varphi(u(N)) = \sum_{t=1}^{N} \varphi(u(t)).$$

Учитывая обязательные мясозаготовки, мы имеем для управляющего параметра $u(t)$ ограничения :

$$u(t) \geq b, \quad t=1,2,...,N.$$

Кроме того, естественно считать выполненным условие

$$x(N) \geq d,$$

где d — плановое задание по разведению скота к концу N-летнего периода.

Итак, мы можем сформулировать следующую математическую задачу: требуется выбрать числа $u(1)$, $u(2)$, … , $u(N)$ таким образом, чтобы максимизировать функцию

$$I = \sum_{t=1}^{N} \varphi(u(t)) \to \max, \tag{7.1}$$

при условиях:

$$x(t) = ax(t-1) - u(t), \ (t = 1,2,...N); \tag{7.2}$$

$$u(t) \geq b, \ t = 1,2,...N; \tag{7.3}$$

$$x(0) = x_0, \ x(N) \geq d, \ x(t) \geq 0. \tag{7.4}$$

Полученная задача называется задачей оптимального управления. Функция целочисленного аргумента $u(t)$ называется управлением, поскольку описывает управление деятельностью фермы. Функция $x(t)$ называется переменной состояния или фазовой переменной, поскольку описывает состояние управляемого объекта (фермы). Соотношения (7.2) обычно называют уравнениями состояния, поскольку они определяют эволюцию состояний объекта при известном управлении. Условия (7.3) являются ограничениями на выбор управления. Условие (7.4) — условие на начальное и конечное состояние объекта.

Обобщая рассматриваемый пример, укажем общее математическое описание дискретных управляемых объектов. Будем считать, что переменная t может принимать лишь значения $t=0,1,2,...,N$, где N — фиксированное натуральное число. В общем случае предполагается, что можно воздействовать на управляемый объект, выбирая r управляющих параметров $u_1(t)$, ... $,u_r(t)$, или, что тоже самое, при каждом t выбирается управляющая точка $\vec{u}(t)$, имеющая r координат:

$$\vec{u}(t) = \{u_1(t), u_2(t), \ldots, u_r(t)\}.$$

Управлением мы условимся называть последовательность точек $\vec{u}(1), \vec{u}(2), \ldots, \vec{u}(N)$ в пространстве переменных u_1, u_2,..., u_r . Будем также считать, что в каждый момент времени t состояние объекта $\vec{x}(t)$ характеризуется n фазовыми координатами x_1 , x_2 , ..., x_n , т.е. в каждый момент времени t фазовое состояние $\vec{x}(t)$ имеет n координат:

$$\vec{x}(t) = \{x_1(t), x_2(t), \ldots x_n(t)\}.$$

При этом последовательность $\vec{x}(0), \vec{x}(1), \ldots, \vec{x}(N)$ состояний объекта в моменты $t=0,1,\ldots,N$ будем называть траекторией движения. В рассмотренном нами примере управление одномерно ($r=1$), и фазовое состояние описывается одним параметром ($n=1$).

Начальное состояние обычно считается заданным: $\vec{x}(0) = \vec{x}_0$. Дальнейшее поведение объекта однозначно определяется, если выбрано некоторое управление, с помощью следующих уравнений состояния:

$$x_i(t) = f_i(t, \vec{x}(t-1), \vec{u}(t)), \tag{7.5}$$

где $i=1,2,...,n$; $t=1,\ 2,\ldots,N$.

Соотношение (7.5) часто называют законом движения дискретного управляемого объекта. Траекторию объекта, удовлетворяющую уравнениям (7.5) называют соответствующей начальному состоянию $\vec{x}(0)$ и управлению $\vec{u}(t)$.

Во многих задачах управление $\vec{u}(t)$ не является произвольным. В нашем примере оно удовлетворяет условию (7.3). В общем же случае для каждого состояния $\vec{x}$ и момента времени t задается в пространстве управлений некоторое непустое множество $U_t(\vec{x})$ — область управления. При этом рассматривают лишь управления, которые удовлетворяют условию

$$\vec{u}(t) \in U_t(\vec{x}(t-1)), \quad t=1,\ 2,\ \ldots,\ N, \tag{7.6}$$

где траектория объекта исходит из начальной точки $\vec{x}(0)$. Управление, удовлетворяющее этому условию, называют допустимым (относительно начального состояния $\vec{x}(0)$). Соотношения (7.5) и (7.6) определяют дискретный управляемый объект. Процесс управления таким объектом осуществляется следующим образом. Поскольку задано начальное состояние $\vec{x}(0)$, нам известна соответствующая область управления $U_1(\vec{x}(0))$. Мы можем выбрать произвольную управляющую точку $\vec{u}(1) \in U_1(\vec{x}(0))$. После этого по формулам (7.5) мы определим состояние $\vec{x}(1)$ при $t=1$. Далее, зная $\vec{x}(1)$, мы можем рассмотреть соответствующую область управления $U_2(\vec{x}(1))$. Выбрав произвольную управляющую точку $\vec{u}(2) \in U_2(\vec{x}(1))$, мы можем согласно (7.5) найти следующее состояние $\vec{x}(2)$ и т.д. Очевидно, что управление $\vec{u}(1), \vec{u}(2), \ldots, \vec{u}(N)$, получающееся в результате такого последовательного выбора, является допустимым (относительно исходного начального состояния $\vec{x}(0)$). При этом траектория объекта $\vec{x}(0), \vec{x}(1), \ldots, \vec{x}(N)$ является соответствующей данному управлению.

Теперь можно поставить общую задачу оптимального управления для управляемого объекта. В качестве критерия эффективности, т.е. функции, показывающей насколько выгодным был выбранный процесс управления, рассмотрим выражение

$$I = \sum_{t=1}^{N} f_0(t, \vec{x}(t-1), \vec{u}(t)) . \tag{7.7}$$

Задача оптимального управления заключается в том, чтобы, зная начальное состояние, выбрать такое допустимое управление для объекта (7.5), (7.6), которое придает функционалу (7.7) максимальное значение.

В некоторых случаях речь может идти о минимальном значении функционала типа (7.7).

Сформулированная задача, которую часто называют основной, может быть также названа задачей с закрепленным левым концом и свободным правым. Здесь начальное состояние является заданным, а состояние в правом конце отрезка времени, т.е. $\vec{x}(N)$ ничем не связано (лишь бы значение функционала (7.7) было максимальным). Кроме основной задачи можно также рассматривать задачу с подвижными концами. В этом случае в фазовом пространстве задаются два множества M_0 и M_N. Требуется определить такое начальное состояние $\vec{x}(0) \in M_0$ и такое допустимое (относительно $\vec{x}(0)$) управление, чтобы было выполнено соотношение $\vec{x}(N) \in M_N$ и при этом функционал (7.7) принимал максимальное значение. Ясно, что если M_0 состоит из одной точки, а M_N совпадает со всем фазовым пространством, то задача с подвижными концами превращается в уже рассмотренную основную задачу.

Отметим, что в нашем примере управления фермой множество M_0 состоит из одной точки, а множество M_N определяется неравенством $x \geq d$, т.е. мы имеем здесь задачу с подвижными концами (точнее, с закрепленным левым и подвижным правым концом).

Наиболее общей задачей оптимального управления является задача с ограничениями на фазовые координаты. В этой задаче для каждого $t = 1, 2, \ldots, N$ задается в фазовом пространстве некоторое непустое множество M_t и требуется найти такое начальное состояние $\vec{x}(0)$ и такое допустимое (относительно $\vec{x}(0)$) управление, чтобы были выполнены соотношения $\vec{x}(t) \in M_t$ $(t = 0, 1, \ldots, N)$, и при этом функционал (7.7) принимал наибольшее возможное значение. Если множества $M_1, M_2, \ldots, M_{N-1}$ совпадают со всем фазовым пространством, то задача превращается в задачу с подвижными концами.

7.2. Дальнейшие примеры и принцип оптимальности. Метод динамического программирования

Выше мы сформулировали общую задачу оптимального управления объектом с дискретным временем. Приведем еще несколько различных примеров, чтобы убедиться в том, что подобные математические задачи встречаются часто и могут иметь весьма разнообразное содержание.

1. ***Нелинейная транспортная задача.*** В некоторых случаях можно рассматривать управляемый объект, не имея в виду его реальное существование во времени. Часто приходится вводить время искусственно, расчленяя решение некоторой задачи на условные шаги. Математическая формулировка задачи в этом случае формально совпадает с задачей оптимального управления и для ее решения можно применять аналогичные методы. Характерным примером могут служить распределительные задачи. Рассмотрим задачу о перевозке груза (сырья) от производителей (складов) к местам переработки (заводам).

Будем считать, что число складов равно трем, а число заводов обозначим через N. Предположим, что запас равен спросу, т.е. если обозначить через a_1, a_2, a_3 количество груза на складах, а через b_1, b_2, ..., b_N потребности заводов, то $a_1 + a_2 + a_3 = b_1 + b_2 + \ldots + b_N$. Стоимость перевозки u единиц груза с i-го склада на t-й завод зависит от u (т.е., от того, сколько сырья нужно привезти), а также от i и t (этими числами характеризуется дальность перевозки). Обозначим эту стоимость через $\varphi_{it}(u)$. Функция $\varphi_{it}(u)$ может, например, иметь вид, изображенный на рис. 7.2.

Задача заключается в том, чтобы перевезти сырье со складов на заводы с минимальными транспортными затратами. Распределение груза будем производить поэтапно. На каждом этапе удовлетворим некоторый завод. Таким образом, мы намечаем количество сырья, которое надо завести на 1-й завод с первого и второго складов, затем количество груза, перевозимого с этих же складов на 2-й завод, на 3-й завод и т.д. Количество же сырья, поставленного третьим складом, определится тогда однозначно: на каждый завод надо довезти с третьего склада недостающее количество сырья. Обозначим через $u_1(t)$ количество сырья, поставленного на t-й завод первым складом, а через $u_2(t)$ — вторым складом. При этом с третьего склада надо будет завести на t-й завод сырье в количестве $b_t(t) - u_1(t) - u_2(t)$.

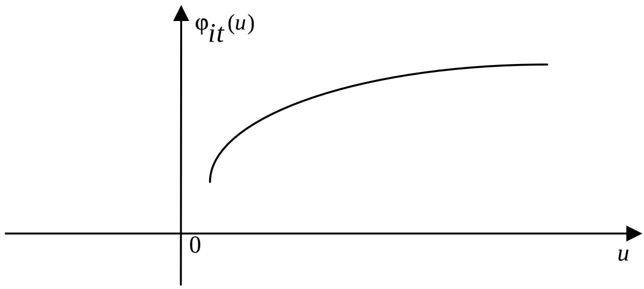


Рис. 7.2. Зависимость стоимости перевозки от количества перевезенного груза

Для составления плана нам достаточно выбрать числа $u_1(t)$, $u_2(t)$ при t=1, 2, … , N. При этом по смыслу задачи

$$u_1(t) \geq 0, u_2(t) \geq 0,\ \ u_1(t) + u_2(t) \leq b_t \text{ при } t = 1, 2, \ldots, N. \qquad (7.8)$$

Иными словами, "управляющая точка" $\vec{u}(t) = (u_1(t); u_2(t))$ должна находиться в треугольнике. Поскольку необходимо выполнить условия

$$u_1(t) \leq a_1,\ u_2(t) \leq a_2,$$

то множество U_t может быть пятиугольником (рис. 7.3).

Состояние складов (состояние объекта) при таком поэтапном распределении будем характеризовать величинами:

$x_1(t)$ — количество сырья, вывозимого с первого склада на первые t заводов

$x_2(t)$ — количество сырья, вывозимого со второго склада на первые t заводов.

Ясно, что

$$\begin{cases} x_1(t) = x_1(t-1) + u_1(t) \\ x_2(t) = x_2(t-1) + u_2(t) \end{cases}, \tag{7.9}$$

где $t = 1, 2, \dots, N$.

Рис. 7.3. Область допустимых управлений нелинейной транспортной задачи

Равенства (7.9) представляют собой уравнения состояния. Для удобства мы будем считать, что

$$x_1(0) = x_2(0) = 0. \tag{7.10}$$

Так как запас равен спросу, то после вывоза сырья на все N заводов склады должны остаться пустыми, т.е.

$$x_1(N) = a_1;\ x_2(N) = a_2. \tag{7.11}$$

Критерием эффективности плана здесь является стоимость всех перевозок:

$$I = \sum_{t=1}^{N} \left[\varphi_{1t}(u_1(t)) + \varphi_{2t}(u_2(t)) + \varphi_{3t}(b_t - u_1(t) - u_2(t) \right].$$

Таким образом, транспортная задача приобретает следующую формулировку: для управляемого объекта (7.9) найти такое допустимое управление $u_1(t)$, $u_2(t)$ (удовлетворяющее условию (7.8)), чтобы для соответствующей траектории $x_1(t)$, $x_2(t)$ $(t = 1, 2, \dots, N)$ с начальным условием (7.10) удовлетворялось конечное условие (7.11), и при этом функционал I принимал наименьшее возможное значение. Очевидно, что это частный случай задачи с подвижными концами, в которой множества M_0 и M_1 состоят из одной точки каждое.

2. ***Задача о распределении ресурсов.*** Мы рассмотрим наиболее простой вариант этой задачи. В нашем распоряжении имеется какой-то запас средств (ресурсов) в количестве K единиц. Эти средства нужно распределить между предприятиями Π_1, Π_2, ..., Π_N. Каждое предприятие Π_t при вложении в него средств и приносит доход, зависящий от u, т.е. представляющий собой функцию $\varphi_t(u)$, которая считается известной. Функции $\varphi_t(u)$ $(t = 1, 2, \ldots , N)$ являются неубывающими. Задача заключается в распределении средств между предприятиями с целью получения максимального суммарного дохода. Хотя в такой постановке задача не содержит упоминания о времени, можно все же операцию распределения средств между предприятиями мысленно развернуть в какой-то последовательности, считая за первый шаг вложение средств в предприятие Π_1 , за второй — в Π_2 и т.д. Управляемым объектом здесь являются средства, а состояние объекта определяется параметром $x(t)$ — количеством средств, которые остаются после вложения в t-е предприятие. Управляющим параметром здесь является $u(t)$ — количество средств, вложенных в предприятие Π_t. Очевидно, что

$$x(t) = x(t-1) - u(t),\ (t=1, 2, \ldots , N).$$

Эти равенства представляют собой уравнения состояния. Управление по своему смыслу неотрицательно и не больше общей суммы наличных средств:

$$u(t) \ge 0, \quad u(t) \le x(t-1).$$

Последние неравенства определяют множество $U_t\ (x(t-1))$ допустимости управления. Задача наша заключается в нахождении оптимального управления $u(1), u(2), \ldots , u(N)$, для которого суммарный доход максимален:

$$I = \sum_{t=1}^{N} \varphi_t(u(t)) \to \max .$$

При этом в начальный момент времени $x(0)=K$, кроме того, $x(N) = 0$. Последнее условие означает, что все средства распределяются.

3. ***Принцип оптимальности Беллмана.*** Решение задач оптимального управления с дискретным временем основано на особом принципе, который называют принципом оптимальности. Метод решения этих задач называется методом динамического программирования. Сам принцип оптимальности очень прост. Его можно пояснить следующим образом.

Критерий оптимальности или целевой функционал (7.7) является суммой слагаемых, каждое из которых зависит от состояния объекта в начале t-го шага и от управления, выбираемого на этом шаге процесса. Можно пытаться так выбрать управление, чтобы оптимизировать именно это t-е слагаемое. Однако, уже выбрав управление, мы изменим состояние объекта и это, может быть, не позволит нам в будущем получить еще больший выигрыш на других слагаемых. Таким образом, шаговое управление должно выбираться дальновидно, с учетом всех его последствий в будущем. Значит, планируя многошаговую операцию, надо выбирать управление с учетом всех его будущих последствий на еще предстоящих шагах. Однако среди всех шагов есть один, который можно планировать без оглядки на будущее, чтобы он сам, как таковой, принес наибольшую выгоду. Это последний шаг. И если управление уже найдено, и в нем перед последним шагом объект находится в состоянии $\vec{x}(N-1)$, то управление $\vec{u}(N)$ должно быть точкой максимума функции

$$f_0\left(N, \vec{x}(N-1), \vec{u}(N)\right).$$

Правда, найти это управление нельзя, поскольку нам неизвестно состояние $\vec{x}(N-1)$ объекта перед последним шагом. Аналогично этому в оптимальном управлении перед T-м шагом, когда объект находится в состоянии $\vec{x}(T-1)$, управления $\vec{u}(T), \vec{u}(T+1),\ldots, \vec{u}(N)$ должны обеспечивать максимум функции

$$I(T) = \sum_{t=T}^{N} f_0\left(t, \vec{x}(t-1), \vec{u}(t)\right). \tag{7.12}$$

Теперь можно сформулировать сам принцип.

Принцип оптимальности. *Оптимальное управление обладает тем свойством, что каково бы ни было начальное состояние и начальное управление, последующее управление должно быть оптимальным по отношению к состоянию, получающемуся в результате действия начального управления.*

Использование этого принципа гарантирует, что управление, выбранное на любом шаге, является не локально лучшим, а лучшим с точки зрения процесса в целом.

Теперь изложим в общих чертах сам метод динамического программирования. Рассматривается задача с заданным $\vec{x}(0)$. Уже отмечалось, что управление на последнем шаге $\vec{u}(N)$ должно быть точкой максимума функции

$$f_0(N, \vec{x}(N-1), \vec{u}(N)).$$

Этот максимум определить нельзя, т.к. нам неизвестно состояние $\vec{x}(N-1)$ после $N-1$-го шага. Предположим, что мы можем находить максимум функции $f_0(N, \vec{x}, \vec{u})$ по переменным $\vec{u}$ при любых допустимых значениях $\vec{x}$. Естественно, что точка максимума $\vec{u}$ может зависеть от значений $\{x_1, x_2, \ldots, x_N\} = \vec{x}$, т.е. $\vec{u}_{\max} = \vec{u}_N(\vec{x}) = \{u_{1N}(\vec{x}), u_{2N}(\vec{x}), \ldots, u_{rN}(\vec{x})\}$. Будем считать функцию $\vec{u}_N(\vec{x})$ определённой. При её нахождении, разумеется, учитываются ограничения на $\vec{u}$ $(\vec{u} \in U_N(\vec{x}))$. Если подставить вместо $\vec{u}$ функцию $\vec{u}(\vec{x})$ в выражение для $f_0(N, \vec{x}, \vec{u})$, то мы получим некоторую функцию

$$\omega_N(\vec{x}) = f_0(N, \vec{x}, \vec{u}_N(\vec{x})) .$$

Рассмотрим теперь $(N-1)$-й шаг. В его начале объект находится в состоянии $\vec{x}(N-2)$, которое нам неизвестно. При этом, $x_i(N-1) = f_i(N-1, \vec{x}(N-2), \vec{u}(N-1))$. Согласно принципу максимума $\vec{u}(N-1)$ должна быть точкой максимума функции

$$f_0(N-1, \vec{x}(N-2), \vec{u}(N-1)) + \omega_N(\vec{f}(N-1, \vec{x}(N-2), \vec{u}(N-1))).$$

Этот максимум определить нет возможности, т.к. состояние $\vec{x}(N-2)$ нам неизвестно. Пусть нам удается найти максимум функции

$$f_0(N-1, \vec{x}, \vec{u}) + \omega_N(\vec{f}(N-1, \vec{x}, \vec{u})), \qquad (7.13)$$

который в общем случае зависит от $\vec{X}$. Обозначим его через $\vec{u}_{N-1}(\vec{x})$ и, подставляя в выражение (13), получим некоторую функцию $\omega_{N-1}(\vec{x})$. Продолжая рассматривать шаги с меньшими номерами, мы на k-ом шаге находим функцию

$$\omega_{k-1}(\vec{x}) = \max_{\vec{u} \in U_{r-1}(\vec{x})} \{ f_0(k-1, \vec{x}, \vec{u}) - \omega_k(\vec{f}(k-1, \vec{x}, \vec{u}))$$

Последнее уравнение называется функциональным уравнением Беллмана. С помощью этого уравнения мы рекуррентно находим функции

$$\omega_N(\vec{x}), \omega_{N-1}(\vec{x}), \ldots, \omega_1(\vec{x})$$

и также точки максимума

$$\vec{u}_N(\vec{x}), \vec{u}_{N-1}(\vec{x}), \ldots, \vec{u}_1(\vec{x}).$$

Эти точки максимума позволяют определить оптимальное управление. Действительно, нам известно начальное состояние $\vec{x}(0)$. Поэтому управление на первом шаге есть

$$\vec{u}(1) = \vec{u}_1(\vec{x}(0)).$$

Зная это управление, мы находим состояние в начале второго шага $\vec{x}(1)$ с помощью уравнений состояния. Затем находим управление на втором шаге:

$$\vec{u}(2) = \vec{u}_2(\vec{x}(1)).$$

Таким образом, определяется оптимальное управление шага *t*:

$$\vec{u}(t) = \vec{u}_t(\vec{x}(t-1)).$$

Соответствующая траектория находится с помощью уравнений состояния параллельно с нахождением оптимального управления.

В процессе оптимизации управления методом динамического программирования многошаговый процесс “проходится” дважды. Первый раз — от конца к началу, в результате чего находятся условные оптимальные управления $\vec{u}_k(\vec{x})$ и условные оптимальные выигрыши $\omega_k(\vec{x})$ за оставшийся “хвост” процесса. Второй раз процесс “проходится” от начала к концу, когда нам остается только “прочитать” уже готовые рекомендации и найти безусловное оптимальное управление $\vec{u}(t)$, состоящее из оптимальных шаговых управлений $\vec{u}(1)$, $\vec{u}(2)$, …, $\vec{u}(N)$.

Первый этап — этап условной оптимизации — несравненно сложнее и длительнее второго. Второй этап почти не требует дополнительных вычислений.

7.3. Примеры решения задач динамического программирования

Пример 7.1. Пользуясь методом динамического программирования, решим конкретную задачу распределения ресурсов, размер которых К=200 млн.руб., между четырьмя предприятиями П1, П2, П3, П4 (*N*=4). Функции дохода на каждом из четырех предприятий задаются равенствами:

$$f_1(u) = 0{,}4u;$$

$$f_2(u) = 0{,}6u;$$

$$f_3(u) = 0{,}8u;$$

$$f_4(u) = 0.7u.$$

Таким образом, функции дохода считаются линейными, а общий целевой функционал имеет вид

$$I = 0{,}4u(1) + 0{,}6u(2) + 0{,}8u(3) + 0{,}7u(4).$$

Управление $u(t)$ $(t = 1, 2, 3, 4)$ нужно выбрать так, чтобы максимизировать I. Уравнения состояния здесь имеют вид

$$x(t) = x(t-1) - u(t),\ t = 1, 2, 3, 4.$$

Управления выбираются из условий

$$u(t) \ge 0,\ u(t) \le x(t-1).$$

Кроме того, мы считаем, что $x(0) = K$, $x(4) = 0$.

Согласно общей схеме на последнем шаге мы максимизируем функцию $f_4(u)$ на отрезке $0 \le u \le x(3)$ Очевидно максимум этой функции будет достигаться в точке $x(3)$ Таким образом, мы имеем

$$\omega(x) = 0{,}7x;\quad u_4(x) = x.$$

Рассмотрим теперь третий шаг. В его начале состояние есть $x(2)$, причем $x(3) = x(2) - u(3)$. Поэтому на третьем шаге мы должны максимизировать функцию

$$0{,}8u(3) + 0{,}7\big(x(2) - u(3)\big)$$

на отрезке $0 \le u(3) \le x(2)$. Следовательно

$$\omega_3(x) = \max_{0 \le u \le x} (0{,}1u + 0{,}7x) = 0{,}8x;\ u_3(x) = x.$$

На втором шаге $x(2) = x(1) - u(2)$, и максимизируется функция

$$0{,}6u(2) + 0{,}8\big(x(1) - u(2)\big),$$

то есть

$$\omega_2(x) = \max_{0 \le u \le x} (0{,}8x - 0{,}2u) = 0{,}8x,\ u_2(x) = 0.$$

Наконец, на первом шаге $x(1) = x(0) - u(1)$, и максимизируется функция

$$0{,}4u(1) + 0{,}8\big(x(0) - u(1)\big);$$

$$\omega_1(x) = \max_{0 \le u \le x} (0{,}8x - 0{,}4u) = 0{,}8x,\ u_1(x) = 0\cdot$$

Поскольку $x(0)$=200, то мы получим

$$u(1) = u_1(200) = 0;\ \omega_1(200) = 160 (= 0{,}8\cdot 200);$$

$$x(1) = 200 - 0 = 200.$$

$$u(2) = u_2(x(1)) = u_2(200) = 0;\ \omega_2(200) = 160;$$

$$x(2) = 200 - 0 = 200.$$

$$u(3) = u_3(x(2)) = u_3(200) = 200;\ \omega_3(200) = 160;$$

$$x(3) = 200 - 200 = 0.$$

$$u(4) = u_4(x(3)) = u_4(0) = 0{,}7\cdot 0 = 0; \quad \omega_4(0) = 0{,}7\cdot 0 = 0;$$
$$x(2) = 200 - 0 = 200.$$

Таким образом, мы имеем следующие управления:

$$u(1) = 0;\; u(2) = 0;\; u(3) = 200;\; u(4) = 0.$$

Общий доход при этом составит $\omega_1(200) = 160$ млн. руб.

Этот результат, однако, можно было бы предвидеть заранее, поскольку третье предприятие имеет наивысший коэффициент прибыли на вложенный рубль и выгоднее всего вкладывать все средства именно в это предприятие.

Пример 7.2. Решим теперь задачу об оптимальных мясозаготовках. В начале периода управления на ферме имелось 1200 голов скота. Период управления 4 года. Минимальные ежегодные мясозаготовки составляют 150 голов. После периода управления численность скота на ферме не должна быть меньше 800 голов. Численность скота, оставляемого на ферме для воспроизводства, в следующем году увеличивается в 1,4 раза. Доход от продажи скота выражается функцией, представленной на рис. 7.4.

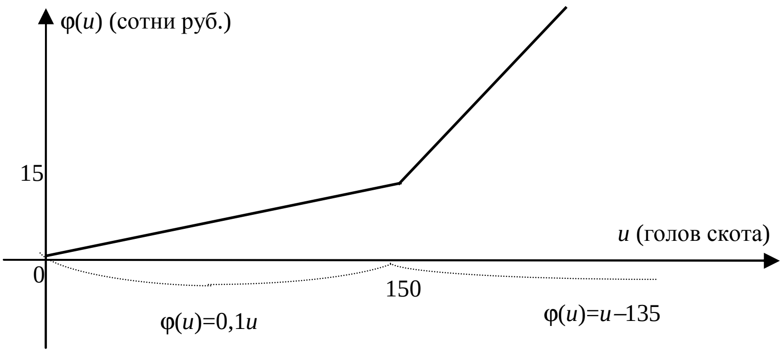


Рис. 7.4. Функция $\varphi(u)$ примера 7.2

Введём обозначения:

$x(t)$ — количество скота, оставленного на ферме к концу t-го года;

$u(t)$ — количество скота, проданное в t-ом году.

Имеем уравнение состояния:

$$x(t) = 1{,}4x(t-1) - u(t), \quad t = 1, 2, 3, 4.$$

Требуется так выбрать управление, чтобы максимизировать

$$I = \varphi(u(1)) + \varphi(u(2)) + \varphi(u(3)) + \varphi(u(4)).$$

При этом $150 \le u(t) \le 1{,}4x(t-1)$ и $x(0) = 1200$; $x(4) \ge 800$. Последние неравенства означают, что $u(4) \le 1{,}4x(3) - 800$.

Начинаем оптимизацию с четвертого шага. Нужно определить максимум функции

$$\varphi(u(4)) = u - 135 \quad \text{при} \quad 150 \le u(4) \le 1{,}4x(3) - 800.$$

Поскольку функция φ монотонно растет, то $u_4(x)$=1.4x–800,

$$\omega_4(x) = 1{,}4x - 800 - 135 = 1{,}4x - 935.$$

На третьем шаге максимизируется функция ($x(3) = 1{,}4x(2) - u(3)$):

$$\varphi(u(3)) + \omega_4(1{,}4x(2) - u(3)) \quad \text{при} \quad 150 \le u(3) \le 1{,}4x(2),$$

то есть нужно определить

$$\omega_3(x) = \max_{150 \le u \le 1{,}4x} \left(u - 135 + 1{,}4(1{,}4x - u - 935)\right) =$$

$$= \max_{150 \le u \le 1{,}4x} \left(1{,}96x - 0{,}44u - 1070\right) = 1{,}96x - 1130;$$

$$u_3(x) = 150.$$

На втором шаге максимизируется функция

$$\varphi(u(2)) + \omega_3(1{,}4x(1) - u(2)) \quad \text{при условиях } 150 \le u(2) \le 1{,}4x(1).$$

$$\omega_2(x) = \max_{150 \le u \le 1{,}4x} \left(u - 135 + 1{,}96(1{,}4x - u) - 1130\right)$$

$$= \max_{150 \le u \le 1{,}4x} \left(2{,}704x - 0{,}96u - 1265\right) = 2{,}704x - 1409;$$

$$u_2(x) = 150.$$

На первом шаге максимизируется функция

$$\varphi(u(1)) + \omega_2(1{,}4x(0) - u(1)) \quad \text{при условиях } 150 \le u(1) \le 1{,}4x(0).$$

$$\omega_1(x) = \max_{150 \le u \le 1{,}4x} \left(u - 135 + 2.704(1{,}4x - u) - 1409\right)$$

$$= \max_{150 \le u \le 1{,}4x} \left(3{,}7856x - 0{,}704u - 1544\right) =$$

$$= 3{,}7856x - 1649{,}6;$$

$$u_1(x) = 150.$$

Этап условной оптимизации закончен. Оптимальный выигрыш за четыре года составит

$$\omega_1(x(0)) = \omega_1(1200) = 4542{,}72 - 1649{,}6 = 2893{,}12;$$

$$u(1) = u_1(1200) = 150;$$

$$x(1) = 1{,}4x(0) - u(1) = 1{,}4 \cdot 1200 - 150 = 1530;$$

$$u(2) = u_2(1530) = 150;$$

$$x(2) = 1{,}4x(1) - u(2) = 1{,}4 \cdot 1530 - 150 = 1992;$$

$$u(3) = u_3(1992) = 150;$$

$$x(3) = 1{,}4x(2) - u(3) = 1{,}4 \cdot 1992 - 150 = 2788{,}8 - 150 = 2638{,}8;$$

$$u(4) = u_4(2638{,}8) = 1{,}4 \cdot 2638{,}8 - 800 = 2894{,}32;$$

$$x(4) = 1{,}4x(3) - u(4) = 800.$$

Таким образом, оптимальная стратегия в данном случае заключается в том, чтобы продавать в первые три года минимальное количество скота (150 голов), а в последнем году продать весь скот, оставив на ферме лишь 800 голов.

Пример 7.3. Попытаемся теперь решить ту же задачу с иной функцией дохода $\varphi(u)$. Пусть эта функция имеет вид (рис. 7.5.)

$$\varphi(u) = \begin{cases} 2u,\ u \le 300; \\ 0{,}5u + 450,\ u > 300. \end{cases}$$

Все остальные условия предыдущей задачи остаются прежними.

На четвертом шаге

$$\omega_4 = \max_{150 \le u \le 1{,}4x - 800} \begin{cases} 2(1{,}4x - 800), 1{,}4x - 800 \le 300; \\ 0{,}5(,4x - 800) + 450, 1{,}4x - 800 > 300; \end{cases}$$

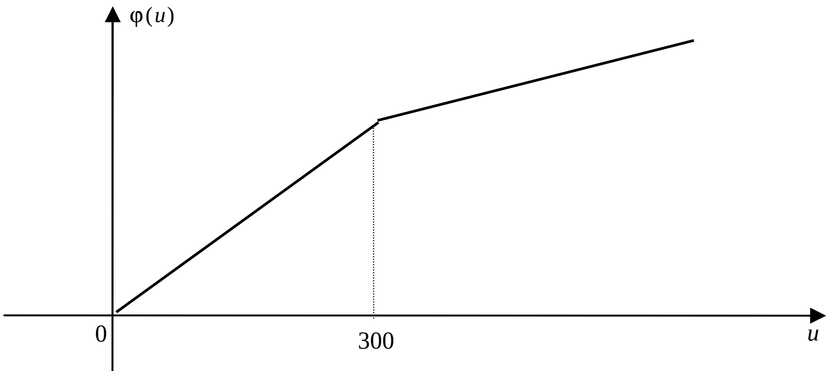


Рис 7.5. Функция $\varphi(u)$ примера 7.3

$$\omega_4 = \begin{cases} 2{,}8x - 1600, x \le \dfrac{5500}{7}; \\ 0{,}7x - 50, x > \dfrac{5500}{7}; \end{cases}$$

$$u_4(x) = 1{,}4x - 800$$

На третьем шаге ($x(3) = 1{,}4x(2) - u(3)$):

$$\omega_3(x) = \max_{150 \le u \le 1{,}4x} \left(\varphi(u) + \omega_4(1{,}4x - u)\right).$$

Вычисление функции $\omega_3(x)$ встречает большие трудности и требует большой изобретательности. Нахождение функций $\omega_2(x)$ и $\omega_1(x)$

встречает еще большие трудности. Таким образом, аналитическое решение задач динамического программирования часто приводит к сложным задачам на глобальный экстремум для функций, зависящих от параметра. Преодолеть эти трудности часто удается с помощью численной реализации алгоритма динамического программирования на ЭВМ.

При численной реализации управляющим параметрам придают лишь дискретное множество значений, которые, например, кратны некоторым величинам. В одних задачах это новое условие является естественным, в других — вводится искусственно. При этом решение, полученное с этим дополнительным условием, дает приближенное представление об истинном оптимальном управлении. Так, в задаче примера 3 предположение, что сдача скота на мясозаготовки может происходить партиями по 50 голов является искусственным. Однако при этом количество возможных значений параметра u будет конечным. Конечным будет и количество возможных состояний. Предположение же, что u принимает значения, кратные единице, является естественным. Однако оно приводит к очень большому числу (хотя и конечному) возможных управлений и состояний. Численная реализация задачи тем проще, чем меньше возможное количество управлений и состояний нужно рассматривать на каждом шаге управления. Поэтому предположение, что u в примере 7.3 принимает значения, кратные 50, приводит к более простой численной реализации задачи. Впрочем, проследить эту реализацию и в этом случае для нас сложно из-за большого числа возможных управлений.

Пример 7.4. Рассмотрим задачу о распределении ресурсов. Исходный запас средств $K = 10$ (условных единиц). Требуется его оптимальным образом распределить между пятью ($N = 5$) предприятиями. Предполагается, что вкладываются лишь целые количества средств. Функции дохода $\varphi_t(u)$ заданы в следующей таблице:

u	$\varphi_1(u)$	$\varphi_2(u)$	$\varphi_3(u)$	$\varphi_4(u)$	$\varphi_5(u)$
1	0,5	0,1	0,6	0,3	1,0
2	1,0	0,5	1,1	0,6	1,2
3	1,4	1,2	1,2	1,3	1,3
4	2,0	1,8	1,4	1,4	1,3
5	2,5	2,5	1,6	1,5	1,3
6	2,8	2,9	1,7	1,5	1,3
7	3,0	3,5	1,8	1,5	1,3

В каждом столбце, начиная с какой-то суммы вложений, доходы перестают возрастать (остаются постоянными). Это связано с тем, что каждое предприятие способно "освоить" лишь ограниченное количество средств. Число возможных значений u на первом шаге равно 11 (с учетом значения $u = 0$). Поскольку функция $\varphi_t(u)$ стабилизируется при $t \geq 7$, мы не приводим еще трех строк этой таблицы. Напомним, что уравнение состояния здесь имеет вид

$$x(t) = x(t-1) - u(t),\ t = 1,\ 2,\ 3,\ 4,\ 5\,,$$

где $x(t)$ — количество еще не распределенных средств; $u(t)$ — количество средств, выделяемых предприятию с номером t,

$$0 \leq u(t) \leq x(t-1),\quad t = 1, 2, 3, 4, 5.$$

Кроме того, $u(t)$ при каждом t может принимать целые значения.

$$x(0) = 10,\ \ x(5) = 0.$$

Мы должны максимизировать функцию $I = \sum_{t=1}^{5} \varphi_t\left(u(t)\right)$.

При $t = 5$ мы находим $\omega_5(x) = \max\limits_{0 \leq u \leq x} \varphi_5(u)$.

Поскольку функция $\varphi_5(x)$ неубывающая, то $\omega_5(x) = \varphi_5(x)$, $u_5(x) = x$. Результаты условной оптимизации заносим в следующую таблицу:

	$t = 5$		$t = 4$		$t = 3$		$t = 2$		$t = 1$	
x	u_5	ω_5	u_4	ω_4	u_3	ω_3	u_2	ω_2	u_1	ϖ_1
1	$\boxed{1}$	1,0	$\boxed{0}$	1,0	0	1,0	0	1,0		
2	2	1,2	1	1,3	1	1,6	0	1,6		
3	3	1,3	2	1,6	2	2,1	0	2,1		
4	4	1,3	3	2,3	2	2,4	0	2,4		
5	5	1,3	3	2,5	1	2,9	0	2,9		
6	6	1,3	4	2,6	2	3,4	5	3,5		
7	7	1,3	4	2,7	2	3,6	5	4,1		
8	8	1,3	5	2,8	4	3,7	$\boxed{5}$	4,6		
9	9	1,3	6	2,8	5	3,9	7	5,1		
10	10	1,3	7	2,8	5	4,1	7	5,6	$\boxed{2}$	$\boxed{5,6}$

Заполняя эту таблицу при t = 4, мы находим

$$\omega_4(x) = \max_{0 \le u \le x}\left(\varphi_4(u) + \omega_5(x-u)\right).$$

Нахождение максимума производится непосредственно при каждом x:

$x = 1$

u	$\varphi_4(u) + \omega_5(1-u)$	
0	0 + 1,0	$\boxed{1{,}0}$
1	0,3 + 0	0,3

$\omega_4(1)$ =1,0
$u_4(1) = 0$

$x = 2$

u	$\varphi_4(u) + \omega_5(2-u)$	
0	0 + 1,2	1,2
1	0,3 + 1,0	$\boxed{1{,}3}$
2	0,6+0	0,6

$\omega_4(2)$ =1,3
$u_4(2) = 1$

$x = 3$

u	$\varphi_4(u) + \omega_5(3-u)$	
0	0 + 1,3	1,3
1	0,3 + 1,2	1,5
2	0,6 + 1,0	$\boxed{1{,}6}$
3	1,3+0	1,3

$\omega_4(3)$ =1,6
$u_4(3) = 2$

$x = 4$

u	$\varphi_4(u) + \omega_5(4-u)$	
0	0 + 1,3	1,3
1	0,3 + 1,3	1,6
2	0,6 + 1,2	1,8
3	1,3 + 1,0	$\boxed{2{,}3}$
4	1,4 + 0	1,4

$\omega_4(4)$ =2,3
$u_4(4) = 3$

$x = 5$

u	$\varphi_4(u) + \omega_5(5-u)$	
0	0 + 1,3	1,3
1	0,3 + 1,3	1,6
2	0,6 + 1,3	1,9
3	1,3 + 1,2	$\boxed{2{,}5}$
4	1,4 + 1,0	2,4
5	1,5 + 0	1,5

$\omega_4(5)$ =2,5
$u_4(5) = 3$

$x = 6$

u	$\varphi_4(u) + \omega_5(6-u)$	
0	0 + 1,3	1,3
1	0,3 + 1,3	1,6
2	0,6 + 1,3	1,9
3	1,3 + 1,3	2,6
4	1,4 + 1,2	$\boxed{2,6}$
5	1,5 + 1,0	2,5
6	1,5 + 0	1,5

$\omega_4(6) = 2,6$

$u_4(6) = 4$

$x = 7$

u	$\varphi_4(u) + \omega_5(7-u)$	
0	0 + 1,3	1,3
1	0,3 + 1,3	1,6
2	0,6 + 1,3	1,9
3	1,3 + 1,3	2,6
4	1,4 + 1,3	$\boxed{2,7}$
5	1,5 + 1,2	2,7
6	1,5 + 1,0	2,5
7	1,5 + 0	1,5

$\omega_4(7) = 2,7$

$u_4(7) = 4$

$x = 8$

u	$\varphi_4(u) + \omega_5(8-u))$	
0	0 + 1,3	1,3
1	0,3 + 1,3	1,6
2	0,6 + 1,3	1,9
3	1,3 + 1,3	2,6
4	1,4 + 1,3	2,7
5	1,5 + 1,3	$\boxed{2,8}$
6	1,5 + 1,2	2,7
7	1,5 + 1,0	2,5
8	1,5 + 0	1,5

$\omega_4(8) = 2,8$

$u_4(8) = 5$

$x = 9$

u	$\varphi_4(u) + \omega_5(9 - u)$	
0	0 + 1,3	1,3
1	0,3 + 1,3	1,6
2	0,6 + 1,3	1,9
3	1,3 + 1,3	2,6
4	1,4 + 1,3	2,7
5	1,5 + 1,3	2,8
6	1,5 + 1,3	$\boxed{2{,}8}$
7	1,5 + 1,2	2,7
8	1,5 + 1,0	2,5
9	1,5 + 0	1,5

$\omega_4(9) = 2{,}8$

$u_4(9) = 6$

$x = 10$

u	$\varphi_4(u) + \omega_5(10 - u)$	
0	0 + 1,3	1,3
1	0,3 + 1,3	1,6
2	0,6 + 1,3	1,9
3	1,3 + 1,3	2,6
4	1,4 + 1,3	2,7
5	1,5 + 1,3	2,8
6	1,5 + 1,3	2,8
7	1,5 + 1,3	$\boxed{2{,}8}$
8	1,5 + 1,2	2,7
9	1,5 + 1,0	2,5
10	1,5 + 0	1,5

$\omega_4(10) = 2{,}8$

$u_4(10) = 7$

При t=3 мы находим

$$\omega_3(x) = \max_{0 \le u \le x} \left(\varphi_3(u) + \omega_4(x - u)\right).$$

Соответствующие столбцы для $t = 3$ заполняются аналогично с использованием столбца для $\omega_4(x)$ и столбца $\varphi_3(x)$ таблицы функций дохода. Так же мы заполняем и столбцы, отвечающие $t = 2$. При этом

$$\omega_2(x) = \max_{0 \le u \le x} \left(\varphi_2(u) + \omega_3(x-u)\right).$$

При заполнении столбцов для $t = 1$ можно ограничиться случаем $x = 10$, поскольку начальное состояние $x(0) = 10$, $u(1) = u_1(10) = 2$. Максимальная прибыль здесь равна 5,6 условных единиц. Состояние при $t = 1$ $x(1) = x(0) - u(1) = 10 - 2 = 8$. По таблице значений $u_2(x)$ находим $u_2(8) = u(2) = 5$. Потом $x(2) = x(1) - u(2) = 8 - 5 = 3$. Далее $u(3) = u_3(3)=2$, $x(3) = x(2) - u(3) = 3 - 2 = 1$, $u(4) = u_4(1) = 0$, $x(4) = x(3) - u(4) = 1 - 0 = 1$.

Таким образом, мы имеем следующее оптимальное управление:

$u(1)=2$, $u(2)=5$, $u(3)=2$, $u(4)=0$, $u(5)=1$,

т.е. первому предприятию следует выделить 2 ед. средств, второму — 5 ед., третьему —2 ед., четвертому — 0 ед. и пятому — 1 ед.

Контрольные вопросы и задачи для самостоятельного решения

1. Что входит в описание дискретного управляемого объекта? Приведите пример дискретного управляемого объекта, определив для него переменные состояния, управляющие параметры, уравнение состояния и целевой функционал.

2. В чем заключается основная задача оптимального управления дискретным управляемым объектом? Сформулируйте задачу с подвижными концами, а также задачу с ограничениями на фазовые переменные.

3. Дайте содержательную и математическую формулировки нелинейной транспортной задачи.

4. Сформулируйте задачу об оптимальном распределении ресурсов в виде задачи оптимального управления.

5. В чем заключается принцип оптимальности Беллмана?

6. Опишите метод динамического программирования?

Решение каждой содержательной задачи состоит из двух этапов: 1) этап формализации, то есть составление оптимизационной математической модели; 2) этап решения задачи с помощью того или иного метода. Для задач динамического программирования этап формализации требует выделения управляемой системы, переменных состояния, управляющих параметров (управлений), составления уравнений состояния и целевого функционала. Этап формализации часто вызывает значительные трудности.

По изложенным выше образцам формализовать задачи **7.1**; **7.2**.

7.1. В склад емкостью W(м^3) требуется поместить N различных типов оборудования. Объем одной единицы оборудования i-го типа (i=1,2,3,...,N) равен V_i(м^3). Стоимость единицы оборудования i-го типа равна C_i(руб). Определить, сколько оборудования каждого типа следует поместить в склад, чтобы общая стоимость складированного оборудования была максимальной.

7.2. Детали N видов могут обрабатываться на двух станках. Время обработки детали i-го вида (i=1,2,3,...,N) на первом станке равно a_i минут, а на втором станке-b_i минут. Очередность обработки одна и та же: сначала деталь обрабатывается на первом станке, а затем на втором. Требуется выбрать такую последовательность обработки деталей, при которой время изготовления всех деталей будет минимальным.

Следующие конкретные задачи решить методом динамического программирования.

7.3. Найти оптимальный план загрузки склада в условиях задачи 7.1 при W=90м^3; N=3; v_1=24м^3; v_2= 19м^3; v_3=16м^3; c_1=960руб; c_2=500руб; c_3=250руб.

7.4. Найти оптимальную последовательность обработки деталей в условиях задачи 7.2 при N=4; a_1=20мин; a_2= 35мин; a_3=15мин; a_4=50мин; b_1=5мин; b_2=10 мин; b_3=20мин; b_4=7мин.

7.5. Найти оптимальное распределение средств К=700тыс.руб. между тремя предприятиями, если зависимость между капиталовложениями и приростом выпуска продукции задается следующей таблицей:

Объем капиталовложений	Прирост выпуска продукции (тыс. руб.)		
(тыс. руб.)	Предприятие 1	Предприятие 2	Предприятие 3
0	0	0	0
100	30	50	40
200	50	80	50
300	90	90	110
400	110	150	120
500	170	190	180
600	180	210	220
700	210	220	240

Оптимальное распределение средств должно обеспечить максимальный суммарный прирост выпуска продукции.

7.6. Решить методом динамического программирования линейную транспортную задачу со следующими условиями:

Запасы	Потребности			
	70	10	20	20
50	1	3	4	2
45	7	6	3	1
25	9	4	8	2

8. ЭЛЕМЕНТЫ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ

8.1. Простейший поток событий

Потоком событий называется последовательность однородных событий, следующих одно за другим в какие-то случайные моменты времени. Примерами потоков событий являются: а) поток вызовов на телефонной станции; б) поток сбоев (отказов) ЭВМ; в) поток железнодорожных составов, поступающих на сортировочную станцию и так далее.

Поток называется обладающим *свойством стационарности*, если вероятность появления k событий на промежутке времени $(t;\ t+\Delta t)$ не зависит от начала промежутка t, а зависит лишь от числа событий k и длины промежутка Δt.

Поток обладает *свойством отсутствия последствия*, если для любых двух непересекающихся промежутков времени длиной Δt_1 и Δt_2 события A_{1k} и A_{2m}, состоящие в появлении на этих промежутках k и m событий соответственно, являются независимыми.

Поток обладает *свойством ординарности*, если вероятность появления на малом промежутке времени Δt более одного события пренебрежимо мала по сравнению с вероятностью появления только одного события.

Более точно можно сказать, что вероятность $P_k(\Delta t)$ появления k событий на промежутке Δt стремится к нулю при $\Delta t \to 0$; при этом, если $k > 1$ это стремление более быстрое, чем при $k = 1$.

Поток называется *простейшим* или *пуассоновским*, если он обладает свойствами стационарности, отсутствия последствия и ординарности.

Для любого потока можно ввести понятие интенсивности, то есть среднего числа событий, появляющихся в единицу времени. Для этого выберем промежуток времени длиной 1 и рассмотрим случайную величину–количество событий, появляющихся на этом промежутке. Для простейшего потока закон распределения этой величины не зависит от расположения единичного отрезка времени из-за свойства стационарности. Математическое ожидание этой величины и называется интенсивностью потока λ. Для простейшего потока λ является не зависящей от времени константой. В общем же случае нестационарного потока λ является функцией t.

Очень многие потоки событий можно считать простейшими или близкими к простейшим. *В дальнейшем все рассматриваемые потоки считаются простейшими.* Если известна интенсивность простейшего потока λ, то вероятность появления k событий на промежутке длительностью Δt определяется формулой Пуассона

$$P_k(\Delta t) = (\lambda \Delta t)^k \cdot \frac{e^{\lambda \Delta t}}{k!}.$$

Для простейшего потока с интенсивностью λ случайный интервал T между соседними событиями имеет показательное распределение с плотностью

$$f(t) = \lambda e^{-\lambda t} \quad , (t > 0).$$

Для случайных величин, распределенных по показательному закону, математическое ожидание m_T и среднее квадратическое отклонение σ_T удовлетворяют соотношениям

$$m_T = \sigma_T = \frac{1}{\lambda}.$$

В расчетах, связанных с потоками событий, удобно пользоваться *элементом вероятности*. Обозначим через P $(t,\ \Delta t)$ вероятность появления на промежутке времени $(t,\ t+\Delta t)$ хотя бы одного события. Тогда

$$P(t, \Delta t) = \lambda(t)\Delta t + o(\Delta t)$$

при $\Delta t \to 0$. Член $\lambda(t)\Delta t$ и называется элементом вероятности. Для простейшего потока λ=const и элемент вероятности равен $\lambda\Delta t$.

В дальнейшем нам часто придется рассматривать несколько простейших потоков вместе. В этом случае говорят о сумме потоков. Получается новый поток, интенсивность которого равна сумме интенсивностей составляющих потоков.

8.2. Марковские случайные процессы с дискретным множеством состояний и непрерывным временем

Пусть некоторая физическая система с течением времени меняет свое состояние (переходит из одного состояния в другое), причем заранее неизвестным, случайным образом. Тогда говорят, что в системе протекает случайный процесс. Этот случайный процесс называется *марковским*, если вероятностные характеристики процесса в будущем зависят от состояния системы в настоящий момент времени и не зависят от того, когда и как система пришла в это состояние. На практике марковские процессы в чистом виде обычно не встречаются. Однако очень часто приходится иметь дело с процессами, для которых влиянием “предыстории” можно пренебречь. При изучении таких процессов с успехом применяются марковские модели. В дальнейшем мы считаем, что все возможные состояния системы можно перенумеровать, и они составляют конечную или бесконечную последовательность S_1, S_2, …, S_n, …, причем переход системы из состояния в состояние происходит “скачком”, то есть мгновенно. Кроме того, мы будем считать, что моменты этих переходов не фиксированы заранее, а являются случайными, то есть переход из состояния в состояние может осуществляться, в принципе, в любой момент. Описанные процессы называются *Марковскими процессами с дискретными состояниями и непрерывным временем*. При анализе таких случайных процессов пользуются геометрической схемой — *графом состояний*. Состояния системы обозначаются прямоугольниками (или кругами, или даже точками), а возможные переходы из состояния в состояние — стрелками, соединяющими состояния.

Пример 8.1. Техническое устройство состоит из двух узлов, каждый из которых в случайный момент времени может отказать, после чего мгновенно начинается ремонт этого узла, продолжающийся случайное время. Возможные состояния системы можно перечислить:

S_0 — оба узла исправны;
S_1 — первый ремонтируется, второй исправен;
S_2 — второй ремонтируется, первый исправен;
S_3 — оба узла ремонтируются.

Построим граф состояний. Стрелка, направленная из S_0 в S_1, означает переход в момент отказа первого узла; стрелка, направленная обратно, из S_1 в S_0, — переход в момент окончания ремонта узла. Остальные стрелки объясняются аналогично. Стрелка из S_0 в S_3 отсутствует по той причине, что узлы предполагаются выходящими из строя независимо друг от друга и строго одновременный выход из строя обоих узлов практически невозможен.

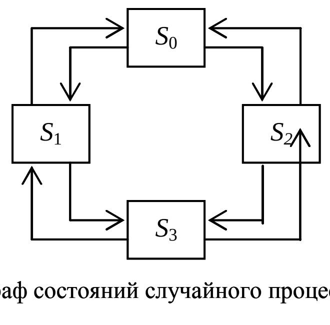


Рис. 8.1. Граф состояний случайного процесса примера 8.1

8.3. Уравнения Колмогорова для вероятностей состояний Финальные вероятности состояний

В теории массового обслуживания переходы системы из состояния в состояние происходят под действием каких-то потоков событий (поток вызовов, поток отказов, поток восстановлений и т. д.). Если все потоки, переводящие систему из состояния в состояние, являются простейшими, то процесс, протекающий в системе, будет марковским. Последнее вытекает из того, что простейший поток не обладает последействием, в нем “будущее” не зависит от “прошлого”.

В дальнейшем считается, что если система находится в состоянии S_i, на нее действует поток, переводящий ее по стрелке $S_i \to S_j$ графа состояний в состояние S_j. Этот поток считается простейшим с интенсивностью λ_{ij}. Если из состояния S_i есть несколько выходов в другие состояния, то на систему в состоянии S_i действует несколько потоков каждый со своей интенсивностью.

Как только появится первое событие потока, отвечающего стрелке $S_i \to S_j$, система переходит из состояния S_i в S_j. Для наглядности на графе состояний у каждой стрелки $S_i \to S_j$ проставляют интенсивность λ_{ij} соответствующего потока. Такой граф называют *размеченным*. Построим размеченный граф состояний примера 1.

Пример 8.2. Предположим, что среднее время ремонта узла не зависит от того, ремонтируется один узел или оба сразу. Это будет так, если ремонтом каждого узла занят отдельный специалист. Пусть система находится в состоянии S_0 и поток отказов первого узла имеет интенсивность λ_1. Этот поток и переводит систему из состояния S_0 в состояние S_1. Наоборот, из S_1 в S_0 систему переводит поток “окончание ремонта” первого узла. Его интенсивность μ_1 равна единице, деленной на среднее время ремонта первого узла. Обозначив через λ_2 и μ_2 интенсивности соответствующих потоков для второго узла, мы получим следующий размеченный граф состояний.

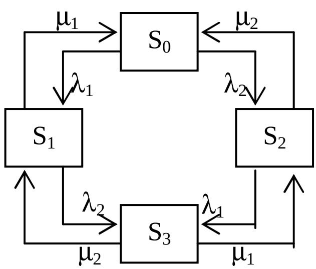


Рис. 8.2. Размеченный граф состояний примера 8.2

Имея размеченный граф состояний можно построить математическую модель случайного процесса в виде системы дифференциальных уравнений, которые называются уравнениями Колмогорова.

Назовем вероятностью i-го состояния вероятность $p_i(t)$ того, что в момент t система находится в состоянии S_i. Очевидно, что при каждом t справедливо равенство $\sum_{i=1}^{n} p_i(t) = 1$.

Рассмотрим функцию $p_i(t)$. Придадим t приращение Δt и найдем $p_i(t + \Delta t)$ — вероятность того, что в момент $t + \Delta t$ система будет находиться в состоянии S_i. Это может произойти при двух условиях: 1) в момент t система находилась в состоянии S_i и за время Δt из него не вышла; 2) в момент t система находилась в другом состоянии S_j и за время Δt перешла в S_i. При этом j — любой номер состояния, из которого в фиксированное состояние S_i идет стрелка графа состояний.

Обозначим через N_i^+ множество номеров тех состояний, из которых идет стрелка в состояние S_i, а через N_i^-— множество номеров тех состояний, в которые выходят стрелки графа из состояния S_i. Для нахождения $p_i(t + \Delta t)$ найдем вначале вероятность того, что за время Δt система оставалась в состоянии S_i. Суммарный поток событий, выводящих систему из состояния S_i, тоже будет простейшим и его интенсивность равна $\sum_{j \in N_i^-} \lambda_{ij}$. Вероятность того, что за время Δt система выйдет из состояния S_i с точностью до членов, стремящихся к нулю быстрее, чем Δt, равна $(\sum_{j \in N_i^-} \lambda_{ij})\Delta t$. Вероятность того, что за время Δt система останется в состоянии S_i равна: $1 - (\sum_{j \in N_i^-} \lambda_{ij})\Delta t$. Поэтому вероятность первого варианта равна $p_i(t)\left[1 - (\sum_{j \in N_i^-} \lambda_{ij})\Delta t\right]$. Найдем вероятность второго варианта. Если при некотором t система находилась в состоянии S_j, где $j \in N_i^+$, то вероятность перейти в состояние S_i очевидно равна $p_j(t)\lambda_{ji}\Delta t$. Поэтому вероятность второго варианта равна

$$\sum_{j \in N_i^+} p_i(t)\lambda_{ji}\Delta t + o(\Delta t).$$

Складывая вероятности обоих вариантов, получим

$$p_i(t+\Delta t) = p_i(t)\left[1-(\sum_{j\in N_i^-}\lambda_{ij})\Delta t\right]+(\sum_{j\in N_i^+} p_j(t)\lambda_{ji})\Delta t+\mathrm{o}(\Delta t)\,.$$

После преобразования имеем:

$$\frac{p_i(t+\Delta t)-p_i(t)}{\Delta t} = \sum_{j\in N_i^+}\lambda_{ji}p_j(t)-(\sum_{j\in N_i^-}\lambda_{ij})p_i(t)+\mathrm{o}(1)\,.$$

Переходя к приделу при $\Delta t \to 0$, получим:

$$\frac{dp_i}{dt} = \sum_{j\in N_i^+}\lambda_{ji}p_j(t)-(\sum_{j\in N_i^-}\lambda_{ij})p_i(t)\,.$$

Эти уравнения при i=1, 2, … , n образуют систему дифференциальных уравнений, которую называют *системой Колмогорова*. Число уравнений в этой системе равно числу возможных состояний процесса. Если таких состояний бесконечное число, то и неизвестных функций и уравнений бесконечное количество.

Для нахождения конкретного решения системы Колмогорова нужно задавать начальные условия. Если в начальный момент система находилась, например, в состоянии S_1, то в начальный момент (при $t = t_0$) $p_1(t_0) = 1$, $p_2(t_0) = 0, \ldots, p_n(t_0) = 0$. Существует единственное решение, удовлетворяющее этим условиям (а также условию $\sum_{i=1}^{n} p_i(t) = 1$).

При $t \to \infty$ очень часто все функции $p_i(t)$ независимо от начальных условий стремятся к некоторым пределам, которые называются *финальными вероятностями состояний.*

Теорема. **8.1**. *Если число состояний системы n конечно и из каждого состояния можно перейти в любое другое по стрелочкам графа, то финальные вероятности существуют.*

Если обозначить финальные вероятности через $p_1, p_2, \ldots, p_n$, подразумевая под ними уже константы, то для их определения будем иметь систему линейных алгебраических уравнений

$$(\sum_{j\in N_i^-}\lambda_{ij})p_i = \sum_{j\in N_i^+}\lambda_{ji}p_j\,,\quad i=1,2,\ldots,n.$$

Когда финальные вероятности существуют, эти уравнения зависимы и для нахождения финальных вероятностей к ним добавляют еще уравнение $p_1 + p_2 + \ldots + p_n = 1$.

При $t \to \infty$ в системе устанавливается предельный стационарный режим, в ходе которого система случайным образом меняет свои состояния, но их вероятности уже не зависят от времени. Финальную вероятность состояния S_i можно истолковать как среднее относительное время пребывания системы в этом состоянии.

Пример 8.3. Определим финальные вероятности состояний S_0, S_1, S_2, S_3 в примере 8.2 считая $\lambda_1 = 1$; $\lambda_2 = 2$; $\mu_1 = 2$; $\mu_2 = 3$. Используя граф состояний примера 8.2, получим для финальных вероятностей систему

$$\begin{cases} (\lambda_1 + \lambda_2) p_0 = \mu_1 p_1 + \mu_2 p_2, \\ (\lambda_2 + \mu_1) p_1 = \lambda_1 p_0 + \mu_2 p_3, \\ (\lambda_1 + \mu_2) p_2 = \lambda_2 p_0 + \mu_1 p_3, \\ (\mu_1 + \mu_2) p_3 = \lambda_2 p_1 + \lambda_1 p_2. \end{cases}$$

Подставляя данные, отбрасывая четвертое уравнение (которое является следствием остальных) и, учитывая, что сумма вероятностей равна 1, получим систему уравнений

$$\begin{cases} 3p_0 = 2p_1 + 3p_{2,} \\ 4p_1 = p_0 + 3p_{3,} \\ 4p_2 = 2p_0 + 2p_{3,} \end{cases}$$

$$p_0 + p_1 + p_2 + p_3 = 1.$$

Решая эту систему, получим: $p_0 = \dfrac{2}{5}$; $p_1 = \dfrac{1}{5}$; $p_2 = \dfrac{4}{15}$; $p_3 = \dfrac{2}{15}$.

То есть в предельном стационарном режиме система в среднем 40% времени будет проводить в состоянии S_0 (оба узла исправны), 20%-в состоянии S_1 (первый узел ремонтируется, второй работает), приблизительно 27% — в состоянии S_2 (второй узел ремонтируется, первый работает) и примерно 13% — в состоянии S_3 (оба узла ремонтируются).

8.4. Схема гибели и размножения

Марковский процесс называется процессом гибели и размножения, если его размеченный граф состояний имеет вид

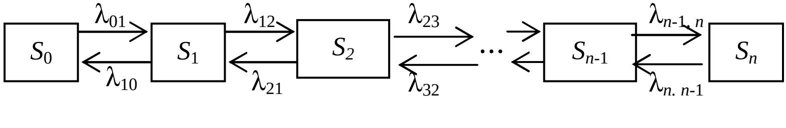


Рис. 8.3. Размеченный граф состояний процесса гибели и размножения

Термин “схема гибели и размножения” ведет начало от биологических задач, где подобной схемой описывается изменение численности популяций. Эта схема часто встречается на практике, в частности в теории массового обслуживания. Поэтому для нее важно вычислить финальные вероятности p_0, p_1, $p_2, \ldots$, p_n. Пользуясь графом состояний, составим и решим алгебраические уравнения для финальных вероятностей.

$$\begin{cases} p_0\lambda_{01} = \lambda_{10}\,p_{1,} \\ (\lambda_{12}+\lambda_{10})\,p_1 = \lambda_{01}\,p_0 + \lambda_{21}\,p_2\,, \\ (\lambda_{21}+\lambda_{23})\,p_2 = \lambda_{12}\,p_1 + \lambda_{32}\,p_3\,, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ (\lambda_{n-1,n-2}+\lambda_{n-1,n}\,)\,p_{n-1} = \lambda_{n-2,n-1}\,p_{n-2} + \lambda_{n,n-1}\,p_n\,, \\ \lambda_{n,n-1}\,p_n = \lambda_{n-1,n}\,p_{n-1}. \end{cases}$$

Из первого уравнения видно, что второе можно записать в виде $\lambda_{12}p_1 = \lambda_{21}\,p_2$, а в силу этого третье уравнение запишется так $\lambda_{23}p_2 = \lambda_{32}\,p_3$, и так далее. В итоге получаем, что финальные вероятности удовлетворяют уравнениям

$$\begin{cases} \lambda_{01}\,p_0 = \lambda_{10}\,p_{1,} \\ \lambda_{12}\,p_1 = \lambda_{21}\,p_{2,} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ \lambda_{k-1,k}\,p_{k-1} = \lambda_{k,k-1}\,p_{k,} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, \\ \lambda_{n-1,n}\,p_{n-1} = \lambda_{n,n-1}\,p_n\,. \end{cases}$$

Из этих равенств легко вычислить все вероятности p_k через p_0. Действительно $p_1 = \dfrac{\lambda_{01}}{\lambda_{10}}\,p_0$; $p_2 = \dfrac{\lambda_{12}}{\lambda_{21}}\,p_1 = \dfrac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}}\,p_0$; $p_3 = \dfrac{\lambda_{23}\lambda_{12}\lambda_{01}}{\lambda_{32}\lambda_{21}\lambda_{10}}\,p_0\,.$ Для любого k (от 1 до n)

$$p_k = \frac{\lambda_{k-1,k}\ldots\lambda_{12}\lambda_{01}}{\lambda_{k,k-1}\ldots\lambda_{21}\lambda_{10}}\,p_0\,.$$

В числителе последней формулы стоит произведение интенсивностей, стоящих у стрелок, ведущих слева направо (от S_0 до S_k), а в знаменателе — произведение всех интенсивностей, стоящих у стрелок, ведущих справа налево (от S_k до S_0). Подставляя выраженные через p_0 вероятности в равенство $p_0 + p_1 + \ldots + p_n = 1$, получим:

$$p_0\left(1+\frac{\lambda_{01}}{\lambda_{10}}+\frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}}+\ldots+\frac{\lambda_{n-1,n}\ldots\lambda_{12}\lambda_{01}}{\lambda_{n,n-1}\ldots\lambda_{21}\lambda_{10}}\right)=1\,.$$

Таким образом, получаем выражение для p_0

$$p_0=\left(1+\frac{\lambda_{01}}{\lambda_{10}}+\frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}}+\ldots+\frac{\lambda_{n-1,n}\ldots\lambda_{12}\lambda_{01}}{\lambda_{n,n-1}\ldots\lambda_{21}\lambda_{10}}\right)^{-1}.$$

8.5. Классификация систем массового обслуживания Основные задачи теории СМО

Одной из основных задач исследования операций является анализ своеобразных систем, называемых системами массового обслуживания (СМО). Примерами таких систем могут служить: телефонные станции, ремонтные мастерские, билетные кассы, справочные бюро, магазины, парикмахерские и т.д. Каждая СМО состоит из некоторого количества обслуживающих единиц, которые называют *каналами обслуживания*. Каналами могут быть линии связи, рабочие точки, кассиры, продавцы, лифты, автомашины и т.д. СМО могут быть одноканальными и многоканальными. Всякая СМО предназначена для обслуживания некоторого *потока заявок*, поступающих в какие-то случайные моменты времени. Обслуживание заявок продолжается какое-то, вообще говоря, случайное время $T_{об}$, после чего канал освобождается и готов к приему следующей заявки.

Процесс работы СМО представляет собой случайный процесс с дискретными состояниями и непрерывным временем; состояние СМО меняется скачком в моменты появления каких-то событий (прихода новой заявки, окончания обслуживания старой, выбывания из очереди заявки, которой надоело ждать).

СМО делятся на типы (или классы) по ряду признаков. Одним из таких признаков является количество обслуживающих каналов. Однако основное деление: *СМО с отказами* и *СМО с очередью*. В СМО с отказами заявка, поступившая в момент, когда каналы заняты, получает отказ, покидает СМО и в дальнейшем в процессе обслуживания не участвует. Примеры СМО с отказами встречаются в телефонии. Заявка на разговор, пришедшая в момент, когда все каналы заняты, получает отказ и покидает СМО не обслуженной. В СМО с очередью заявка, пришедшая в момент, когда все каналы заняты, не уходит, а становится в очередь и ожидает возможности быть обслуженной. На практике чаще встречаются (и имеют большее значение) СМО с очередью. Теория массового обслуживания даже имеет второе название: “теория очередей”. Существуют и другие признаки, по которым производят классификацию СМО, но мы их рассматривать не будем.

Основной задачей теории массового обслуживания является определение по заданным условиям работы СМО (число каналов, их производительность, правила работы, характер потока заявок) показателей эффективности СМО, которые характеризуют способность СМО справляться с потоком заявок. В качестве таких показателей (в зависимости от обстановки и целей исследования) могут применяться различные величины. Основными из этих величин являются:

A — абсолютная пропускная способность СМО, то есть среднее число заявок, обслуживаемых в единицу времени;

Q — относительная пропускная способность (средняя доля пришедших заявок, обслуживаемых системой);

$\bar{k}$ — среднее число занятых каналов;

$P_{отк}$ — вероятность отказа, т.е. того, что заявка покинет СМО не обслуженной;

$L_{сист}$ — среднее число заявок, находящихся в системе (обслуживаемых или стоящих в очереди);

$W_{сист}$ — среднее время пребывания заявки в СМО;

$L_{оч}$ — среднее число заявок в очереди;

$W_{оч}$ — среднее время пребывания заявки в очереди.

Отметим, что если входной поток заявок стационарен и имеет интенсивность λ, то последние четыре параметра удовлетворяют следующим формулам Литтла

$$W_{сист} = \frac{1}{\lambda} L_{сист}, \quad W_{оч} = \frac{1}{\lambda} L_{оч},$$

которые справедливы для любых СМО, для которых эти параметры имеют смысл. В некоторых случаях часть из перечисленных показателей могут не иметь смысла. С другой стороны, в конкретных СМО существенную роль могут играть не упомянутые выше специфические показатели эффективности.

8.6. Простейшие СМО и их характеристики

Ниже все потоки событий, переводящие СМО из состояния в состояние, будут считаться простейшими. В их числе будет и так называемый “поток обслуживаний”, под которым подразумевается поток заявок, обслуженных одним непрерывно занятым каналом. В этом потоке интервал между событиями имеет показательное распределение. Время обслуживания $T_{об}$ — показательно распределенная случайная величина. При этом среднее время обслуживания $\bar{t}_{об}$ определяет интенсивность потока обслуживаний $\mu = \dfrac{1}{\bar{t}_{об}}$.

1. *n-канальная СМО с отказами* (*задача Эрланга*)

Эта задача возникла из технических нужд телефонии и была решена в начале нашего века датским математиком Эрлангом. Задача ставится так.

Имеется n каналов, на которые поступает поток заявок с интенсивностью λ. Поток обслуживаний имеет интенсивность $\mu = (\bar{t}_{об})^{-1}$. Найти финальные вероятности состояний СМО, а также характеристики эффективности A, Q, $P_{отк}$, $\bar{k}$.

Состояние СМО будем определять по числу заявок, находящихся в системе (в данном случае оно совпадает с числом занятых каналов). Получаем следующие состояния:

S_0 — в СМО нет ни одной заявки;

S_1 — в СМО находится одна заявка (один канал занят остальные свободны);

…………………………………….

S_k — в СМО находится k заявок (k каналов заняты, остальные свободны);

…………………………………….

S_n — в СМО находится n заявок (все n каналов заняты).

Граф состояний СМО соответствует схеме гибели и размножения и после разметки имеет вид (рис. 8.4)

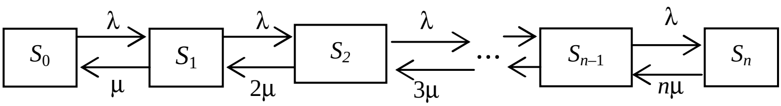


Рис. 8.4. Размеченный граф состояний n-канальной СМО с отказами

По уже готовым формулам схемы гибели и размножения получим

$$p_0 = \left(1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2\mu^2} + \ldots + \frac{\lambda^k}{k!\mu^k} + \ldots + \frac{\lambda^n}{n!\mu^n}\right)^{-1};$$

$$p_1 = \frac{\lambda}{\mu} p_0;\ p_2 = \frac{\lambda^2}{2!\mu^2} p_0;\ldots p_k = \frac{\lambda^k}{k!\mu^k} p_0;\ldots p_n = \frac{\lambda^n}{n!\mu^n} p_0.$$

Если ввести обозначение $\rho = \frac{\lambda}{\mu}$, то эти формулы можно переписать в виде:

$$p_0 = (1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + ... + \frac{\rho^n}{n!})^{-1}$$

$$p_1 = \rho p_0;\ p_2 = \frac{\rho^2}{2!} p_0; ... p_n = \frac{\rho^n}{n!} p_0 .$$

Величина $\rho = \frac{\lambda}{\mu}$ называется приведенной интенсивностью потока заявок. По финальным вероятностям вычислим характеристики эффективности СМО. Очевидно, что

$$P_{отк} = p_n = \frac{\rho^n}{n!} p_0$$

Относительная пропускная способность Q в данном случае это вероятность того, что заявка будет обслужена:

$$Q = 1 - p_{отк} = 1 - \frac{\rho^n}{n!} p_0$$

Абсолютную пропускную способность получим, умножая интенсивность потока заявок λ на Q:

$$A = \lambda Q = \lambda(1 - \frac{\rho^n}{n!} p_0)$$

Зная финальные вероятности нетрудно найти и $\bar{k}$ — среднее число занятых каналов как математическое ожидание числа занятых каналов:

$$\bar{k} = \rho(1 - \frac{\rho^n}{n!} p_0)$$

Пример 8.4. Имеется станция связи с тремя каналами (n = 3), интенсивность потока заявок λ = 1,5 (заявки в минуту); среднее время обслуживания одной заявки $\bar{t}$ = 2 (мин). Найти финальные вероятности состояний и характеристики эффективности СМО: A, Q, $p_{отк}$, $\bar{k}$ **.**

Очевидно ρ = 3. $p_0 = (1 + 3 + \frac{9}{2} + \frac{27}{6})^{-1} = \frac{1}{13}$;

$$p_1 = \rho p_0 = \frac{9}{13};\; p_2 = \frac{\rho^2}{2} p_0 = \frac{9}{26};\; p_3 = \frac{\rho^3}{3!} p_0 = \frac{9}{26};$$

$$A = 1.5(1 - \frac{9}{2}) \approx 0.981; Q \approx 0.654;\; p_{отк} = \frac{9}{26} \approx 0.346;$$

$$\bar{k} = 3(1 - \frac{9}{26}) \approx 1.96$$

Из полученных ответов видно, что СМО в значительной степени перегружена. Из трех каналов занято в среднем около двух, а из поступающих заявок 35% остаются не обслуженными.

2. *Одноканальная СМО с неограниченной очередью*

Пусть имеется одноканальная система с очередью, на которую не накладывается никакие ограничения (ни по длине очереди, ни по времени ожидания). На систему поступает поток заявок с интенсивностью λ; поток обслуживаний имеет интенсивность $\mu = (\bar{t}_{об})^{-1}$. Требуется, как и раньше, найти финальные вероятности и характеристики эффективности.

Состояние системы, как и раньше, определяются количеством заявок в системе. Они описываются так:

S_0 — канал свободен;

S_1 — канал занят (обслуживает заявку), очереди нет;

S_2 — канал занят, одна заявка стоит в очереди;

………………………………………………………………..

S_k — канал занят, $k-1$ заявка стоит в очереди.

………………………………………………………………..

Теоретически число состояний здесь бесконечно. Граф состояний имеет следующий вид

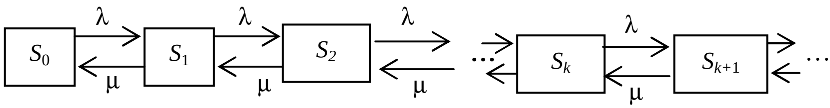


Рис. 8.5. Размеченный граф состояний одноканальной СМО с неограниченной очередью

Это схема гибели и размножения, но с бесконечным числом состояний. Можно доказать, что при $\rho = \lambda/\mu < 1$ финальные вероятности существуют, а при $\rho \geq 1$ очередь при $t \to \infty$ растет неограниченно. Формулы для финальных состояний можно считать такими же, как в обычной схеме гибели и размножения. Число слагаемых в формуле для p_0 будет бесконечным.

$$p_0 = \left[1+\frac{\lambda}{\mu}+\left(\frac{\lambda}{\mu}\right)^2+\ldots+\left(\frac{\lambda}{\mu}\right)^k+\ldots\right]^{-1} = \left(1+\rho+\rho^2+\ldots+\rho^k+\ldots\right)^{-1}$$

При $\rho < 1$ имеющийся здесь ряд сходится и его сумма равна $(1-\rho)^{-1}$. Поэтому $p_0 = 1-\rho$, $p_1 = \rho(1-\rho)$, $p_2 = \rho^2(1-\rho)$, … , $p_k = \rho^k(1-\rho)$, … . Вероятности p_0, p_1, p_2, … , p_k, … образуют геометрическую прогрессию со знаменателем ρ. Максимальная из этих вероятностей p_0 — вероятность того, что канал вообще свободен.

Найдем *среднее число заявок* в СМО $L_{сист}$. Случайная величина Z (число заявок в системе) имеет возможные значения $0, 1, 2, \ldots, k, \ldots$ с вероятностями $p_0, p_1, p_2, \ldots, p_k, \ldots$. Ее математическое ожидание:

$$L_{сист} = 0\cdot p_0 + 1\cdot p_1 + \ldots + k\cdot p_k + \ldots \; .$$

Подставляя сюда выражения для p_k, имеем:

$$L_{сист} = \sum_{k=1}^{\infty} k\rho^k(1-\rho) = \rho(1-\rho)\sum_{k=1}^{\infty} k\rho^{k-1} = \rho(1-\rho)\sum_{k=1}^{\infty}\frac{d\rho^k}{d\rho} =$$

$$= \rho(1-\rho)\frac{d}{d\rho}\sum_{k=1}^{\infty}\rho^k = \rho(1-\rho)\frac{d}{d\rho}\left(\frac{\rho}{1-\rho}\right) = \rho(1-\rho)\frac{1}{(1-\rho)^2} = \frac{\rho}{1-\rho}.$$

Таким образом, получаем: $L_{сист} = \dfrac{\rho}{1-\rho}$.

Формула Литтла сразу дает: $W_{сист} = \dfrac{\rho}{\lambda(1-\rho)}$.

Найдем среднее число заявок в очереди. Число заявок в очереди равно числу заявок в системе минус число заявок под обслуживанием. Значит (по правилу сложения математических ожиданий) среднее число заявок в очереди $L_{оч}$ равно среднему числу заявок в системе $L_{сист}$ минус среднее число заявок под обслуживанием. Число заявок под обслуживанием либо ноль (если канал свободен), либо единица (если канал занят). Математическое ожидание такой случайной величины равно вероятности того, что канал занят $P_{зан}$. Очевидно $P_{зан}=1-p_0=\rho$.

Такими образом, получаем $L_{оч} = L_{сист} - \rho = \dfrac{\rho}{1-\rho} - \rho$, или окончательно $L_{оч} = \dfrac{\rho^2}{1-\rho}$.

По формуле Литтла найдем среднее время пребывания заявки в очереди $W_{оч} = \dfrac{\rho^2}{\lambda(1-\rho)}$.

Абсолютную и относительную пропускные способности A и Q вычислять нет надобности, поскольку каждая заявка рано или поздно будет обслужена, A=λ, Q=1. Все характеристики эффективности найдены.

Пример 8.5. В порту имеется один причал для разгрузки судов. Интенсивность прибывающих судов равна 0,4 (судов в сутки). Среднее время разгрузки судна составляет 2 суток. Предполагается, что очередь судов, ожидающих разгрузки, может быть неограниченной длины. Определить показатели эффективности работы причала, а также вероятность того, что ожидают разгрузки не более чем 2 судна.

Очевидно, что $\rho = \lambda/\mu = \lambda \bar{t}_{об} = 0{,}4 \cdot 2 =$ 0,8. Поскольку $\rho < 1$, то финальные вероятности существуют. Вычислим некоторые из них. $p_0 = 1 - \rho = 0{,}2$. Далее $p_1 = \rho(1 - \rho) = 0{,}2 \cdot 0{,}8 = 0{,}16$; $p_2 = \rho^2(1-\rho) = 0{,}64 \cdot 0{,}2 = 0{,}128$; $p_3 = \rho^3(1-\rho) = 0{,}1024$. Вероятность того, что ожидают разгрузки не более двух судов, равна $P = p_1 + p_2 + p_3 = 0{,}16 + 0{,}128 + 0{,}1024 = 0{,}3904$.

Среднее количество судов в нашей СМО $L_{сист} = 0{,}8/0{,}2 = 4$. Среднее время пребывания судна в системе $W_{сист} = 4/0{,}4 = 10$ (суток). Средняя длина очереди $L_{оч} = \dfrac{\rho^2}{1-\rho}$ = 0,64/0,2 = 3,2, а по формуле Литтла среднее время пребывания заявки в очереди $W_{оч} = 3{,}2/0{,}4 = 8$ (суток).

Проведенные расчеты показывают, что эффективность разгрузки судов невысокая. Для повышения эффективности требуется уменьшить среднее время разгрузки судна или увеличить количество причалов (каналов обслуживания).

3. *n-канальная СМО с неограниченной очередью*

Пусть система имеет n каналов обслуживания, каждый из которых характеризуется потоком обслуживаний интенсивности μ. Поток заявок, как и в предыдущей задаче, имеет интенсивность λ и на очередь не накладывается никаких ограничений. Здесь мы имеем схему гибели и размножения с бесконечным количеством состояний, первые n+1 из которых определяются количеством занятых каналов и отсутствием очереди. Остальные — длиной очереди. Опуская подробности, приведем окончательные формулы для финальных вероятностей и некоторых показателей эффективности СМО. Введем обозначение $\rho = \lambda/(n\mu)$. Можно показать, что при $\rho < 1$ финальные вероятности существуют и справедливы равенства $p_k = p_0 \dfrac{n^k \rho^k}{k!}$, при $0 \le k \le n$; $p_k = p_0 \dfrac{n^n \rho^k}{n!}$, при $k \ge n + 1$, где

$$p_0 = \left[\frac{(n\rho)^n}{n!}(1-\rho)^{-1} + \sum_{k=0}^{n-1} \frac{(n\rho)^k}{k!} \right]^{-1}.$$

Для *среднего количества заявок в системе* справедлива формула

$$L_{сист} = n\rho + \frac{\rho p_n}{(1-\rho)^2}.$$

Из формул Литтла получим выражение для *среднего времени пребывания заявки в системе*

$$W_{сист} = \frac{1}{\lambda}\left(n\rho + \frac{\rho p_n}{(1-\rho)^2} \right).$$

Для средней длины очереди справедлива формула

$$L_{оч} = \frac{\rho p_n}{(1-\rho)^2}.$$

Среднее время пребывания в очереди равно

$$W_{оч} = \frac{p_n}{n\mu(1-\rho)^2}.$$

Пример 8.6. Порт имеет три причала для разгрузки судов. Среднее время разгрузки одного судна у каждого из причалов — 0,5 суток. На рейд порта пребывает в среднем 4 судна в сутки. При наличии свободного причала судно сразу становится под разгрузку. В противном случае оно ждет своей очереди на рейде. Определить среднее количество судов, ждущих разгрузки, а также среднее время ожидания в очереди.

Интенсивность потока заявок $\lambda = 4$, а для потока обслуживаний каждого из причалов $\mu = 1/0{,}5 = 2$. Поэтому $\rho = \lambda/(n\mu) = 4/6 = 2/3 < 1$. Вычислим финальные вероятности.

$$p_0 = \left[\frac{(3\cdot 2/3)^3}{3!}(1-\frac{2}{3})^{-1} + \sum_{k=0}^{2}\frac{(3\cdot 2/3)^k}{k!}\right]^{-1} = \frac{1}{9};$$

$$p_1 = \frac{1}{9}\cdot\frac{3\cdot 2/3}{1!} = \frac{2}{9};\ p_2 = \frac{1}{9}\cdot\frac{3^2(2/3)^2}{2!} = \frac{2}{9};\ p_3 = \frac{1}{9}\cdot\frac{3^3(2/3)^3}{3!} = \frac{4}{27};\ \ldots$$

Средняя длина очереди $L_{оч} = \dfrac{(2/3)p_3}{(1-(2/3))^2} = \dfrac{8}{9}$. Среднее время ожидания $W_{оч} = \dfrac{p_3}{3\cdot 2\cdot(1-(2/3))^2} = \dfrac{2}{9}$ суток. Таким образом, рассматриваемая СМО недогружена. В среднем в очереди будет находиться менее одного судна, а время ожидания составляет не многим более пятой части суток.

4. *СМО с ограниченной очередью*

СМО с ограниченной очередью отличается от рассмотренных выше систем тем, что длина очереди не может превышать некоторое заданное число m. Если заявка поступает в систему в момент, когда все места в очереди заняты, она покидает систему, т.е. получает отказ.

Определение предельных вероятностей и показателей эффективности таких СМО производится с помощью того же подхода, что и выше, с той лишь разницей, что суммировать нужно не бесконечную прогрессию (как, например, в пункте 2)), а конечную. Весь анализ такой одноканальной системы производится на основе графа рис. 8.5. Среднее время пребывания заявки в очереди и в системе определяется по формулам Литтла.

Пример 8.7. При условиях примера 8.5 определить показатели эффективности работы причала, если длина очереди не может превосходить трех судов.

Здесь $m = 3$ Найдем финальные вероятности состояний.

$$p_0 = (1 + \rho + \rho^2 + \rho^3 + \rho^4)^{-1} = (1-\rho)/(1-\rho^5) = 0{,}2/(1\text{-}0{,}8^5) \approx 0{,}297$$

$$p_1 = \rho\, p_0 \approx 0{,}2376,\; p_2 = \rho^2 p_0 \approx 0{,}1901,$$
$$p_3 = \rho^3 p_0 \approx 0{,}1521,\; p_4 = \rho^4 p_0 \approx 0{,}1216.$$

Вероятность того, что пришедшее судно покинет причал без разгрузки равно $P_{от\ к} \approx p_4 = 0{,}1216$. Относительная пропускная способность причала $Q = 1\text{-}P_{отк} \approx 1\text{-}0{,}1216 = 0{,}8784$. Абсолютная пропускная способность $A = \lambda\cdot Q \approx 0{,}4\cdot 0{,}8784 = 0{,}3514$. То есть в среднем у причала разгружается приблизительно 0,35 судна в сутки.

Среднее количество судов в СМО можно вычислить как математическое ожидание

$$L_{сист} = \sum_{k=1}^{4} k\rho^k p_0 = \rho p_0 \sum_{k=1}^{4} k\rho^{k-1} = \rho p_0 \sum_{k=1}^{4} \frac{d\rho^k}{d\rho} = \rho p_0 \frac{d}{d\rho} \sum_{k=1}^{4} \rho^k =$$
$$= \rho p_0 \frac{d}{d\rho}\left(\frac{1-\rho^5}{1-\rho}\right) = \rho p_0 \frac{1-5\rho^4+4\rho^5}{(1-\rho)^2}$$
$$= 0{,}8 \cdot 0{,}297(1-5\cdot 0{,}8^4 + 4\cdot 0{,}8^5)/0{,}2^2 \approx 1{,}56$$

Среднее время пребывания судна в системе по формулам Литтла равно $W_{сист} \approx 1{,}56/0{,}4 = 3{,}9$ (суток). Средняя длина очереди $L_{оч} = L_{сист} - \rho \approx 1{,}56 - 0{,}8 = 0{,}76$. Наконец среднее время пребывания в очереди $W_{оч} \approx 0{,}76/0{,}4 = 1{,}9$ (суток).

5. ***Замкнутая СМО с одним каналом и m источниками заявок***

Один рабочий обслуживает m станков, каждый из которых время от времени требует наладки (исправления). Интенсивность потока требований каждого работающего станка равна λ. Если станок вышел из строя в момент, когда рабочий свободен, он сразу же поступает на обслуживание. Если станок вышел из строя в момент, когда рабочий занят, он становится в очередь и ждет, пока рабочий освободится. Среднее время наладки $\bar{t}_{об} = \frac{1}{\mu}$. Интенсивность потока заявок, поступающих к рабочему, зависит от того, сколько станков работает. Если работает k станков, она равна $k\cdot\lambda$. Требуется найти финальные вероятности состояний, среднее число работающих станков и вероятность того, рабочий будет занят. В этой СМО финальные вероятности будут существовать при любых значениях λ и μ, так как число состояний системы конечно. Эти состояния описываются так:

S_0 — все станки работают;

S_1 — один станок под наладкой, очереди нет;
S_2 — один станок обслуживается, один стоит в очереди;
……………………………………………………………
S_k — в очереди $k-1$ станок;
…………………………………………………………….
S_m — в очереди $m-1$ станок.
Размеченный граф состояний имеет вид

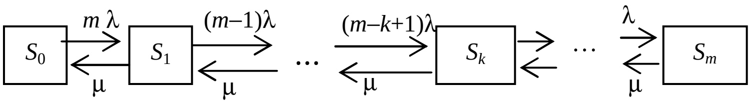


Рис. 8.6. Размеченный граф состояний замкнутой СМО

Это обычная схема гибели и размножения. Поэтому
$p_1 = m\rho \cdot p_0$, $p_2 = m(m-1)\rho^2 \cdot p_0$, … , $p_k = m(m-1)\ \dots\ (m-k+1)\rho^k \cdot p_0$, … , $p_m = m!\rho^m\, p_0$, где $\rho = \dfrac{\lambda}{\mu}$. Для p_0 получаем формулу

$$p_0 = (1 + m\rho + m(m-1)\rho^2 + \dots + m(m-1) \dots \cdot (m-k+1)\rho^k + \dots + m!\rho^m)^{-1}.$$

Среднее число работающих станков является математическим ожиданием случайной величины N с возможными значениями m, m–1,…, 0 и вероятностями этих значений $p_0, p_1, p_2, \dots, p_m$.

$$\overline{N} = mp_0 + (m-1)p_1 + \dots + p_{m-1}.$$

Вероятность того, что рабочий будет занят, равна $p_{зан} = 1 - p_0$.

Пример 8.8. Среднее время безотказной работы станка равно 10 часам. Рабочий обслуживает 8 станков. Среднее время наладки станка 20 минут (1/3 часа). Найти среднее число работающих станков и вероятность занятости рабочего.

Интенсивность требований от каждого станка равна $\lambda = 0{,}1$. Интенсивность потока восстановлений станков $\mu = 3$ (станка в час). Далее $\rho = 1/30$. Вычислим финальные вероятности состояний.

$$p_0 = (1 + \frac{8}{30} + \frac{8\cdot 7}{30^2} + \frac{8\cdot 7\cdot 6}{30^3} + \frac{8\cdot 7\cdot 6\cdot 5}{30^4} + \frac{8\cdot 7\cdot 6\cdot 5\cdot 4}{30^5} +$$

$$+ \frac{8\cdot 7\cdot 6\cdot 5\cdot 4\cdot 3}{30^6} + \frac{8\cdot 7\cdot 6\cdot 5\cdot 4\cdot 3\cdot 2}{30^7} + \frac{8!}{30^8})^{-1} \approx 0.744\,;$$

$$p_1 = \frac{8}{30} p_0 \approx 0.198;\ p_2 = \frac{8 \cdot 7}{30^2} p_0 \approx 0.046;\ p_3 = 9.26 \cdot 10^{-3};$$

$$p_4 = 1.54 \cdot 10^{-3};\ p_5 = 2.06 \cdot 10^{-4};\ p_6 = 2.06 \cdot 10^{-5};\ p_7 = 1.37 \cdot 10^{-6};$$

$$p_8 = 4.57 \cdot 10^{-8}.$$

Вероятность того, что рабочий будет занят: $p_{зан} = 1 - p_0 \approx 0.256$.

Среднее число работающих станков:

$$\overline{N} = 8p_0 + 7p_1 + 6p_2 + 5p_3 + 4p_4 + 3p_5 + 2p_6 + p_7 \approx 7.668.$$

Таким образом, в среднем почти все станки работают, а рабочий занят в среднем 26% рабочего времени. Рабочий явно недогружен. Количество закрепленных за ним станков можно увеличить. Однако это приведет к уменьшению среднего относительного количества работающих станков.

В заключение отметим, что на практике часто появляются *СМО с ограниченным временем ожидания*. Их часто называют СМО с «нетерпеливыми» заявками. Такие заявки могут уйти из очереди, если время ожидания превышает некоторую величину. Такого рода заявки возникают в технологических системах, в которых задержка с началом обслуживания приводит к потере качества продукции, в системах оперативного управления, когда срочные сообщения теряют ценность, если они не поступают на обслуживание в течение определенного времени и других подобных системах. В простейших математических моделях таких систем предполагается, что заявка может находиться в очереди случайное время. Это время считается случайной величиной, распределенной по показательному закону с параметром ν. Поэтому обычно в схеме процесса гибели и размножения вводится еще один простейший поток, а именно поток ухода заявок из очереди, имеющий интенсивность ν. Соответствующие показатели эффективности СМО с ограниченным временем ожидания могут быть найдены на базе результатов, полученных для процесса гибели и размножения.

Контрольные вопросы и задачи для самостоятельного решения

1. Дайте определение простейшего потока событий. Как связан простейший поток событий с законом Пуассона и показательным распределением?
2. Какой случайный процесс называется Марковским? Марковские процессы с дискретным множеством состояний и непрерывным временем. Что такое граф состояний такого процесса?

3. Поток событий, переводящий процесс из одного состояния в другое. Что такое размеченный граф состояний?
4. Уравнения Колмогорова для вероятностей состояний. Что такое финальные вероятности состояний? Система уравнений для финальных вероятностей состояний.
5. Какой Марковский процесс называется процессом гибели и размножения? Определение финальных вероятностей состояний такого процесса.
6. Системы массового обслуживания и их классификация. Основные показатели эффективности СМО. Формула Литтла.
7. Опишите n-канальную СМО с отказами. Каковы ее граф состояний и основные характеристики?
8. Финальные вероятности и характеристики эффективности одноканальной СМО с неограниченной очередью.
9. Основные характеристики n-канальной СМО с неограниченной очередью.
10. Основные характеристики СМО с ограниченной очередью.
11. Замкнутая СМО с одним каналом обслуживания и m источниками заявок.

Выполнить следующие упражнения.

8.1. Построить граф состояний системы, состоящей из двух автоматов по продаже газированной воды, каждый из которых в случайный момент времени может оказаться занятым или свободным.

8.2. Построить граф состояний системы, представляющей собой лампочку, которая в случайный момент времени может быть включена, выключена или выведена из строя (может перегореть).

8.3. Найти финальные вероятности случайных процессов, имеющих размеченные графы состояний, изображенные на рис. 8.7.

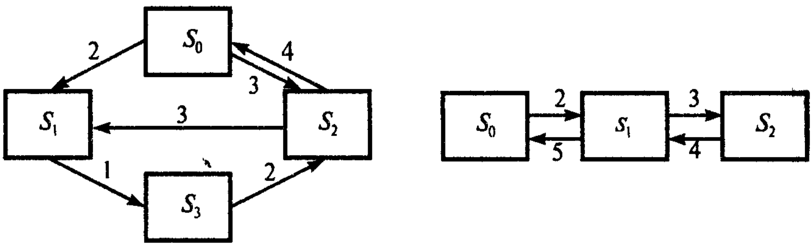


Рис. 8.7. Размеченные графы состояний.

8.4. Рассматривается круглосуточная работа пункта профилактического осмотра автомобилей с одним каналом обслуживания (одной группой проведения осмотра). На осмотр и выявление дефектов каждого автомобиля затрачивается в среднем 0,5 часа. На осмотр поступает в среднем 36 автомашин в сутки. Потоки заявок и обслуживаний простейшие. Если автомобиль, прибывший на пункт осмотра, не застает свободных каналов обслуживания, то он покидает пункт осмотра. Определить вероятности состояний и характеристики обслуживания пункта осмотра.

8.5. Решить задачу 8.4 для случая n=4 каналов обслуживания (групп проведения осмотра). Определить наименьшее число каналов, при котором относительная пропускная способность пункта осмотра будет не менее 0,9.

8.6. Анализируется работа междугородного переговорного пункта в небольшом городке. Пункт имеет один телефонный аппарат для переговоров. Средняя длительность переговоров (с учетом вызова абонента в другом городе) составляет 5 минут. В среднем в сутки поступает 240 заявок на переговоры. Никаких ограничений на длину очереди нет. Потоки заявок и обслуживаний простейшие. Определить вероятности состояний и характеристики обслуживания переговорного пункта.

8.7. Решить задачу 8.6 для случая, когда на переговорном пункте имеются три телефонных аппарата.

8.8. Решить задачу 8.4 при условии, что автомобиль покидает пункт обслуживания лишь в случае, когда длина очереди на осмотр превысит 5 автомобилей.

8.9. Решить задачу 8.6 при условии, что длина очереди не должна превышать 60 человек.

8.10. На уборке урожая работают 15 комбайнов. Среднее время безотказной работы комбайна 12 часов. Для обслуживания комбайнов выделена ремонтная бригада, приступающая к ремонту сразу после выхода комбайна из строя и устраняющая поломку в среднем за 1 час. Считая потоки заявок и обслуживаний простейшими определить среднее количество работающих комбайнов и вероятность занятости ремонтной бригады.

9. ПОТОКИ В СЕТЯХ

9.1. Основные понятия

В этой главе будем использовать следующие понятия и обозначения.

V — конечное множество, которое будем называть *множеством вершин*.

(x, y) — *упорядоченная пара*, первый элемент которой x, а второй — y.

$V^2 = V \times V$ — *квадрат множества* V (декартово произведение множества V на себя). Квадрат множества V представляет собой множество всех упорядоченных пар, которые можно составить из элементов множества V, т.е. $V^2 = \{(x, y) \mid x \in V \text{ и } y \in V\}$.

$I = \{(x, x) \mid x \in V\}$ — множество всех упорядоченных пар, в которых первый и второй элементы равны.

Определим множество E как подмножество V^2, не содержащее упорядоченных пар с одинаковыми элементами, т.е. $E \subseteq V^2 / I$. Множество E назовем *множеством дуг*. Если дуга $e = (x, y)$ принадлежит множеству E, то вершину x будем называть *начальной вершиной* дуги, а вершину y — *конечной вершиной* дуги. Дуга (x, y) имеет направление от вершины x к вершине y.

Пару $G = (V, E)$ назовем *ориентированным графом (орграфом) без петель*. Орграф можно представить графически в виде *диаграммы*: каждая вершина представляется точкой или окружностью, а каждая дуга — стрелочкой, направленной от начальной вершины к конечной.

Орграф $G = (V, E)$, $V = \{v_1, v_2, v_3, v_4\}$, $E = \{(v_1, v_3), (v_3, v_2), (v_2, v_4), (v_3, v_4)\}$ представлен диаграммой на рис. 9.1. Один и тот же орграф может быть представлен различными диаграммами, так как расположение вершин не имеет значения.

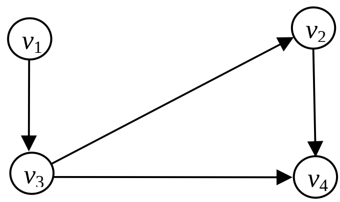


Рис. 9.1. Диаграмма орграфа

Количество дуг, входящих в вершину v_i назовем полустепенью захода вершины v_i , а количество дуг, выходящих из вершины v_i — полустепенью исхода вершины v_i.

Последовательность вершин орграфа назовем *цепью*, если любой паре рядом стоящих вершин v_i и v_{i+1}, соответствует дуга (v_i, v_{i+1}), либо (v_{i+1}, v_i). Первую вершину последовательности назовем *начальной вершиной цепи*, а последнюю — *конечной вершиной цепи*. Будем говорить, что цепь *соединяет* начальную вершину с конечной. Дугу вида (v_i, v_{i+1}) назовем *прямой*, а дугу вида (v_{i+1}, v_i) — *обратной*. Цепь, которая состоит только из прямых дуг, назовем *путем*. *Длина цепи* определяется количеством входящих в нее дуг (на единицу меньше количества входящих в нее вершин).

На множестве E определим целочисленную функцию $c(e) \geq 0$. Значение $c(e)$ будем называть *пропускной способностью* дуги e. Пару $N = (G, C)$ назовем *сетью*. Сеть можно представить диаграммой, на которой около каждой дуги указывается ее пропускная способность.

Сеть можно использовать в качестве математической модели систем, между элементами которой могут перемещаться или перетекать некоторые объекты. Примером такой системы может служить сеть автодорог, соединяющих населенные пункты, по которым движутся автомобили. Максимальное количество автомобилей, которые могут проехать по участку дороги за единицу времени, определяет пропускную способность этого участка (зависит от ширины проезжей части, качества дорожного покрытия, действующих ограничений скорости движения и т.д.). Другим примером может служить сеть трубопроводов, соединяющих нефтепромыслы с нефтеперерабатывающими заводами. Трубопроводы могут соединяться и разветвляться в промежуточных пунктах. Пропускная способность отрезка трубопровода определяется максимальным количеством нефти, которое может быть перекачено по этому отрезку за единицу времени (зависит от диаметра трубы, мощности нагнетающего насоса и т.д.).

В сети выделяют вершину s, называемую *источником*, ей соответствует элемент системы, из которого начинается перемещение объектов, и вершину t, называемую *стоком*, ей соответствует элемент системы, в котором заканчивается перемещение объектов. Объекты, которые перемещаются или перетекают из источника в сток, будем называть *единицами потока*. Функция $f(x, y)$, определенная на множестве дуг сети называется *потоком*, если выполняются следующие условия:

1) $\forall(x, y) \in E;\ 0 \le f(x, y) \le c(x, y)$ (9.1)

2) $\forall x \in V,\ x \ne s,\ x \ne t;\quad \sum\limits_{y \in V} f(x, y) - \sum\limits_{y \in V} f(y, x) = 0$ (9.2)

3) $\sum\limits_{y \in V} f(s, y) - \sum\limits_{y \in V} f(y, s) = v$ (9.3)

4) $\sum\limits_{y \in V} f(y, t) - \sum\limits_{y \in V} f(t, y) = v$ (9.4)

Значение $f(x, y)$ определяет количество единиц потока, проходящей по дуге (x, y) и называется *потоком в данной дуге*.

Первое условие для каждой дуги (x, y) ограничивает число единиц потока, проходящих по дуге (x, y), пропускной способностью этой дуги.

Второе условие запрещает утечки потока в промежуточных вершинах (в вершинах, отличных от источника и стока), то есть для каждой промежуточной вершины число приходящих единиц потока в эту вершину должно быть равно числу единиц потока, выходящих из нее.

Третье и четвертое условие задает баланс единиц потока, выходящих из источника и приходящих в сток: сколько единиц потока вышло из источника, столько же единиц потока должно прийти в сток.

Суммарное число единиц потока, выходящих из источника или входящих в сток (в третьем и четвертом условиях обозначено через v), называется *величиной потока*.

Дуги графа можно отнести к трем различным категориям:

1) дуги, в которых поток не может изменяться (например, если пропускная способность равна нулю, или по каким либо техническим причинам);

2) дуги, в которых поток может увеличиваться (если поток меньше пропускной способности);

3) дуги, в которых поток может уменьшаться (если поток больше нуля).

Множество дуг, относящихся к первой категории, обозначим через N. Дуги второй категории назовем *увеличивающими* и множество этих дуг обозначим через I. Дуги третьей категории назовем *уменьшающими* и множество этих дуг обозначим через R. Очевидно, любая дуга графа принадлежит хотя бы одному из трех этих множеств, а некоторые из них могут принадлежать как множеству I, так и множеству R. Максимальную величину, на которую может быть увеличен поток в дуге $(x, y) \in I$, обозначим через $i(x, y)$ и назовем *увеличением на дуге* (x, y), а максимальную величину, на которую может быть уменьшен поток в дуге $(x, y) \in R$ — через $r(x, y)$ и назовем *уменьшем на дуге* (x, y). Очевидно, что $i(x, y) = c(x, y) — f(x, y)$, а $r(x, y) = f(x, y)$.

Дугу, в которой поток равен пропускной способности, назовем *насыщенной*, а дугу, в которой поток равен нулю — *пустой*.

9.2. Поиск увеличивающей цепи

Пусть заданы сеть $N = (G, C)$ и поток F. Предположим, что мы хотим переслать дополнительное количество единиц из источника в сток. Если это в принципе возможно сделать, то одним из способов решения этой задачи является отыскание пути P_{st} от источника к стоку, состоящего только из увеличивающих дуг. По этому пути дополнительно можно переслать самое большее k_i единиц потока, не превышающее минимальное увеличение на дугах этого пути, т. е. $k_i = \min \{ i(x, y) \mid (x, y) \in P_{st} \}$. Если поток на каждой дуге пути P_{st} увеличить на k_i единиц, то на этом пути хотя бы одна из дуг станет насыщенной и величина потока в сети увеличится на k_i.

Второй способ решения задачи заключается в следующем. Некоторые единицы потока, которые дошли по сети из истока в сток, могут возвращаться обратно в исток, если существует путь P_{ts} от стока к истоку. Если количество единиц, проходящих по этому пути, уменьшить, то общее количество единиц потока, переданных от истока к стоку, увеличится. Поток на пути P_{ts} можно уменьшить, если этот путь состоит только из уменьшающих дуг. Уменьшить поток можно самое большее на k_r единиц потока, не превышающее минимальное увеличение на дугах этого пути, т.е. $k_r = \min \{ r(x, y) \mid (x, y) \in P_{ts} \}$. Если поток на каждой дуге пути P_{ts} уменьшить на k_r единиц, то на этом пути хотя бы одна из дуг станет пустой и величина потока в сети увеличится на k_r.

Рассмотренные выше способы решения задачи показывают, что для увеличения потока в сети, необходимо увеличивать поток на прямых дугах (первый способ) и уменьшать поток на обратных (второй способ). Для того, что бы скомбинировать эти два способа увеличения потока, необходимо найти цепь, соединяющую исток со стоком, в которой все прямые дуги являются увеличивающими, а все обратные — уменьшающими. Такая цепь называется *увеличивающей*. Максимальное количество единиц потока, которые можно дополнительно переслать из источника в сток, изменяя потоки на дугах увеличивающей цепи, определяется как минимум из двух величин:

$\min \{ i(x, y) \mid (x, y)$ — прямая дуга увеличивающей цепи $\}$

$\min \{ r(x, y) \mid (x, y)$ — обратная дуга увеличивающей цепи $\}$.

Минимальная из этих двух величин называется *максимальным увеличением потока* по соответствующей цепи. Для того, чтобы максимально увеличить поток по увеличивающей цепи, нужно на каждой прямой дуге увеличить поток на максимальное увеличение потока, а на каждой обратной дуге — уменьшить на максимальное увеличение потока. В результате хотя бы одна из дуг цепи станет насыщенной или пустой.

Итак, для того, что бы переслать дополнительное количество единиц потока из источника в сток, необходимо найти увеличивающую цепь. Идея алгоритма поиска увеличивающей цепи состоит в окрашивании дуг, по которым возможно увеличить поток в сети. Окрашиваются так же и вершины, до которых существуют увеличивающие цепи.

Алгоритм поиска увеличивающей цепи

1. Определить множества дуг N, I и R. Дуги множества N из дальнейшего рассмотрения исключить. Окрасить вершину s и включить ее в множество M окрашенных, но еще не обработанных алгоритмом вершин.

2. Пока вершина t не окрашена и множество M не пусто, взять произвольную вершину x из множества M и для каждой неокрашенной вершины y выполнить:

1) если $(x, y) \in I$, то окрасить дугу (x, y), вершину y окрасить и включить ее в множество M;

2) если $(y, x) \in R$, то окрасить дугу (y, x), вершину y окрасить и включить ее в множество M.

3. Если вершина t окрашена, то выбрать окрашенные дуги, образующие увеличивающую цепь, иначе увеличивающей цепи не существует. Конец алгоритма.

Пример 9.1. Найти увеличивающую цепь в сети (рис. 9.2) и максимально увеличить по ней поток. Вершину 1 считать истоком, вершину 7 — стоком. На дугах сети первое число обозначает пропускную способность дуги, второе — поток в дуге.

Определяем в данной сети множества дуг N, I и R:

$N = \varnothing$;

$I = \{(1,6), (2,3), (2,4), (2,5), (3,4), (4,7), (5,1), (5,7), (6,3), (7,6)\}$;

$R = \{(1,2), (1,3), (2,3), (2,4),(3,7), (4,7)\}$.

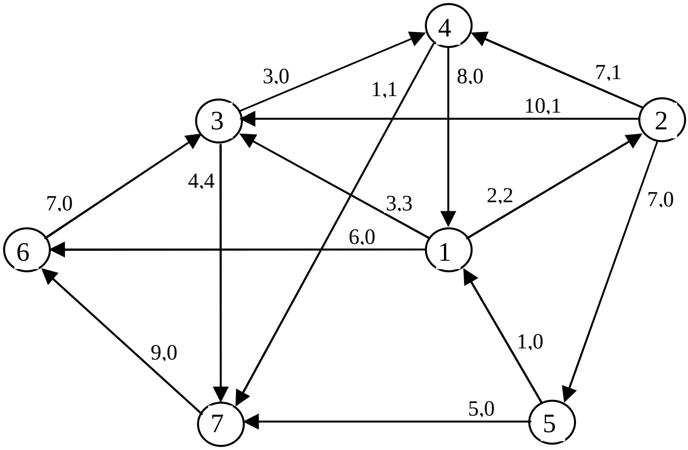


Рис. 9.2. Диаграмма сети

Окрашиваем вершину 1 и включаем ее в множество M, $M = \{1\}$.

Берем из множества M вершину 1 (множество M становится пустым) и окрашиваем дугу (1,6), вершину 6 и вершину 6 включаем в множество M, $M = \{6\}$. Окрасить больше ничего не можем. Результат окраски представлен на рис. 9.3.

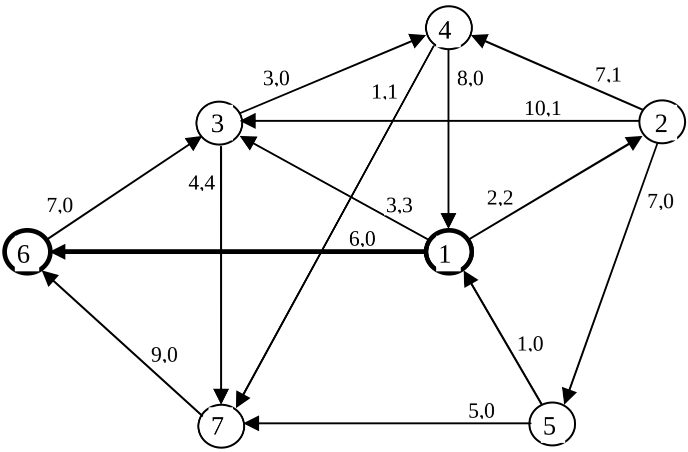


Рис. 9.3. Диаграмма сети

Множество M не пусто, берем из него вершину 6 и окрашиваем дугу (6,3), вершину 3 и вершину 3 включаем в множество M, $M = \{3\}$. Окрасить больше ничего не можем. Результат окраски представлен на рис. 9.4.

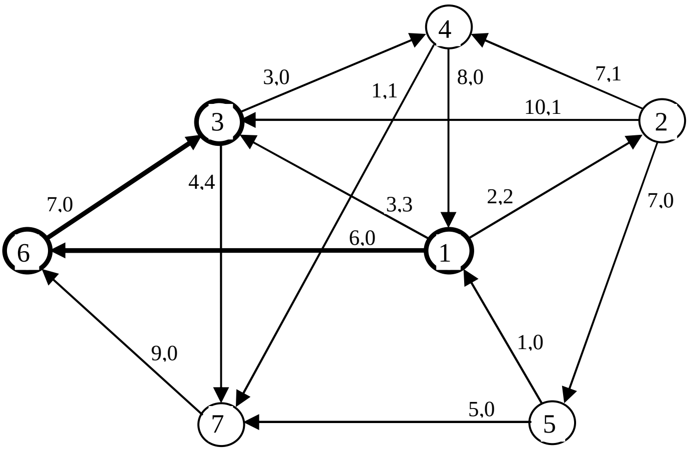


Рис. 9.4. Диаграмма сети

Множество $M = \{3\}$ не пусто, берем из него вершину 3 и окрашиваем дуги (2,3), (3,4), вершины 2 и 4 и вершины 2 и 4 включаем в множество M, $M = \{2,4\}$. Окрасить больше ничего не можем. Результат окраски представлен на рис. 9.5.

Множество $M = \{2,4\}$ не пусто, берем из него вершину 2 и окрашиваем дугу (2,5), вершину 5 и вершину 5 включаем в множество M, $M = \{4,5\}$. Окрасить больше ничего не можем. Результат окраски представлен на рис. 9.6.

Множество $M = \{4,5\}$ не пусто, берем из него вершину 5 и окрашиваем дугу (5,7) и вершину 7. Вершина 7 (сток) окрашена. Результат окраски представлен на рис. 9.7. Увеличивающую цепь образуют дуги (1,6), (6,3), (2,3), (2,5), (5,7). В этой цепи дуга (2,3) обратная, остальные — прямые. Максимальное увеличение потока по этой цепи составит одну единицу. После увеличения потока на одну единицу (рис. 9.8) дуга (2,3) станет пустой.

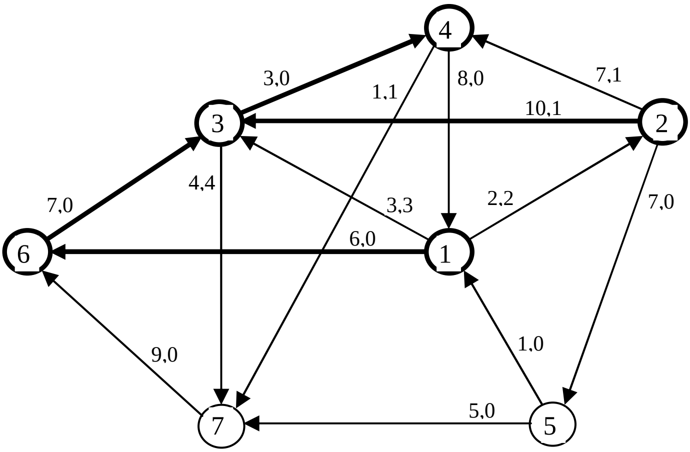


Рис. 9.5. Диаграмма сети

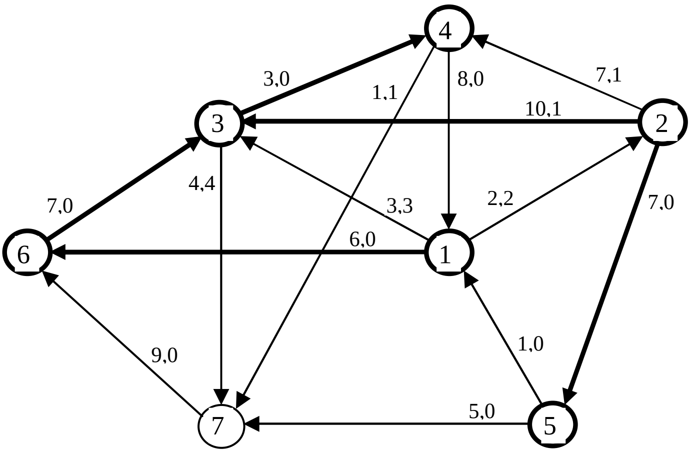


Рис. 9.6. Диаграмма сети

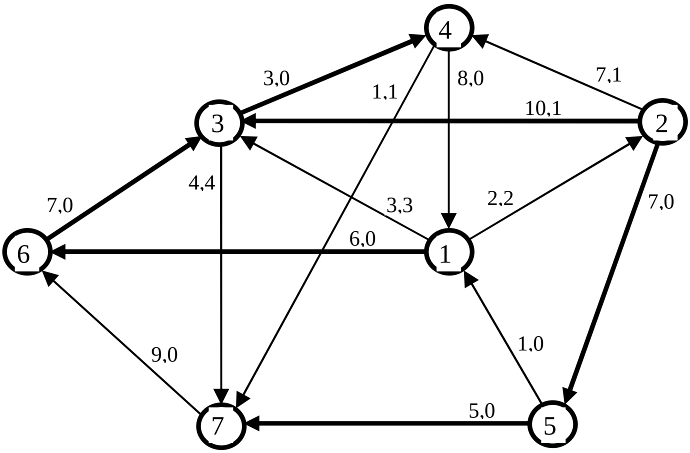


Рис. 9.7. Диаграмма сети

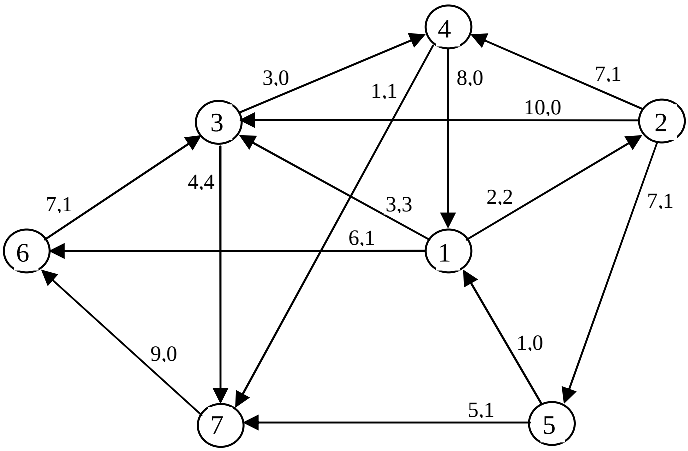


Рис. 9.8. Диаграмма сети

9.3. Задача о максимальном потоке

Напомним, что потоком в сети $N = (G, C)$ называется функция F, определенная на множестве дуг сети, если выполняются условия (9.1) — (9.4) (см. выше). Значение v в условиях 9.3 и 9.4 называется величиной потока. Задача о максимальном потоке заключается в поиске потока, величина v которого максимальна, т. е. в максимизации величины v при соблюдении ограничений (9.1) — (9.4). Как видно, эта задача является задачей линейного программирования и может быть решена с помощью симплекс-метода. Здесь же мы рассмотрим более простой и естественный подход к решению рассматриваемой задачи, предложенный Фордом и Фалкерсоном.

Идея алгоритма Форда и Фалкерсона очень проста. Выбирается некоторый начальный поток (можно взять нулевой поток) и с помощью алгоритма поиска увеличивающей цепи выполняется поиск увеличивающей цепи. Если цепь найдена, то поток вдоль найденной цепи увеличивается на максимально возможную величину. Затем выполняется поиск новой увеличивающей цепи и т. д. Если на некоторой итерации увеличивающую цепь найти не удается, выполнение алгоритма заканчивается: текущий поток является максимальным.

Пример 9.2. Найти максимальный поток в сети (рис. 9.9). Вершину 1 считать истоком, вершину 7 — стоком. На дугах сети указана пропускная способность.

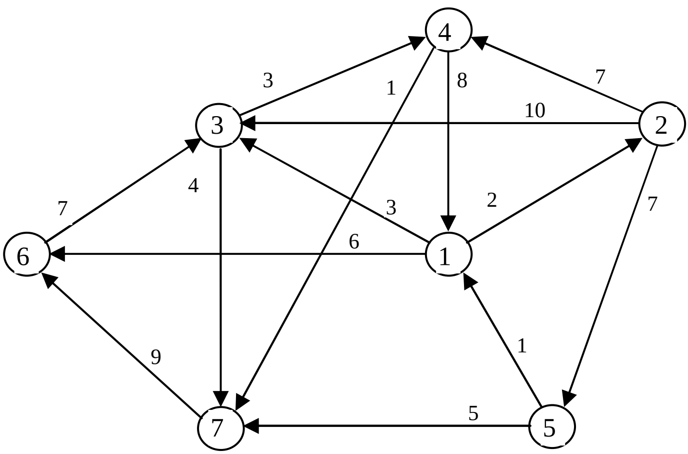


Рис. 9.9. Диаграмма сети

В качестве начального потока выберем нулевой поток (рис. 9.10).

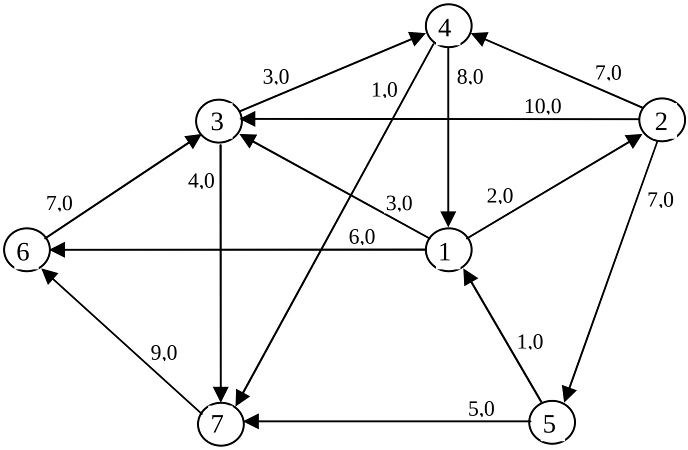


Рис. 9.10. Диаграмма сети

При поиске увеличивающей цепи окрасятся вершины и дуги, отмеченные на рис. 9.11.

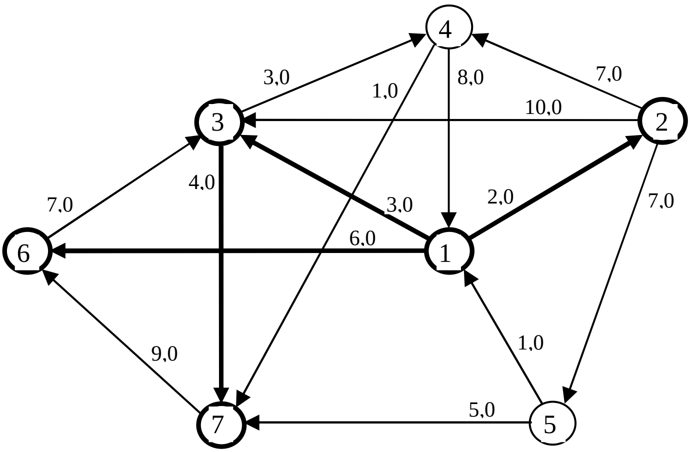


Рис. 9.11. Диаграмма сети

Увеличивающую цепь образуют дуги (1,3), (3,7). Поток по этой цепи можно увеличить на три единицы. Дуга (1,3) станет насыщенной. Результат увеличения потока представлен на рис. 9.12.

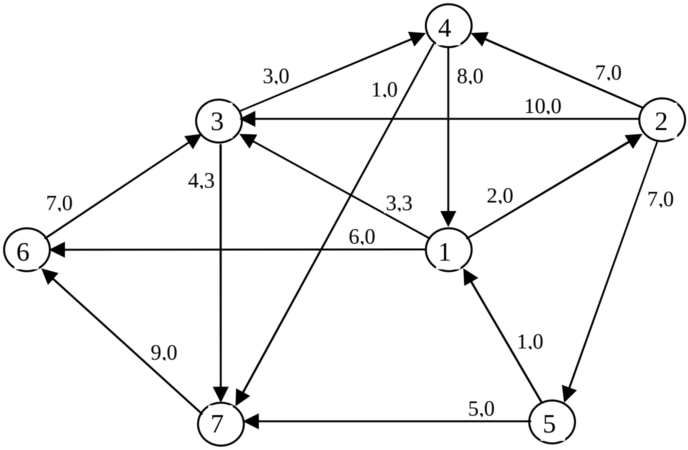


Рис. 9.12. Диаграмма сети

При поиске увеличивающей цепи в сети (рис. 9.12) окрасятся вершины и дуги, отмеченные на рис. 9.13.

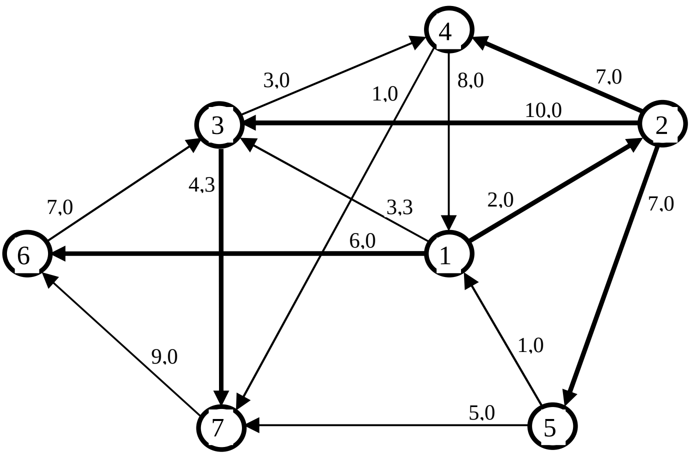


Рис. 9.13. Диаграмма сети

Увеличивающую цепь образуют дуги (1,2), (2,3), (3,7). Поток по этой цепи можно увеличить на одну единицу. Дуга (3,7) станет насыщенной. Результат увеличения потока представлен на рис. 9.14.

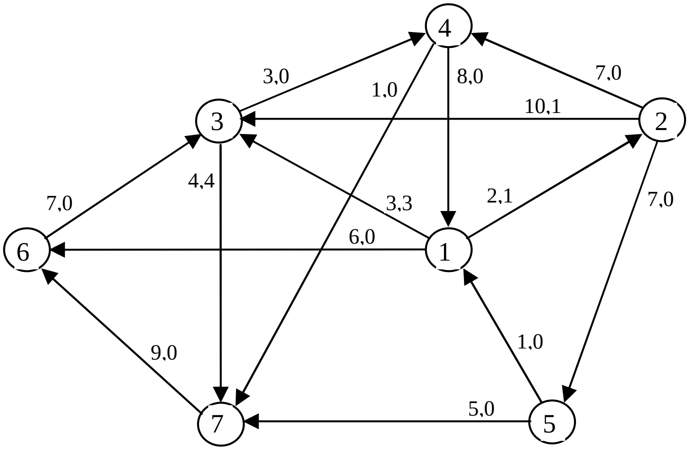


Рис. 9.14. Диаграмма сети

При поиске увеличивающей цепи в сети (рис. 9.14) окрасятся вершины и дуги, отмеченные на рис. 9.15.

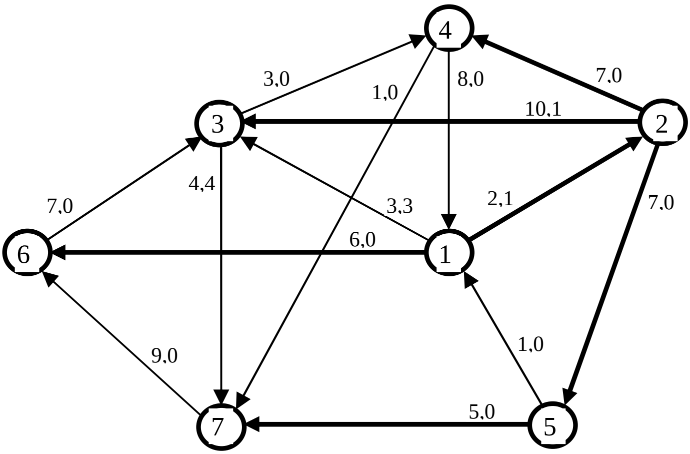


Рис. 9.15. Диаграмма сети

Увеличивающую цепь образуют дуги (1,2), (2,5), (5,7). Поток по этой цепи можно увеличить на одну единицу. Дуга (1,2) станет насыщенной. Результат увеличения потока представлен на рис. 9.16.

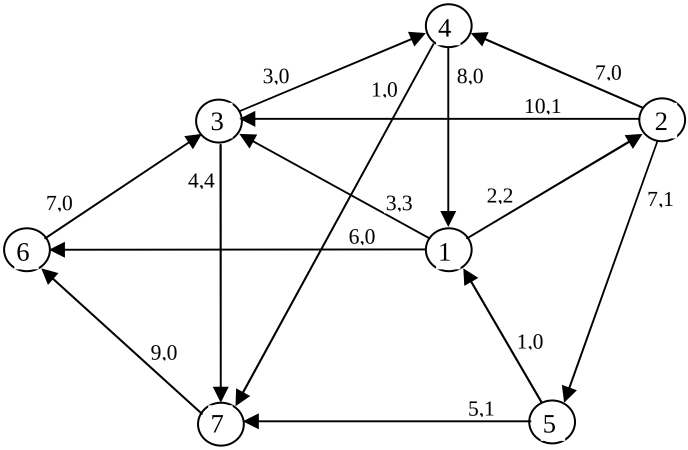


Рис. 9.16. Диаграмма сети

При поиске увеличивающей цепи в сети (рис. 9.16) окрасятся вершины и дуги, отмеченные на рис. 9.17.

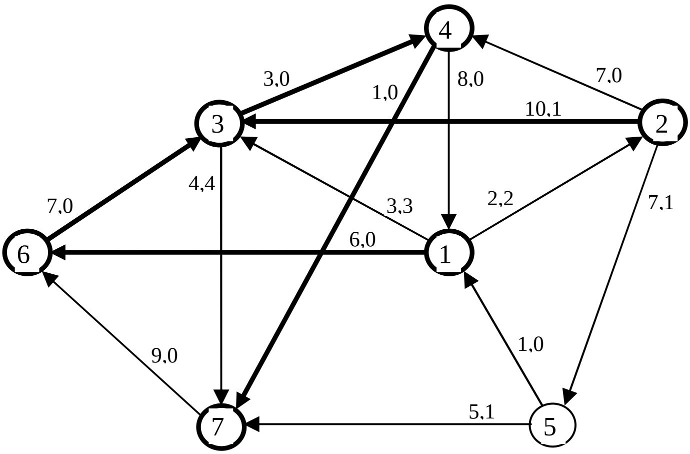


Рис. 9.17. Диаграмма сети

Увеличивающую цепь образуют дуги (1,6), (6,3), (3,4), (4,7). Поток по этой цепи можно увеличить на одну единицу. Дуга (4,7) станет насыщенной. Результат увеличения потока представлен на рис. 9.18.

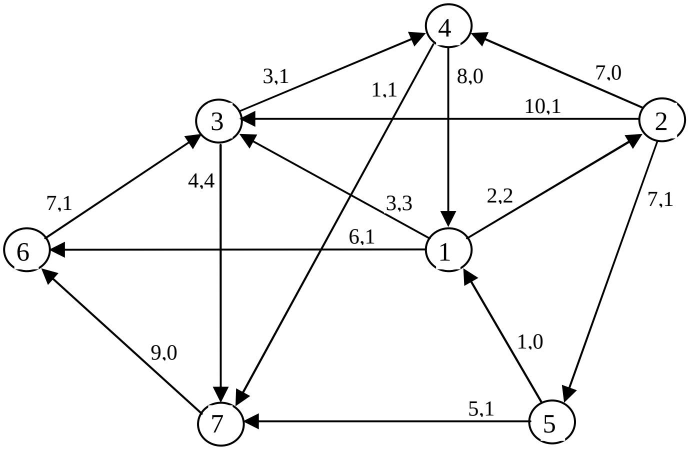


Рис. 9.18. Диаграмма сети

При поиске увеличивающей цепи в сети (рис. 9.18) окрасятся вершины и дуги, отмеченные на рис. 9.19.

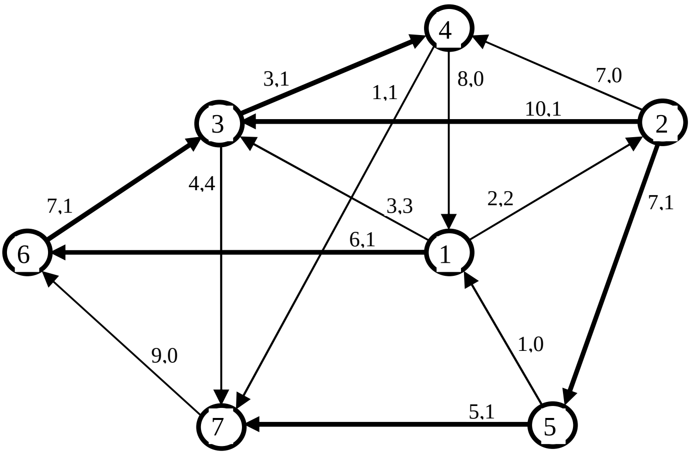


Рис. 9.19. Диаграмма сети

Увеличивающую цепи образуют дуги (1,6), (6,3), (2,3), (2,5), (5,7). Поток по этой цепи можно увеличить на одну единицу. Дуга (2,3) станет пустой. Результат увеличения потока представлен на рис. 9.20.

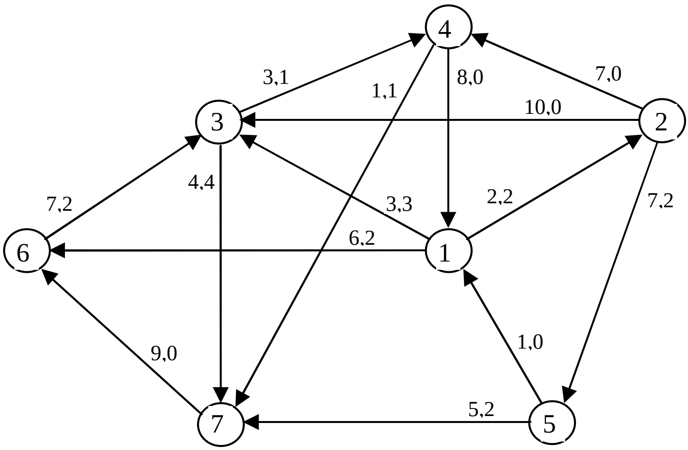


Рис. 9.20. Диаграмма сети

При поиске увеличивающей цепи в сети (рис. 9.20) окрасятся вершины и дуги, отмеченные на рис. 9.21. Вершина 7 (сток) остается неокрашенным, значит увеличивающей цепи нет и поток увеличить нельзя. Максимальный поток найден, его величина равна семи.

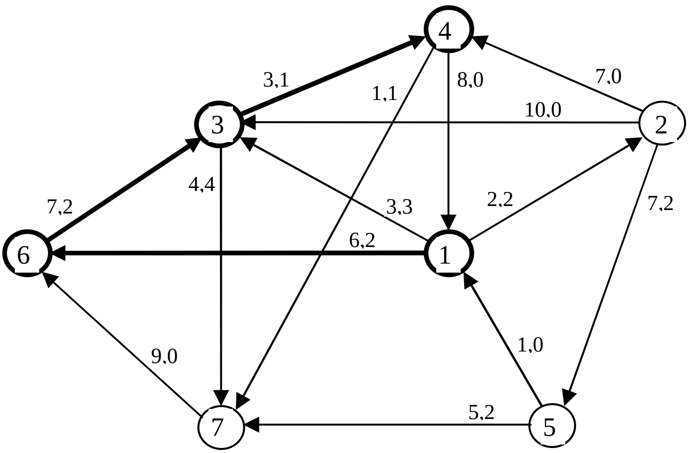


Рис. 9.21. Диаграмма сети

9.4. Поиск разреза с минимальной пропускной способностью

Орграф называется *связным* (более точно — слабо связным), если для каждой пары вершин существует соединяющая их цепь. Орграф сети, в которой существует поток, естественно является связным. *Разрезом* называется любое подмножество дуг, удаление которых делает орграф несвязным. *Простым разрезом* называется разрез, не содержащий в себе никакого другого разреза.

Найти один из разрезов можно следующим образом. Разобьем множество вершин V орграфа на два подмножества V_1 и V_2 так, что $V_1 \cap V_2 = \varnothing$ и $V_1 \cup V_2 = V$. Разрез образует множество таких дуг, начальная вершина которых принадлежит V_1 , а конечная — V_2 или начальная вершина которых принадлежит V_2 , а конечная V_1.

Пусть заданы сеть $N = (G, C)$ и поток F. Разобьем множество вершин V на два подмножества V_s и V_t так, что исток s принадлежит V_s, а сток t — множеству V_t. Обозначим через E^{st} множество дуг, начальная вершина которых принадлежит множеству V_s, а конечная — множеству V_t, а через E^{ts} — множество дуг, начальная вершина которых принадлежит множествуV_t, а конечная — множеству V_s. Объединение множеств E^{st} и E^{ts} образует разрез. Будем называть сумму пропускных способностей дуг множества E^{st} *пропускной способностью разреза*. Очевидно, что невозможно пропустить через сеть единиц потока больше, чем величина пропускной способности разреза, поэтому максимальная величина потока обязательно должна быть меньше или равна наименьшей величине пропускной способности разрезов, отделяющих сток от истока. Следовательно, разрез, пропускная способность которого равна максимальной величине потока, представляет собой разрез с минимальной пропускной способностью.

Пусть на последней итерации алгоритма поиска максимального потока не удалось найти увеличивающую цепь. Это значит, что при поиске увеличивающей цепи некоторые вершины, включая исток, окрашены (множество V_s), а остальные, включая сток — не окрашены (множество V_t). Множества V_s и V_t определяют разрез, отделяющий сток от истока. Дуги этого разреза, принадлежащие множеству E^{st}, насыщенные, иначе конечная вершина каждой из этих дуг была бы окрашена в результате выполнения алгоритма поиска увеличивающей цепи. Дуги разреза, принадлежащие множеству E^{ts}, пустые, иначе начальная вершина каждой из этих дуг была бы окрашена в результате выполнения алгоритма поиска увеличивающей цепи. Поэтому величина потока совпадает с пропускной способностью разреза, следовательно, полученный таким образом разрез имеет минимальную пропускную способность по отношению ко всем другим разрезам, отделяющих сток от истока.

Пример 9.3. Найти разрез с минимальной пропускной способностью в сети (рис. 9.9).

На последней итерации поиска максимального потока (см. пример 9.2, рис.9.21) окрашенными оказались вершины множества V_s = {1, 3, 4, 6}, а неокрашенными — вершины множества V_t = {2, 5, 7}. Множества V_s и V_t определяют разрез, отделяющий сток от истока. Множеству E^{st} принадлежат дуги (1,2), (3,7) и (4,7). Они насыщенные. Сумма пропускных способностей этих дуг равна величине максимального потока. Множеству E^{ts} принадлежат дуги (2,3), (2,4), (5,1), (7,6). Все они пустые. На рис. 9.22 представлена сеть (рис. 9.9) без дуг разреза с минимальной пропускной способностью.

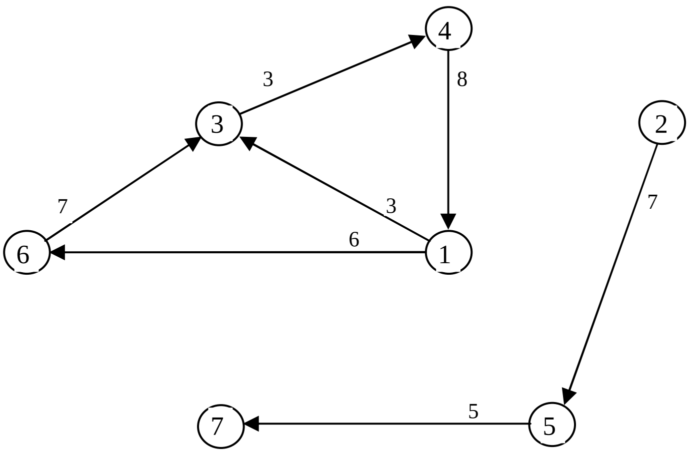


Рис. 9.22. Диаграмма сети

9.5. Поиск потока минимальной стоимости

Пусть в сети задана стоимость прохождения единицы потока по каждой дуге. Стоимость прохождения единицы потока по дуге (x,y) обозначим $a(x,y)$. Будем предполагать, что каждая из величин $a(x,y)$ целочисленная. Пусть задан поток f величины v, определяющий способ пересылки v единиц потока из источника в сток. Тогда $\sum_{(x,y)} a(x,y)\cdot f(x,y)$ определяет *стоимость потока*. Рассмотрим задачу, состоящую в организации пересылки заданного количества v единиц потока в сети из источника в сток с заданными на дугах стоимостями прохождения одной единицы потока. Другими словами, нам нужно найти поток заданной величины минимальной стоимости.

Эту задачу можно сформулировать так:

найти min $\left\{ \sum\limits_{(x,y)} a(x,y)\cdot f(x,y) \right\}$ при ограничениях 9.1 — 9.4,

т. е. она является задачей линейного программирования и может быть решена с помощью симплекс-метода. Здесь же мы рассмотрим алгоритм решения задачи о потоке минимальной стоимости, разработанный Фордом и Фалкерсоном.

Идея алгоритма Форда и Фалкерсона состоит в следующем. В качестве исходного потока принимается нулевой поток. Вначале из истока в сток пересылается как можно больше единиц потока, полная стоимость прохождения по сети каждой из них равна нулю. Для этого каждой вершине x графа приписывается метка $p(x) = 0$ и выполняется поиск увеличивающих цепей. При поиске увеличивающей цепи дуга (x,y) может быть окрашена при всех необходимых условиях (см. алгоритм поиска увеличивающей цепи), только если $p(y) - p(x) = a(x,y)$. Если увеличивающая цепь найдена, то увеличивается поток по этой цепи и выполняется поиск новой увеличивающей цепи.

Если же увеличивающая цепь не найдена (остались неокрашенными сток и может быть еще некоторые вершины), то пересылается как можно больше единиц потока, полная стоимость прохождения по сети каждой из них равна единице. Для этого метка каждой неокрашенной вершины увеличивается на единицу и выполняется поиск увеличивающей цепи так, как это описано выше.

Процесс заканчивается, когда будет получен поток заданной или максимальной величины.

Пример 9.4. Найти максимальный поток минимальной стоимости в сети (рис. 9.23). Вершину 1 считать истоком, вершину 7 — стоком. На дугах сети первое число обозначает пропускную способность, а второе — стоимость прохождения единицы потока. Начальный поток нулевой. Количество единиц потока на дуге будем указывать третьим числом.

Сначала каждой вершине приписываем метку 0. Разность меток концевой и начальной вершин каждой дуги равна нулю, но в нашей сети нет нулевых стоимостей на дугах, поэтому ни одна дуга не может быть окрашена. Окрашенным остается только исток. Поэтому метки всех вершин, за исключением истока, нужно увеличить на единицу (рис. 9.24).

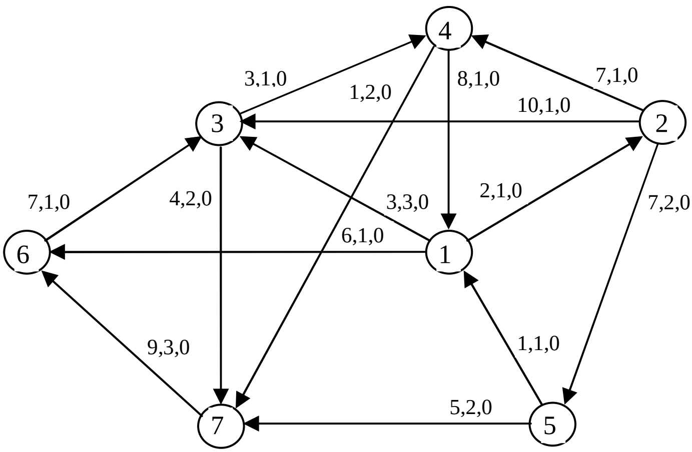


Рис. 9.23. Диаграмма сети

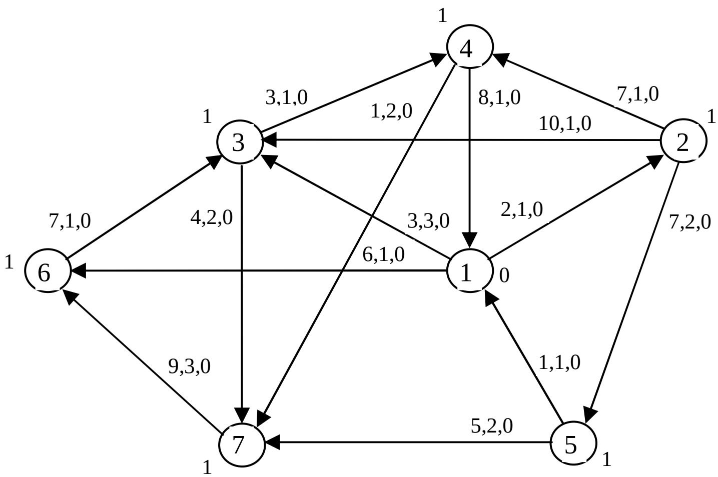


Рис. 9.24. Диаграмма сети

В этой ситуации окрашивается вершина 1(исток), дуги (1,6) и (1,2), т. к. они ненасыщенны и стоимость прохождения единицы потока по ним равна разности меток концевой и начальной вершин. Больше ничего окрасить нельзя и метки неокрашенных вершин увеличиваем на единицу (рис. 9.25). После этого вершины и дуги окрашиваются так, как показано на рис. 9.26.

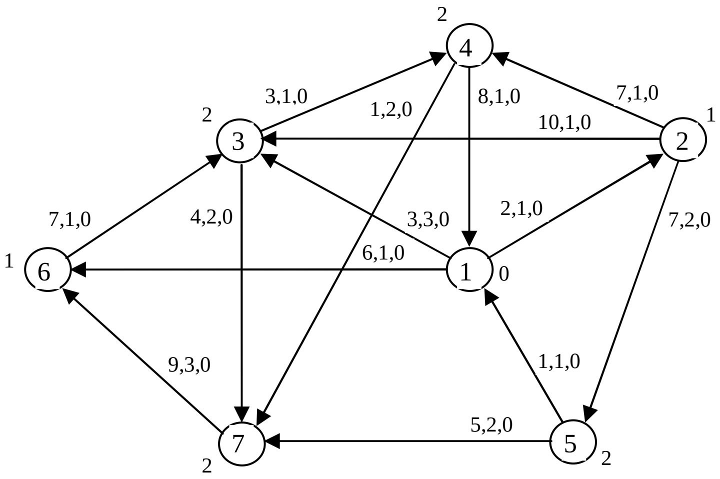


Рис. 9.25. Диаграмма сети

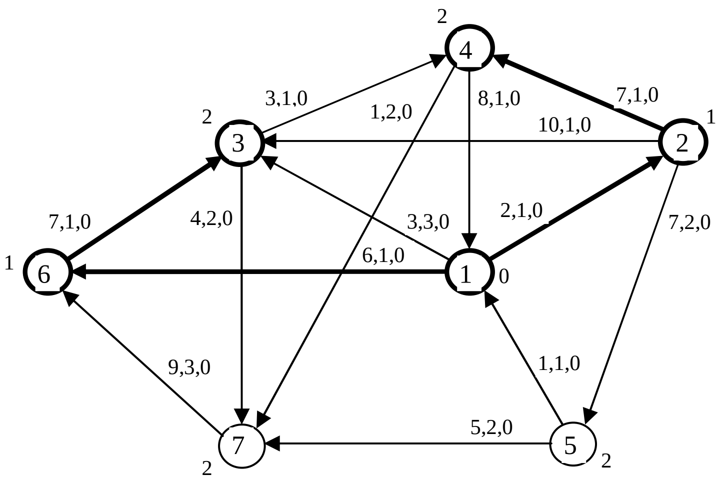


Рис. 9.26. Диаграмма сети

Теперь изменение меток вершин и окрашивание дуг и вершин представлено на рис. 9.27.

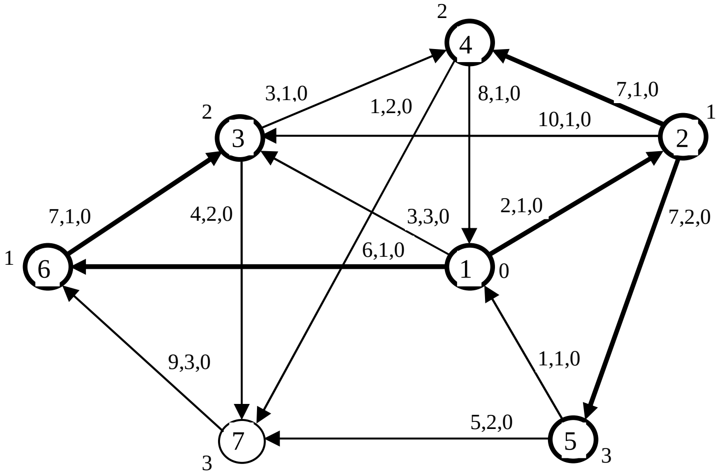


Рис. 9.27. Диаграмма сети

Увеличиваем метку неотмеченной вершины 7 и выполняем поиск увеличивающей цепи (рис. 9.28).

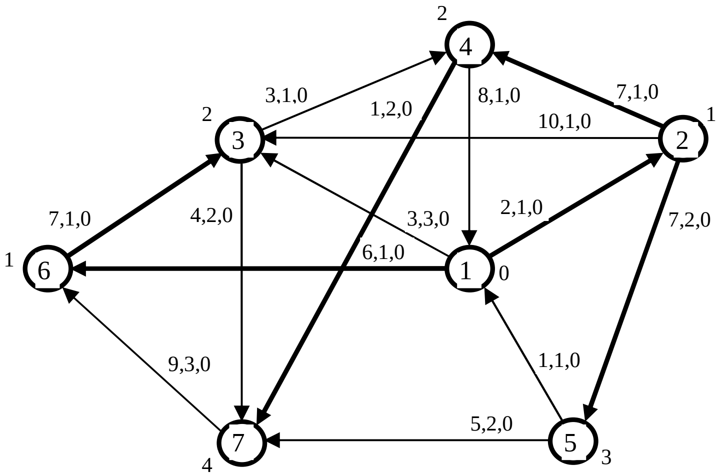


Рис. 9.28. Диаграмма сети

Увеличивающая цепь найдена: (1,2), (2,4), (4,7). По этой цепи можем увеличить поток на единицу. Окраски вершин и дуг при поиске новой увеличивающей цепи показаны на рис. 2.29.

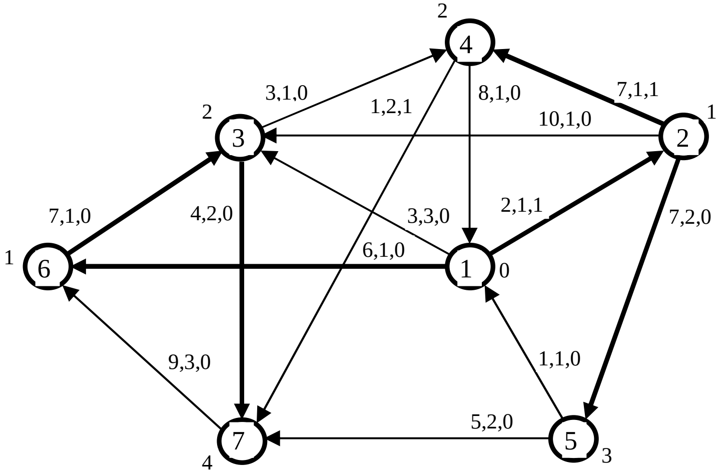


Рис. 9.29. Диаграмма сети

Увеличивающая цепь найдена: (1,6), (6,3), (3,7). Увеличиваем поток по этой цепи на четыре единицы и ищем новую увеличивающую цепь (рис. 9.30).

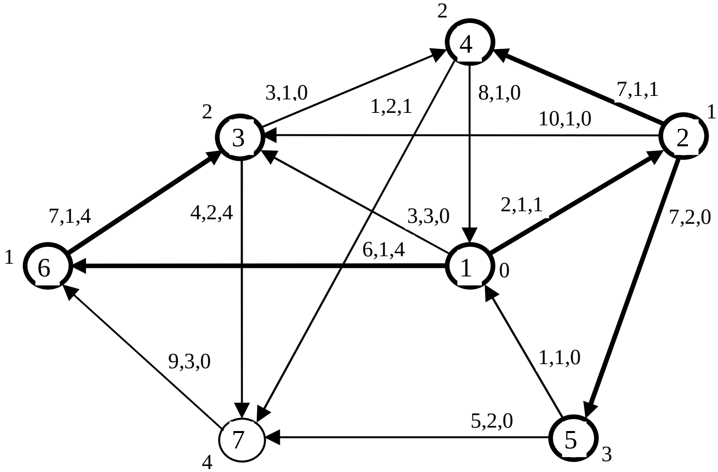


Рис. 9.30. Диаграмма сети

Вершина 7 осталась неокрашенной. Увеличиваем метку этой вершины и продолжаем поиск увеличивающей цепи (рис. 9.31).

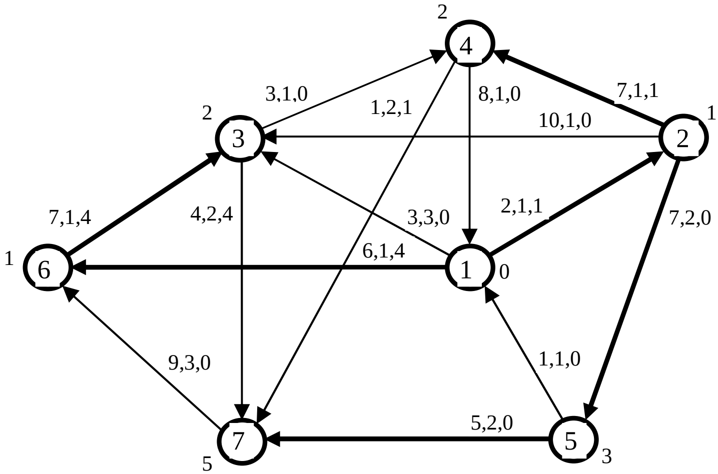


Рис. 9.31. Диаграмма сети

Увеличивающая цепь найдена: (1,2), (2,5), (5,7). Увеличиваем поток по этой цепи на единицу и ищем новую увеличивающую цепь (рис. 9.32).

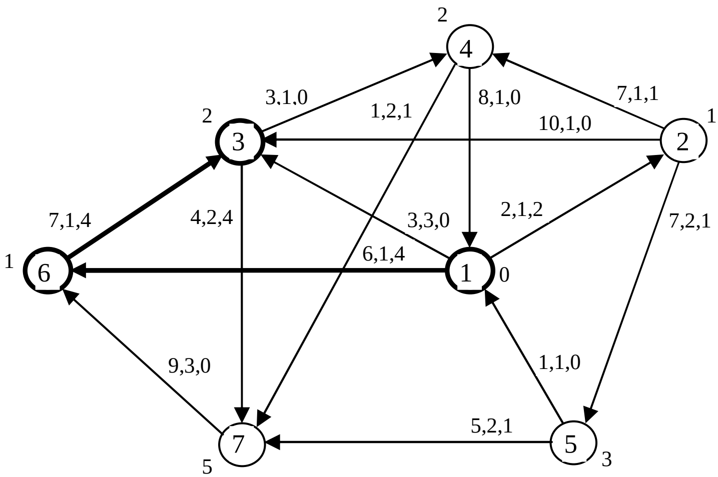


Рис. 9.32. Диаграмма сети

Увеличиваем метки неокрашенных вершин и продолжаем поиск увеличивающей цепи (рис. 9.33).

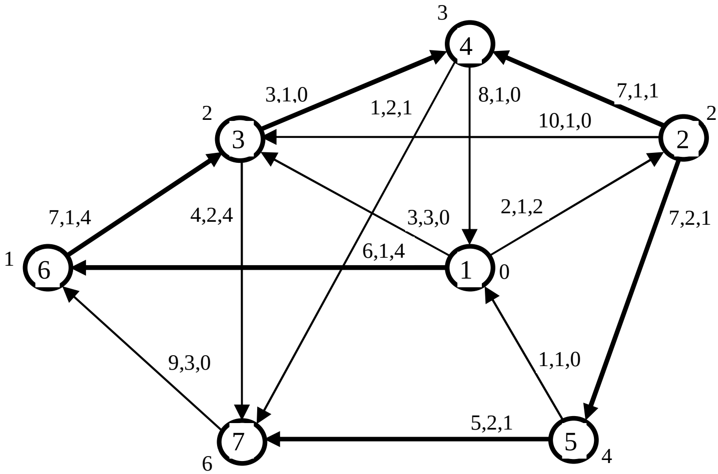


Рис. 9.33. Диаграмма сети

Увеличивающая цепь найдена: (1,6), (6,3), (3,4), (4,2), (2,5), (5,7). Увеличиваем поток по этой цепи на единицу и получаем поток максимальной величины (рис. 9.34).

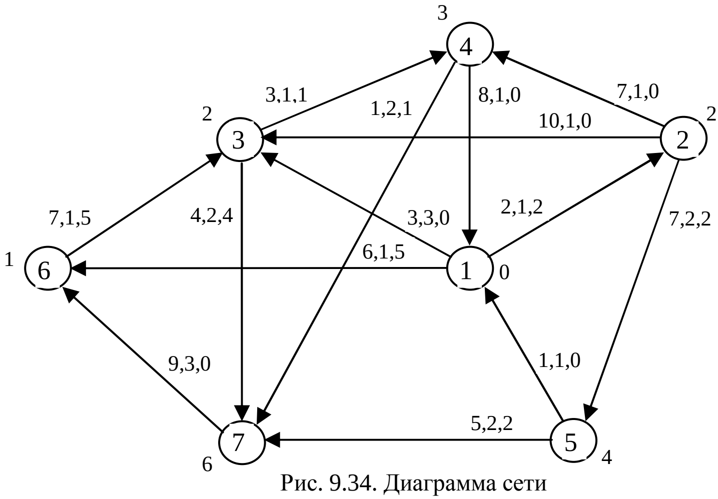


Рис. 9.34. Диаграмма сети

Стоимость полученного потока величины 7 равна 31, а стоимость потока такой же величины на рис. 9.19 равна 35, т. е. мы получили новый максимальный поток меньшей стоимости.

9.6. Контрольные вопросы и задачи для самостоятельного решения

1. Что такое ориентированный граф без петель?
2. Что такое сеть?
3. Приведите примеры систем, в качестве математической модели которых можно использовать сеть.
4. Дайте определение цепи и пути в орграфе.
5. Дайте определение потока.
6. Запишите и объясните условия потока.
7. Задан поток в сети. Как определить величину потока? Дайте два варианта ответа.
8. Дайте определение увеличивающим и уменьшающим дугам сети.
9. На какую величину можно и изменить поток в дуге?
10. Какая дуга называется насыщенной, а какая — пустой?
11. Что такое увеличивающая цепь?
12. Как определить максимальное увеличение потока по заданной цепи?
13. Как увеличить поток по увеличивающей цепи? Докажите, что при таком увеличении условия потока не нарушаются.
14. Опишите алгоритм поиска увеличивающей цепи.
15. Можно ли найти различные увеличивающие цепи в сети с заданным потоком? Приведите примеры.
16. Как определить, существует ли в сети поток заданной величины?
17. При каком условии, при выполнении алгоритма поиска увеличивающей цепи, дуга (x, y) не окрашивается при окрашенной вершине x и неокрашенной вершине y?
18. При каком условии, при выполнении алгоритма поиска увеличивающей цепи, дуга (y, x) не окрашивается при окрашенной вершине x и неокрашенной вершине y?
19. Опишите алгоритм поиска максимального потока.
20. Что называется разрезом? Какой разрез называется простым?
21. Как определить пропускную способность разреза?
22. Как соотносятся величина потока и пропускная способность какого либо разреза?
23. При каких значениях потока на дугах разреза с минимальной пропускной способностью величина потока в сети максимальна?
24. Может ли быть в сети несколько разрезов с минимальной пропускной способностью?

25. Как определяется стоимость потока?

26. В чем суть алгоритма поиска потока минимальной стоимости?

27. Как найти поток минимальной стоимости, величина которого меньше величины максимального потока?

В сетях 9.1 — 9.15, где на дугах первое число обозначает пропускную способность, а второе — стоимость прохождения единицы потока, найти:

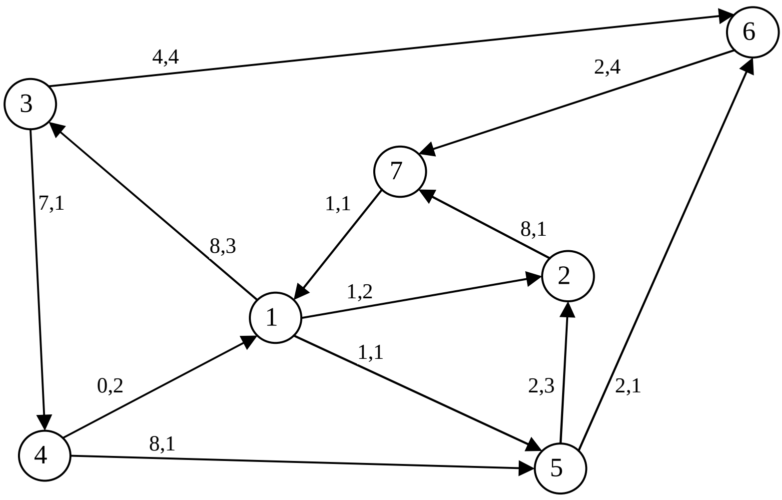
а) максимальный поток;

б) разрез с минимальной пропускной способностью;

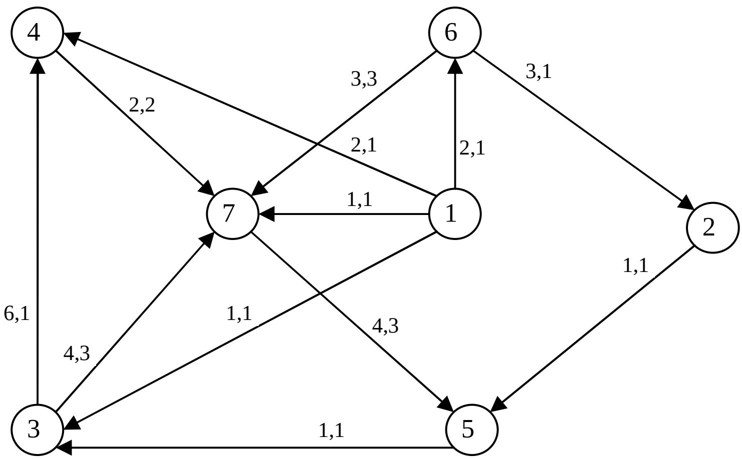
в) поток минимальной стоимости, величина которого в два раза (с округлением в большую сторону) меньше величины максимального потока;

г) максимальный поток минимальной стоимости.

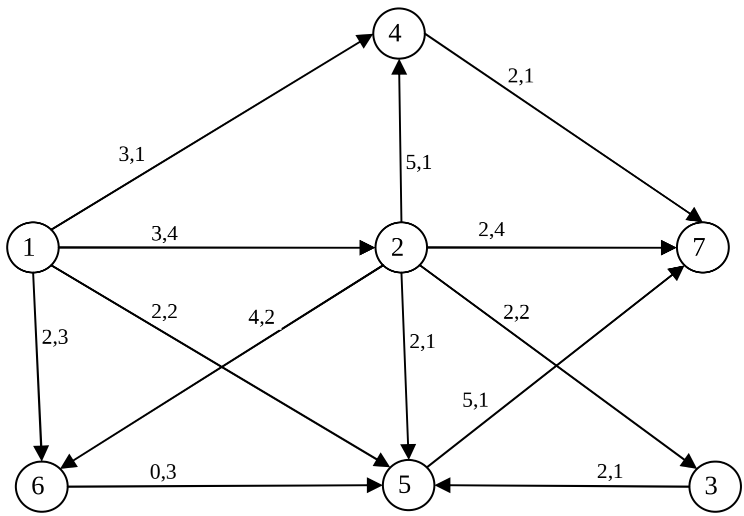
9.1.



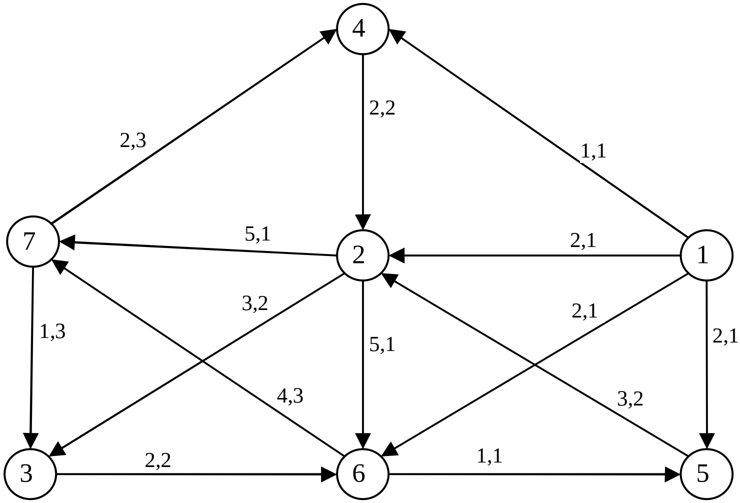
9.2.


4
6
3,3
3,1
2,2
2,1
2,1
7
1,1
1
2
1,1
6,1
1,1
4,3
4,3
3
1,1
5

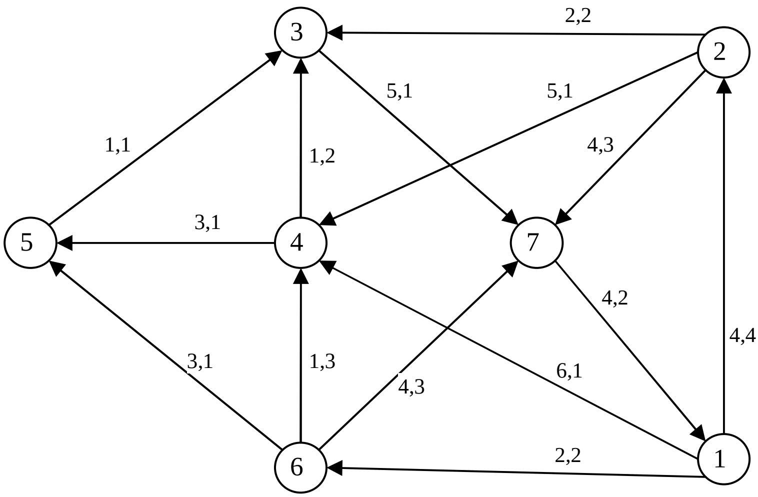
9.3


4
2,1
3,1
5,1
1
3,4
2
2,4
7
2,2
4,2
2,2
2,3
2,1
5,1
6
0,3
5
2,1
3

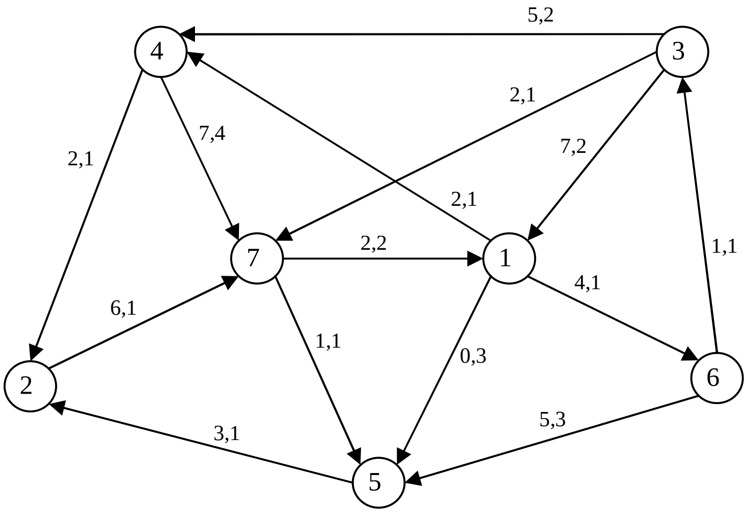
9.4


4
2,2
2,3
1,1
7
5,1
2
2,1
1
3,2
2,1
1,3
5,1
2,1
4,3
3,2
3
2,2
6
1,1
5

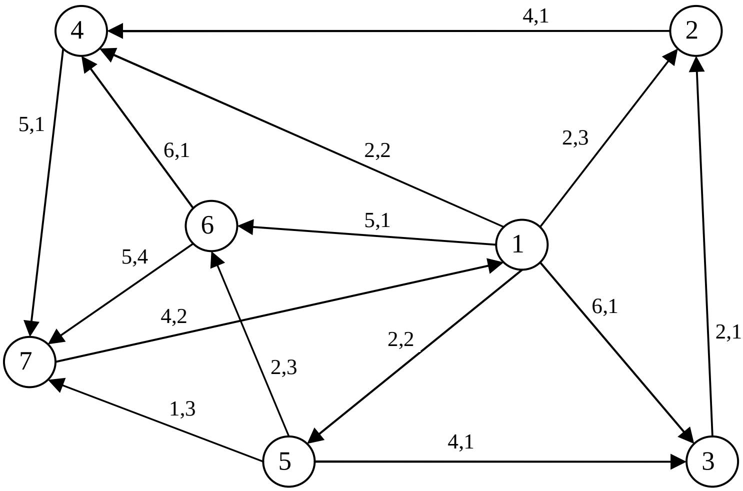
9.5


3
2,2
2
5,1
5,1
1,1
4,3
1,2
3,1
5
4
7
4,2
4,4
3,1
1,3
6,1
4,3
2,2
6
1

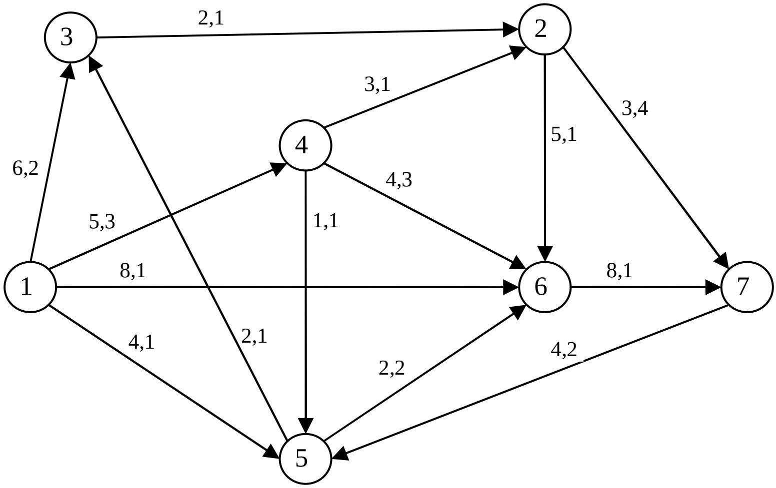
9.6


5,2
4
3
2,1
7,2
7,4
2,1
2,1
7
2,2
1
1,1
4,1
6,1
1,1
0,3
2
6
5,3
3,1
5

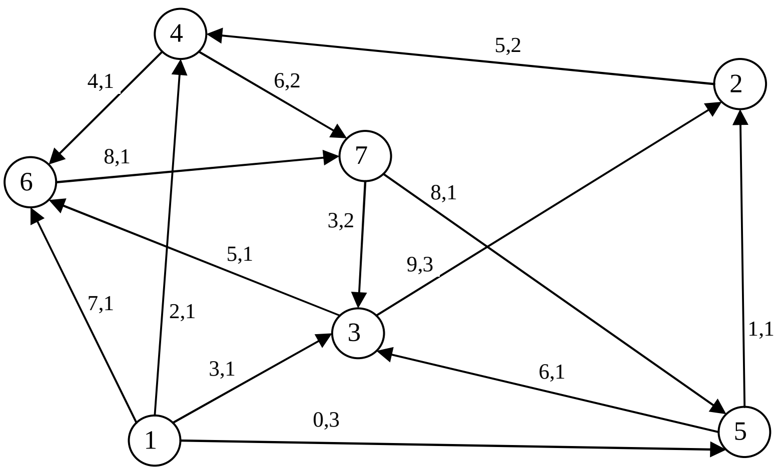
9.7


4
4,1
2
5,1
6,1
2,2
2,3
5,1
6
1
5,4
6,1
4,2
2,2
2,1
7
2,3
1,3
4,1
5
3

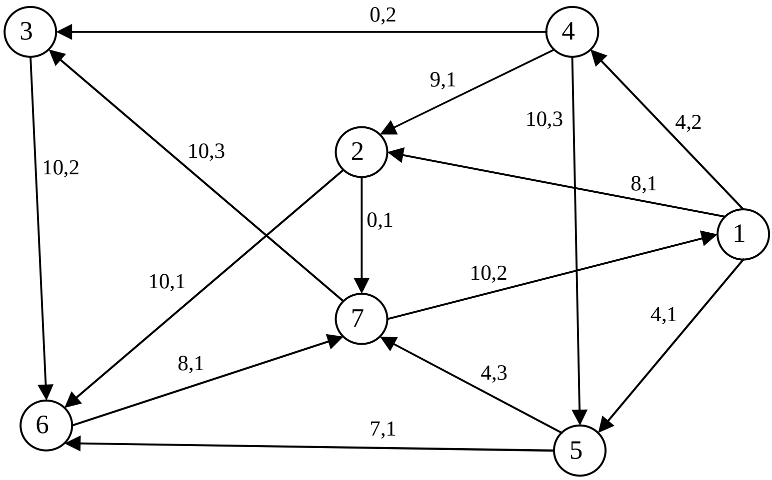
9.8


3
2,1
2
3,1
3,4
4
5,1
6,2
4,3
5,3
1,1
1
8,1
6
8,1
7
4,1
2,1
4,2
2,2
5

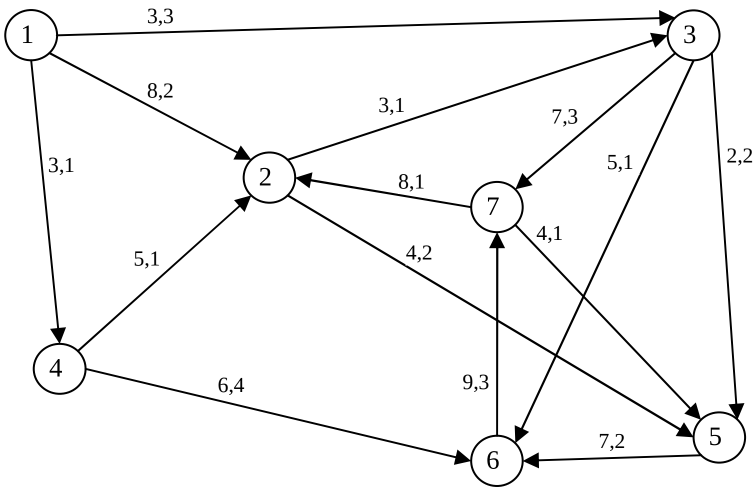
9.9


4
5,2
4,1
6,2
2
6
8,1
7
8,1
3,2
5,1
9,3
7,1
2,1
1,1
3
3,1
6,1
0,3
1
5

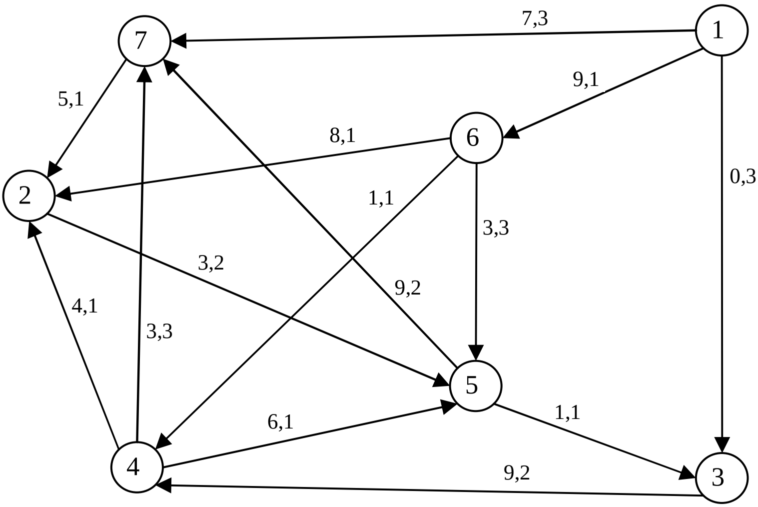
9.10


0,2
3
4
9,1
10,3
4,2
10,3
2
10,2
8,1
0,1
1
10,2
10,1
7
4,1
8,1
4,3
6
7,1
5

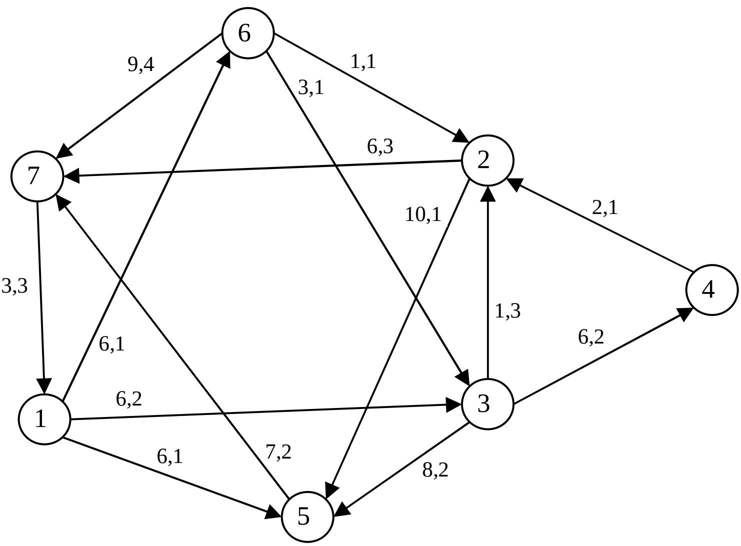
9.11


1
3,3
3
8,2
3,1
7,3
3,1
5,1
2,2
2
8,1
7
4,1
5,1
4,2
4
6,4
9,3
7,2
5
6

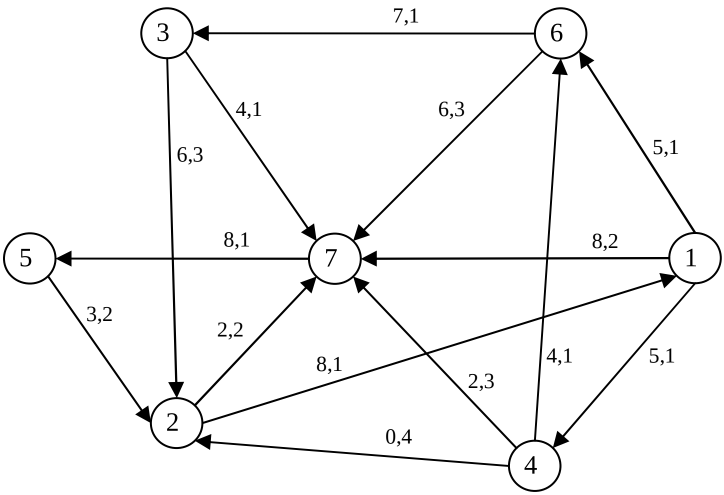
9.12


1
2
3
4
5
6
7
7,3
9,1
0,3
5,1
8,1
1,1
3,3
3,2
9,2
4,1
3,3
6,1
1,1
9,2

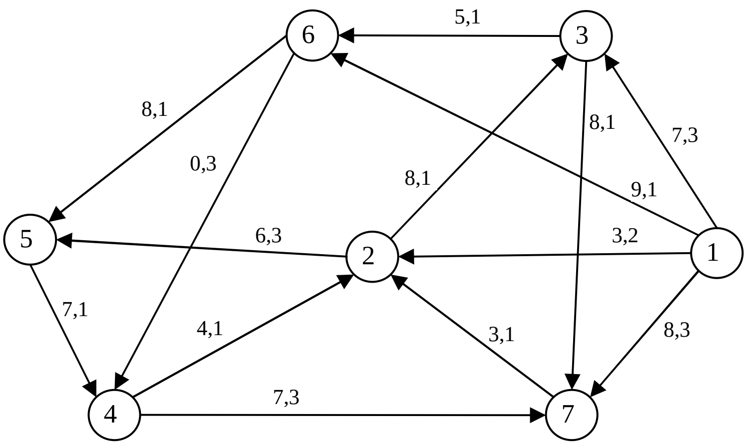
9.13


1
2
3
4
5
6
7
9,4
1,1
3,1
6,3
10,1
2,1
3,3
1,3
6,1
6,2
6,2
6,1
7,2
8,2

9.14


7,1
3
6
4,1
6,3
6,3
5,1
5
8,1
7
8,2
1
3,2
2,2
8,1
2,3
4,1
5,1
2
0,4
4

9.15


5,1
6
3
8,1
8,1
7,3
0,3
8,1
9,1
5
6,3
3,2
2
1
7,1
4,1
3,1
8,3
7,3
4
7

Д о б а в л е н и е. Понятие об устойчивости задач линейного программирования. Регуляризация неустойчивых задач

При численном решении задачи линейного программирования вида

$$\begin{aligned} & z = (\vec{c}, \vec{x}) \to \max \\ & A\vec{x} \le \vec{b}, \vec{x} \ge 0 \end{aligned} \tag{*}$$

ЭВМ оперирует лишь приближенными значениями параметров задачи, округляемыми в процессе счета. По сути дела происходит замена задачи (*) некоторой задачей

$$\begin{aligned} & z = (\vec{c}(\delta), \vec{x}) \to \max \\ & A(\delta)\vec{x} \le \vec{b}(\delta), \vec{x} \ge 0, \end{aligned} \tag{**}$$

где относительно матрицы $A(\delta)$ и векторов $\vec{b}(\delta)$, $\vec{c}(\delta)$ известно только, что они в определенной степени близки к истинным значениям матрицы A и векторов $\vec{b}$, $\vec{c}$. Кроме того, при исследовании математических моделей реальных явлений параметры модели получаются на основании экспериментальных данных и, чаще всего, известны приближенно с определенной степенью точности. Поэтому, если исследуемая модель имеет вид (*), то можно сказать лишь, что истинная модель описывается одной из задач вида (**). В связи со сказанным вопросы о взаимосвязи этих задач представляют большой практический интерес.

Близость матриц и векторов будем выражать в обычной евклидовой метрике, т. е. расстоянием между двумя матрицами $A=(a_{ij})$ и $A^{(1)}=(a_{ij}^{(1)})$ размера $m \times n$ назовем число

$$\left\| A - A^{(1)} \right\| = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij} - a_{ij}^{(1)})^2} .$$

Аналогично, если $\vec{b} = \{b_1, b_2, \dots, b_m\}, \vec{b}^{(1)} = \{b_1^{(1)}, b_2^{(1)}, \dots, b_m^{(1)}\}$, то

$$\left\| \vec{b} - \vec{b}^{(1)} \right\| = \sqrt{\sum_{i=1}^{m} (b_i - b_i^{(1)})^2} .$$

При фиксированных матрице A и векторах $\vec{b}$, $\vec{c}$ для любого $\delta>0$ символами $A(\delta)$, $\vec{b}(\delta)$, $\vec{c}(\delta)$ будем обозначать любые элементы из соответствующих δ-окрестностей:

$$\left\| A - A(\delta) \right\| < \delta, \quad \left\| \vec{b} - \vec{b}(\delta) \right\| < \delta, \quad \left\| \vec{c} - \vec{c}(\delta) \right\| < \delta . \tag{D_1}$$

Задачу (**) будем называть *возмущенной задачей*, принадлежащей δ-окрестности задачи (*), если выполнены условия (D_1).

Обозначим через $X^{(0)}$ множество решений задачи (*), а через d — оптимальное значение ее целевой функции. Аналогично через $X^{(0)}(\delta)$, $d(\delta)$ обозначим множество решений и оптимальное значение возмущенной задачи (**).

Определение D_1. *Задачу* (*) *назовем устойчивой, если существует такое число* $\delta_0>0$, *что для всех* δ *таких, что* $0 \le \delta \le \delta_0$, *задача* (**) *имеет решение. Другими словами, задача* (*) *устойчива, если она имеет решение, а также имеет решение любая задача, получающаяся из нее небольшими изменениями параметров.*

Определение D_2. *Задачу* (*) *назовем устойчивой по функционалу, если*

а) *она устойчива*;

б) *для любого* $\varepsilon > 0$ *существует такое* $\delta > 0$, *что как только выполнены условия* (D_1), *то выполнено неравенство* $\|d - d(\delta)\| < \varepsilon$. *Иначе можно сказать оптимальное значение целевой функции задачи* (*) *как функция* $d(A,\ \vec{b},\ \vec{c}\,)$ *ее параметров в этом случае непрерывна.*

Определение D_3. *Задачу* (*) *назовем устойчивой по решению, если*

а) *она устойчива*;

б) *для любого* $\varepsilon > 0$ *существует такое* $\delta > 0$, *что как только выполнены условия* (D_1) *для любого* $\vec{x}^*(\delta) \in X^{(0)}(\delta)$ *найдется* $\vec{x}^* \in X^{(0)}$, *удовлетворяющий неравенству* $\|\vec{x}^* - \vec{x}^*(\delta)\| < \varepsilon$.

Ясно, что если задача (*) неустойчива в смысле хотя бы одного из определений D_1 — D_3, то ее решение на ЭВМ может привести к результатам, сильно отличающимся от истинных.

С помощью средств, выходящих за рамки настоящего пособия, можно установить, что все три определения устойчивости эквивалентны. Точнее, если задача (*) устойчива в смысле определения D_1, то она устойчива и по функционалу и по решению. Приведем также без доказательства необходимые и достаточные условия устойчивости задачи (*).

Теорема D_1. *Стандартная задача линейного программирования* (*) *устойчива тогда и только тогда, когда существуют такие векторы* $\vec{x}^0 \in R^n$ *и* $\vec{p}^0 \in R^m$, *что* $\vec{x}^0 > 0, \vec{p}^0 > 0, \quad A\vec{x}^0 < \vec{b}, \quad A^T\vec{p}^0 > \vec{c}$ *где* A^T — *матрица, транспонированная по отношению к матрице* A.

Сформулированная теорема не позволяет устанавливать устойчивость или неустойчивость задачи (*) по ее внешнему виду, однако является весьма полезной при численном решении задач, которые имеют решение, но могут оказаться неустойчивыми.

В рамках общей методологии регуляризации некорректных задач, принадлежащей А.Н. Тихонову (см., например, [28]), построены методы, позволяющие решать произвольную задачу линейного программирования с любой степенью точности безотносительно к тому, устойчива она или нет. Однако при этом регуляризованная задача уже не является задачей линейного программирования. Здесь мы изложим метод регуляризации, не выводящий за рамки линейного программирования.

Предположим, что данная задача вида (*) не является устойчивой. Как следует из теоремы D_1, это может происходить по двум причинам: а) не существует положительного вектора $\vec{x}^0 > 0$, для которого $A\vec{x}^0 < \vec{b}$; б) не существует положительного вектора $\vec{p}^0 > 0$, для которого $A^T\vec{p}^0 > \vec{c}$. Будем предполагать, что задача (*) имеет решение. Это означает, в частности, что допустима как задача (*), так и двойственная к ней, т.е. существуют векторы $\vec{x}_1 \geq 0 : A\vec{x}_1 \leq \vec{b}$ и $\vec{p}_1 \geq 0 : A^T\vec{p}_1 \geq \vec{c}$. Сопоставим задаче (*) возмущенную задачу вида

$$\begin{aligned} &z = (\vec{c} - \delta\vec{e}, \vec{x}) \to \max \\ &A\vec{x} \leq \vec{b} + \delta\vec{v},\ \vec{x} \geq 0 \end{aligned}, \qquad (***)$$

где число $\delta > 0$, $\vec{e} = \{1,1,\dots,1\} \in R^n$, $\vec{v} = \{1,1,\dots,1\} \in R^m$.

Теорема D_2. *Если задача* (*) *имеет решение, то задача* (***) *устойчива при любом* $\delta > 0$.

Доказательство. Покажем, что для задачи (***) выполняются условия теоремы D_1. Положим $\vec{x}^0 = \vec{x}_1 + \alpha\vec{e},\ \vec{p}^0 = \vec{p}_1 + \alpha\vec{v}$. Поскольку $A\vec{x}_1 < \vec{b} + \delta\vec{v}$, $A^T\vec{p}_1 > \vec{c} - \delta\vec{e}$, то найдется такое число $\alpha > 0$, что

$$\begin{aligned} A\vec{x}^0 &= A\vec{x}_1 + \alpha A\vec{e} < \vec{b} + \delta\vec{v}, \\ A^T\vec{p}^0 &= A^T\vec{p}_1 + \alpha A^T\vec{v} > \vec{c} - \delta\vec{e}. \end{aligned}$$

При этом очевидно, что $\vec{x}^0 > 0,\ \vec{p}^0 > 0$. Теорема доказана.

Задачу (***) назовем регуляризованной по отношению к задаче (*). Следующие две теоремы показывают, что с помощью решения регуляризованной задачи в принципе можно найти с любой степенью точности решение (если оно существует) задачи вида (*) безотносительно к ее устойчивости или неустойчивости.

Теорема D_3. *Пусть* d, $d(\delta)$ — *оптимальные значения соответственно задачи вида* (*) *и регуляризованной по отношению к ней задачи* (***), *тогда справедливо равенство* $\lim\limits_{\delta\to 0} d(\delta) = d$.

Доказательство. Пусть $\vec{x}^*$ и $\vec{p}^*$ решения соответственно задачи (*) и двойственной к ней. Вектор $\vec{x}^*$ является допустимым для задачи (***) при любом $\delta > 0$. Следовательно, $(\vec{c} - \delta\vec{e}, \vec{x}^*) \le d(\delta)$ или $(\vec{c}, \vec{x}^*) - \delta(\vec{e}, \vec{x}^*) \le d(\delta)$. Учитывая, что $(\vec{c}, \vec{x}^*) = d$, получим $d - \delta(\vec{e}, \vec{x}^*) \le d(\delta)$. Проводя аналогичные рассуждения для задач двойственных к задачам (*), (***), получим $d(\delta) \le d + \delta(\vec{v}, \vec{p}^*)$. Объединяя эти неравенства, имеем

$$d - \delta(\vec{e}, \vec{x}^*) \le d(\delta) \le d + \delta(\vec{v}, \vec{p}^*) .$$

Отсюда вытекает утверждение теоремы.

Приведем без доказательства теорему, показывающую, что решение $\vec{x}(\delta)$ задачи (***) при малых $\delta > 0$ близко к множеству $X^{(0)}$ решений задачи (*).

Теорема D_4. *Для любого* $\varepsilon > 0$ *существует такое число* $\delta > 0$, *что для каждого решения* $\vec{x}(\delta) \in X^{(0)}(\delta)$ *задачи* (***) *найдется решение* $\vec{x}^* \in X^{(0)}$ *задачи* (*), *удовлетворяющее неравенству* $\left\|\vec{x}^* - \vec{x}(\delta)\right\| < \varepsilon$.

Таким образом, теоремы D_3 и D_4 обосновывают следующий способ приближенного решения задачи вида (*), имеющей решение, но, вообще говоря, неустойчивой. Выберем последовательность положительных чисел $\{\delta_k\}$ такую, что $\lim\limits_{k\to\infty} \delta_k = 0$. Будем поочередно решать задачи (***) с $\delta = \delta_k$ при k=1, 2, … . Решение каждой последующей задачи является более точным приближением к решению задачи (*), чем решение предыдущей.

Заключение

Завершая наше пособие, еще раз отметим, что в нем затронуты лишь основные математические методы и алгоритмы исследования операций и теории игр. Эту дисциплину можно рассматривать значительно шире, включая в нее методы оптимизации в задачах большой размерности, стохастическое программирование, методы принятия решений при нечеткой исходной информации и т. д. С этими разделами можно познакомиться по книге [13]. Мы же сочли возможным их не касаться, поскольку они относятся к другим разделам прикладной математики, таким, например, как методы оптимизации, многомерный статистический анализ и т. д. Авторы надеются, что ввиду своей компактности настоящее пособие найдет широкое применение при проведении всех видов занятий по курсу «Исследование операций и теория игр».

Библиографический список

1. *Акулич, И.Л.* Математическое программирование в примерах и задачах / И.Л. Акулич. — М.: Высш. шк. ,1986. — 318 с.
2. *Ашманов, С.А.* Линейное программирование / С.А. Ашманов. — М.: Наука, Главная редакция физико-математической литературы, 1981.— 340 с.
3. *Ашманов, С.А.* Теория оптимизации в задачах и упражнениях / С.А. Ашманов, А.В. Тихонов. — М.: Наука, 1991. — 447 с.
4. *Беллман, Р.* Динамическое программирование / Р Беллман. — М.: ИЛ, 1960. — 430 с.
5. *Болтянский, В.Г.* Оптимальное управление дискретными системами / В.Г. Болтянский. — М.: Наука, 1973. — 446 с.
6. *Данциг, Дж.* Линейное программирование, его обобщения и приложения / Дж. Данциг. — М.: Прогресс, 1966. — 600 с.
7. *Дикин, И.И.* Метод внутренних точек в линейном и нелинейном программировании / И.И. Дикин. — Изд. группа URSS, 2010. — 120 с.
8. *Вагнер, Г.* Основы исследования операций / Г Вагнер. — М.: Мир, 1972 — 1973. Т.1 — 3. — 987 с.
9. *Вентцель, Е.С.* Исследование операций (Задачи, принципы, методология) / Е.С. Вентцель. — М: Наука, 1980. — 208 с.
10. *Волков, И.К.* Исследование операций / И.К. Волков, Е.А. Загоруйко. — М.: Изд. МГТУ им. Н.Э. Баумана, 2004. — 440 с.
11. *Гольштейн, Е.Г.* Задачи линейного программирования транспортного типа / Е.Г. Гольштейн, Д.Б. Юдин. — М.: Наука, 1969. — 382 с.
12. *Гольштейн, Е.Г.*Линейное программирование / Е.Г. Гольштейн, Д.Б. Юдин. — М.: Наука, 1969. — 387 с.
13. *Зайченко, Ю.П.* Исследование операций / Ю.П. Зайченко. — Киев: Выща школа, 1988. — 550с.
14. *Заславский, Ю.Л.* Сборник задач по линейному программированию / Ю.Л. Заславский. — М.: Наука, 1969. — 256с.
15. *Ивченко, Г.И.* Теория массового обслуживания / Г.И. Ивченко, В.А. Каштанов, И.Н. Коваленко. — М.: Высш. шк., 1986. — 256 с.
16. Исследование операций в экономике / под редакцией профессора Н.Ш. Кремера. — М.: ЮНИТИ, 2003. — 407с.
17. *Калихман, И.Л.* Сборник задач по математическому программированию / И.Л. Калихман . — М.: Высш. шк., 1975. — 270 с.

18. *Карпелевич, Ф.И.* Элементы линейной алгебры и линейного программирования / Ф.И. Карпелевич, Л.Е. Садовский. — М.: Наука, 1967. — 274 с.
19. *Крушевский, А.В.* Теория игр / А.В. Крушевский. — Киев: Издательское объединение «Вища школа», 1977. — 216 с.
20. *Кузнецов, Ю.Н.* Математическое программирование / Ю.Н. Кузнецов, В.И. Кузубов, А.Б. Волощенко. — М.: Высш. шк., 1980. — 371 с.
21. Линейное и нелинейное программирование / под редакцией профессора И.Н. Ляшенко. — Киев: Издательское объединение «Вища школа», 1975. — 370с.
22. *Майника, Э.* Алгоритмы оптимизации на сетях и графах: Пер. с англ. / Э. Майника. — М.: Мир, 1981. — 323 с.
23. *Морозов, В.В.* Исследование операций в задачах и упражнениях / В.В. Морозов, А.Г. Сухарев, В.В. Федоров. — М: Высш. шк., 1986. — 314 с.
24. *Нейман, Дж.* Теория игр и экономическое поведение / Дж. Нейман, О. Моргерштерн. — М.: Наука, 1970. — 708 с.
25. *Саати, Т.Л.* Математические методы исследования операций / Т.Л. Саати. — М.: Воениздат, 1963. — 353 с.
26. *Схрейвер, А.* Теория линейного и целочисленного программирования: В 2-х т.: Пер с англ. / А. Схрейвер. — М.: Мир, 1991. — 360 с, 342 с.
27. *Таха, Х.* Введение в исследование операций / Х. Таха. — Изд. Вильямс, 2005. — 903с.
28. *Тихонов, А.Н.* Методы решения некорректных задач / А.Н. Тихонов, В.Я. Арсенин. — М.: Наука, 1974. — 222с.
29. *Форд, Л. Р.* Потоки в сетях: Пер. с англ. / Л. Р. Форд, Д. Р. Фалкерсон. — М.: Мир, 1966. — 276 с.
30. *Хачиян, Л.Г.* Полиномиальный алгоритм в линейном программировании / Л. Г. Хачиян. // ЖВМ и МФ. — 1980. — т. 20. — №1, с. 51 — 68.
31. *Хедли, Дж.* Нелинейное и динамическое программирование / Дж. Хедли. — М.: Мир, 1967. — 560с.

Учебное издание

Брусенцев Александр Григорьевич
Петрашев Владимир Иванович
Рязанов Юрий Дмитриевич

Исследование операций и теория игр

Подписано в печать Формат 60x84/16. Усл.печ.л. 15,2. Уч.-изд.л.16,2.
Тираж 500 экз. Заказ Цена
Отпечатано в Белгородском государственном технологическом университете
им. В.Г. Шухова
308012, г. Белгород, ул. Костюкова, 46