



• САНКТ-ПЕТЕРБУРГ •
• МОСКВА •
• КРАСНОДАР •
2013

Формат издания 84 x 108/32
Формат после обрезки: 128 x 200 мм.
Формат верстки: 98 x 172 мм.
(с колонтитолом и колонцифрой)

Раскладка:

Внутреннее поле **14 мм.**
Наружное поле **16 мм.**
Верхнее поле **13 мм.**
Нижнее поле **15 мм.**

Переплет 7БЦ, полноцвет. Корешок прямой.
Припрессовка матовой пленки.

Пустая страница

Б. А. ЕСИПОВ

МЕТОДЫ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

УЧЕБНОЕ ПОСОБИЕ

ДОПУЩЕНО
УМО по классическому университетскому образованию
в качестве учебного пособия для студентов
высших учебных заведений,
обучающихся по направлению ВПО 010300
«Фундаментальная информатика
и информационные технологии»



САНКТ-ПЕТЕРБУРГ • МОСКВА • КРАСНОДАР
2013

ББК 22.18
Е 83

Есипов Б. А.

Е 83 Методы исследования операций: Учебное пособие. — 2-е изд., испр. и доп. — СПб.: Издательство «Лань», 2013. — 304 с.: ил. — (Учебники для вузов. Специальная литература).

ISBN 978-5-8114-0917-4

Исследование операций формирует определенную методологию совершенствования существующих и создания новых систем на основе математического моделирования и применения математических методов и современных компьютерных средств.

Учебное пособие содержит все основные разделы методов исследования операций и может быть непосредственно использовано при изучении курсов «Методы оптимизации и исследование операций», «Теория игр и исследование операций», «Методы оптимизации», «Методы оптимальных решений», которые читаются при подготовке бакалавров и магистров целого ряда направлений. Элементы теории изложены компактно. При этом материал конструктивного характера: идеи методов, алгоритмы, наглядное их объяснение, числовые примеры, — составляет основное содержание пособия. Во втором издании книги существенно расширен материал по теории игр.

Книга предназначена для студентов вузов, изучающих методы оптимизации и исследования операций. Она может быть полезна научным работникам и специалистам промышленности.

ББК 22.18

Рецензенты:

А. Н. ТЕРЕХОВ — доктор физико-математических наук, профессор, зав. кафедрой системного программирования Санкт-Петербургского государственного университета;

В. П. ГЕРГЕЛЬ — доктор технических наук, профессор, декан факультета вычислительной математики и кибернетики Нижегородского государственного университета.

Обложка

А. Ю. ЛАПШИН

*Охраняется законом РФ об авторском праве.
Воспроизведение всей книги или любой ее части
запрещается без письменного разрешения издателя.
Любые попытки нарушения закона
будут преследоваться в судебном порядке.*

© Издательство «Лань», 2013
© Б. А. Есипов, 2013
© Издательство «Лань»,
художественное оформление, 2013

ВВЕДЕНИЕ

Одним из главных направлений деятельности специалиста в любой сфере является совершенствование существующих и разработка новых изделий, систем или технологий: создание лучших машин, экономное расходование ресурсов, сокращение сроков строительства и т. д.

Обычно та или иная цель может быть достигнута разными путями, но всегда важно знать наилучший из них, так как в реальных условиях приходится считаться с ограниченностью материальных ресурсов и времени, расходуемых на достижение цели. Понятие «лучший» начинает что-либо означать, когда назван количественный показатель или критерий качества принимаемого решения.

Например, изделие *A* лучше изделия *B* с точки зрения затрат материала; система *C* лучше системы *D* по показателю надежности и т. п. Вот почему получение наилучших вариантов возможно только при количественном описании предметной области, т. е. на основе математической модели.

Методология анализа сложных систем, их математическое моделирование и нахождение на этой основе наилучших (оптимальных) решений в общем виде изучается в направлении «Исследование операций». В его рамках изучаются и математические «Методы оптимизации». Последнее название применяют для методов нахождения экстремумов функций и функционалов, когда математическая модель задачи уже сформулирована.

Большой вклад в развитие методологии исследования операций и методов оптимизации внесли российские и зарубежные ученые: Л. В. Канторович, Л. С. Понтрягин, Н. Н. Моисеев, Н. Н. Воробьев, Ю. Б. Гермейер, Н. С. Краснощеков, Б. В. Гнедко, Дж. Данциг, Г. Кун, А. Таккер, Р. Беллман, Р. Гомори, Т. Саати и многие другие. Основопологающими для развития и популяризации идей исследования операций в нашей стране явились книги Е. С. Вентцель, К. Черчмена, Ф. Морза и Д. Кимбелла, Р. Акофа, Г. Вагнера, Г. Оуэна, Х. Таха.

При подготовке учебного пособия использован опыт преподавания на факультете информатики Самарского государственного аэрокосмического университета. Это предопределило перечень вопросов, которые освещены в первую очередь. При изложении автор старался предельно кратко излагать сугубо теоретические вопросы, оставляя место основным идеям и методам, имеющим универсальное значение. В виду краткости изложения в книге приведены лишь самые простые примеры и решения. Особый акцент сделан на практические алгоритмы решения разнообразных задач, которые можно положить в основу разрабатываемых компьютерных программ. На некоторые теоретические вопросы, требующие более детального рассмотрения, а также сложные примеры сделаны конкретные ссылки на литературные источники. Автор выражает благодарность к. ф.-м. н. А. О. Корепанову и д. ф.-м. н. А. А. Ковалеву за помощь в подготовке материалов учебного пособия, а также рецензентам за ценные замечания и рекомендации.

ГЛАВА 1

МЕТОДОЛОГИЯ СИСТЕМНОГО АНАЛИЗА И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

1.1. СИСТЕМНЫЙ АНАЛИЗ, СИСТЕМА, ОПТИМИЗАЦИЯ

В середине XX века во многих областях деятельности человека сформировалась необходимость изучения и совершенствования сложных систем, в том числе систем организационного типа: обороны, производственных систем, отраслей, предприятий и др.

Оказалось:

- 1) в сложных системах взаимосвязь между элементами играет большую роль, чем свойства самих элементов;
- 2) в таких системах элементы могут быть разнородны (оборудование, персонал, материалы, транспорт, условия поставок и сбыта и т. д.).

Термины *системный анализ* и *исследование операций* возникли, когда были образованы группы по исследованию военных операций в армии США и Англии во время Второй мировой войны.

К этому времени был накоплен опыт применения математических методов для моделирования и решения некоторых задач экономики (В. Леонтьев, Л. В. Канторович), теоретически обоснована возможность решения задач большой размерности на ЭВМ. Позже были созданы первые образцы таких машин. Совпали необходимость и возможность. Возникли новые идеи совершенствования организационных систем на основе математической теории игр (Нейман–Моргенштерн). Методология, сформулированная тогда и вобравшая в себя все научные достижения в области изучения сложных систем, нашла применение

не только в боевых операциях, но и в других сферах¹. В настоящее время системный (операционный) подход — основа для изучения и совершенствования сложных систем. Применительно к специалистам в области применения математики и информационных технологий в промышленности — исследование операций формирует определенную технологию совершенствования существующих и создания новых систем как технических, так и организационных.

Система — множество элементов с определенными способами взаимодействия между ними, которые все вместе выполняют цель системы.

Процесс — все, что происходит в системе. Система работает, значит, в ней происходит процесс.

Операция — часть процесса, которая наделена свойствами всей системы. Операция — это управляемое мероприятие, выполняющее определенную цель, сопоставимую с целью всей системы. Например, операция составления расписания учебных занятий для *учебного процесса в системе* «университет».

При исследовании сложных организационных систем:

- 1) невозможен экспериментальный метод исследования;
- 2) невозможно описание поведения систем только на основе какой-либо естественнонаучной теории;
- 3) при описании таких систем количество факторов, которые необходимо учитывать, велико.

Поэтому очевидно, что такие системы невозможно моделировать, изучать и совершенствовать без использования компьютерных средств и технологий.

1.2.

СХЕМА ОПЕРАЦИОННОГО ПРОЕКТА

Весь комплекс работ по изучению и совершенствованию системы (операции) проводит операционная группа системных аналитиков. Этот проект осуществляют в интересах лица, принимающего решения (ЛПР). ЛПР может отверг-

¹ Подробный исторический экскурс становления и развития идей исследования операций можно найти в [2].

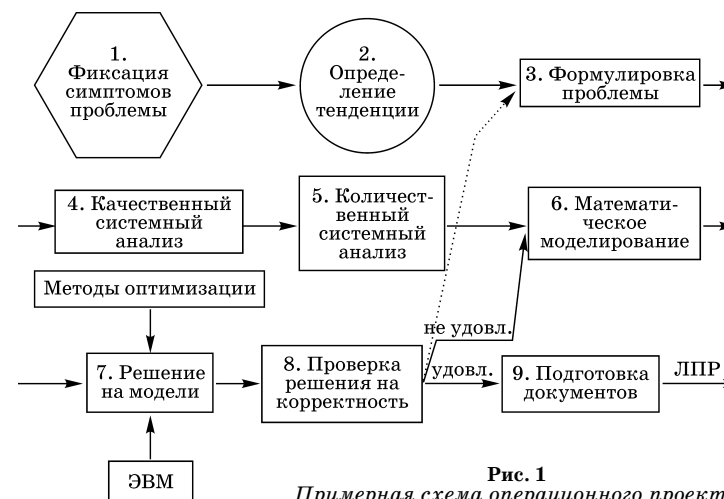


Рис. 1
Примерная схема операционного проекта

нуть проект, а может принять. На рис. 1 изображена примерная схема этапов операционного проекта.

Дадим краткую характеристику этапов операционного исследования.

1. Поводом для изучения и совершенствования системы служат зафиксированные симптомы, обнаруживающие проблемные вопросы в работе системы.

2. Установленные симптомы проблемы могут образовывать связанную цепочку фактов (тенденцию), которая помогает сформулировать проблему.

3. Важнейшим этапом исследования системы является четкая формулировка проблемы, которая присутствует на данном уровне жизнедеятельности системы.

4. Качественный системный анализ — это расщепление целостной системы (операции) на отдельные элементы (сущности). Для этого нужно:

- выделить изучаемую систему (операцию) из вышестоящей системы (операции);
- сформулировать цель, выполняемую системой (операцией);
- перечислить факторы, которые влияют на достижение цели;

— определить возможные ограничения, в рамках которых можно совершенствовать систему (операцию).

5. Количественный системный анализ предполагает описание всех перечисленных факторов, которые участвуют в операции на количественном уровне, т. е. на основе измеримых параметров. Для этого:

— устанавливается критерий K — величина, количественно измеряющая степень достижения цели системы (операции);

— вводятся количественные внутренние параметры системы, которые измеряют факторы, участвующие в описании системы (операции);

— все множество этих параметров необходимо разбить на две части:

а) неуправляемые параметры (константы), которые мы в данной конкретной системе (операции) менять не можем (производительность, нормы расхода материалов и т. п.), их обозначают как коэффициенты $A = (a_1, a_2, \dots, a_k)$;

б) управляемые параметры (переменные) — величины, которые мы можем менять: $X = (x_1, x_2, \dots, x_n)$.

6. Суть математического моделирования — установление количественных связей между введенными величинами K , A и X в виде так называемой операционной модели.

Первая часть операционной модели — это модель *целевой функции*, она устанавливает функциональную зависимость критерия K от неуправляемых параметров A и управляемых величин X в виде $K = f(X, A)$; f может быть функцией, заданной аналитически, таблично или алгоритмом². Для целевой функции указывается направление улучшения критерия:

$$K = f(X, A) \rightarrow \min (\max). \quad (1.1)$$

Этим выражением и определяется смысл оптимизации системы (операции).

Вторая часть операционной модели — математическое описание *ограничений* на выбор переменных X . Все огра-

² Иногда в качестве элементов множества X выступают функции. В этом случае f — функционал. Задачи такого рода называются вариационными и в данном пособии не рассматриваются.

ничения в общем виде можно записать как неравенства (равенства):

$$\varphi_i(X, A) \leq 0, \quad i = \overline{1, m}. \quad (1.2)$$

Каждая функция $\varphi_i(X, A)$ называется *функцией ограничения*. В некоторых задачах имеются требования к виду переменных X или K .

$$\left. \begin{array}{l} X \in D; \\ K \in M. \end{array} \right\} \quad (1.3)$$

Например, часто возникает требование, чтобы X или K были целыми. В некоторых случаях они должны принадлежать некоторому стандартному множеству значений.

Модель в виде (1.1)–(1.3) — модель операционного вида, или оптимизационная модель (неоптимизационная — без целевой функции).

Модель (1.1)–(1.3) позволяет поставить задачу оптимизации системы (операции) как математическую: найти такие управляемые переменные X , которые удовлетворяли бы системе ограничений (1.2), (1.3) и обеспечивали бы наилучшее значение критерия K .

7. Решение поставленной математической задачи требует привлечения методов оптимизации, включающих кроме классических математических методов также и специальные методы исследования операций. Реальные задачи приводят к большой размерности (до десятков тысяч). Поэтому современные методы нахождения оптимальных решений ориентированы на использование компьютерных средств.

8. При сопоставлении полученного решения с содержательной постановкой задачи можно обнаружить противоречия или какие-нибудь некорректные элементы решения. Причиной некорректности могут быть ошибки в математической модели или неучет некоторых существенных ограничений. На этом этапе может участвовать лицо, принимающее решение (ЛПР). Если полученное решение приемлемо — оно принимается, если нет — необходимо вернуться на этап математического моделирования или даже на более ранние этапы исследования.

9. Найденное оптимальное решение X^* позволяет подготовить управляющее решение в форме документа для ЛПР.

Таким образом, операционное исследование — итерационный процесс, который сходится к определенному оптимальному решению. Рассмотренная схема является примерной. Она позволяет лучше понять смысл системного анализа и исследования операций как науки. **Исследование операций (ИСО) — это наука о количественном обосновании оптимальных решений на основе построения и использования математической модели.** К сожалению, процесс перевода на количественное описание сложных систем и операций не является технологией. Математическая модель может быть удачной или неудачной для получения практических решений. Вот почему знаменитый операционист Томас Саати иронично определил науку «Исследование операций» как «искусство давать плохие советы в тех практических случаях, в которых другие науки ничего не могут посоветовать».

1.3. ОСОБЕННОСТИ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ОПЕРАЦИЙ

Математическое моделирование — самый сложный этап ИСО. Математическая модель — это такое описание системы, которое позволяет специалисту выполнить цель исследования и оптимизации системы. Для одной и той же системы можно построить различные модели, чтобы выявить различные свойства, например аэродинамическую, прочностную и т. д. Модель должна быть адекватной реальной задаче, т. е. решения, полученные в ее рамках, должны обладать такими же свойствами, что и реальная система.

К сожалению, не существует какого-либо алгоритма, по которому необходимо создать модель. Можно руководствоваться лишь принципами моделирования:

1. Необходимо решить вопрос о размерности модели. Если число параметров увеличить, то мы более точно отобразим реальное событие, но в модели будет трудно вы-

явить основные свойства. Задача становится необозримой и может не иметь решения. Поэтому число переменных по возможности стараются уменьшить, оставляя главные, существенные. Но, уменьшая число переменных, мы можем опустить существенные, и модель становится неадекватной.

2. Модель зависит от точности, с которой нужно получить решение.

3. Модель зависит от того, насколько мы знаем исходные данные.

4. Подход к построению модели может быть двояким:

4.1. Создание оригинальной модели (без учета предыдущей в этой области), т. е. разработка «с чистого листа». Это может дать, а может и не дать хороший результат. Преимущество: свежий взгляд на вещи, нет чужих ошибок. Но это может быть дороже и потребует много времени.

4.2. Использование типовых моделей для моделирования конкретных операций. В настоящее время существует большое количество типовых моделей, описывающих наиболее распространенные виды операций и систем:

- модель линейного программирования (ЛП);
- модель динамического программирования;
- игровые модели;
- модель массового обслуживания;
- модель систем управления запасами

и многие другие.

Такой подход наиболее эффективен для обучения специалистов, так как дает им инструмент для математического моделирования реальных систем в их предметной области.

1.4. ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ В ДЕТЕРМИНИРОВАННОМ СЛУЧАЕ, В УСЛОВИЯХ РИСКА И НЕОПРЕДЕЛЕННОСТИ

Если в операционной модели

$$K = f(X, A) \rightarrow \min(\max); \quad (1.4)$$

$$\varphi_i(X, A) \leq 0, \quad i = \overline{1, m} \quad (1.5)$$

все неуправляемые переменные A — заранее точно известные величины (const), а f — детерминированная функция, то такая модель называется *детерминированной*.

В некоторых задачах это не выполняется. Среди неуправляемых переменных могут быть такие $Z = (z_1, z_2, \dots, z_l)$, которые нам неизвестны. В этом случае K меняется не только с изменением X , но и Z .

$$K = f(X, A, Z).$$

Задача на максимум K становится некорректной. Если известны априорные вероятностные распределения Z , то такую модель называют *стохастической* (моделью в условиях риска). Для получения оптимального решения в этом случае используют критерий, который не зависит от Z , например, среднее значение $M[K]$ и получают решение, оптимальное «в среднем». Когда функция f меняется от Z линейно или слабо, можно искусственно свести задачу к детерминированной, заменив Z на $M(Z)$, так как в этом случае $M[K] = f(X, A, M[Z])$. В общем случае

$$M[K] = \int \dots \int \theta(z_1 \dots z_l) \cdot f(X, A, Z) dz_1 \dots dz_l \rightarrow \max,$$

где функция θ — плотность распределения вероятностей Z .

Если же статистика Z принципиально неизвестна, то это задача принятия решений *в условиях неопределенности*. (В частности, Z могут определяться сознательно противодействующим противником (см. главу 8).) Для получения оптимального решения в этом случае обычно применяют *принцип гарантированного результата (максимина)*, который обеспечивает максимум K при наихудших условиях Z . То есть $K_{\text{опт}} = \max_X \min_Z f(X, A, Z)$.

1.5. ПРИМЕР МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ОПЕРАЦИИ (ЗАДАЧА О КРАСКЕ)

Для демонстрации существования операционного подхода рассмотрим простой пример.

Содержательное описание задачи. Для окраски помещения необходимо купить 15 кг краски. Ее можно купить

в банках двух типов: по 1,5 кг стоимостью 10 руб. каждая или банках весом 0,9 кг стоимостью 8,5 руб. каждая. Для перевозки используется ящик, в который может уместиться 8 банок первого типа или 25 банок второго типа.

Необходимо дать математическую формулировку задачи минимизации стоимости покупки. Найти, сколько целых банок каждого типа надо купить.

Дать графическую интерпретацию решения. Сравнить с решением, которое получится, если банка краски первого типа будет стоить 17 руб.

Решение.

Обозначим x_1 — количество банок первого типа, x_2 — количество банок второго типа. Так как количество купленной краски не должно быть меньше 15 кг, то $1,5x_1 + 0,9x_2 \geq 15$. Каждая банка первого типа занимает $1/8$ объема ящика, а второго — $1/25$, поэтому ограничение на объем ящика можно выразить следующим образом:

$$\frac{1}{8}x_1 + \frac{1}{25}x_2 \leq 1.$$

Стоимость покупки обозначим L : $L = 10x_1 + 8,5x_2 \rightarrow \min$.

Получаем задачу, принадлежащую к классу задач *линейного программирования (ЛП)*:

$$\begin{cases} L = 10x_1 + 8,5x_2 \rightarrow \min; \end{cases} \quad (1.6)$$

$$\begin{cases} 1,5x_1 + 0,9x_2 \geq 15; \end{cases} \quad (1.7)$$

$$\begin{cases} \frac{1}{8}x_1 + \frac{1}{25}x_2 \leq 1. \end{cases} \quad (1.8)$$

Если банки вскрывать нельзя, то необходимо добавить ограничения:

$$x_1 \text{ и } x_2 — \text{целые}. \quad (1.9)$$

Задача (1.6)–(1.9) — это задача *линейного целочисленного программирования (ЛЦП)*.

Если количество переменных в задачах ЛП и ЛЦП большое, то их необходимо решать с использованием специальных алгоритмов (которые мы изучим) и компьютерных программ. В нашем случае — две переменные, следовательно, можно использовать простое геометрическое решение.

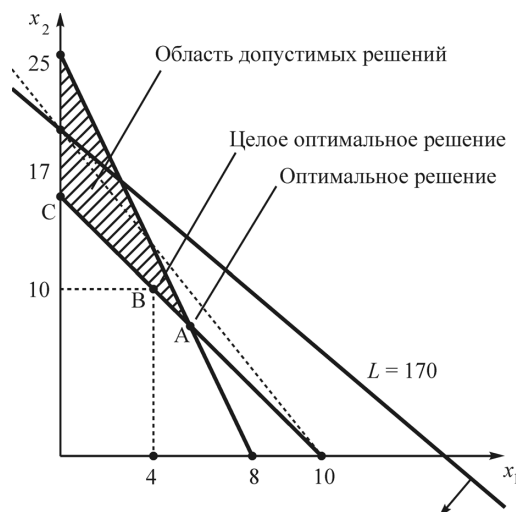


Рис. 2
Геометрическое
решение задачи
о краске

Рассмотрим неравенства (1.7) и (1.8). Сначала рассмотрим случай равенства:

$1,5x_1 + 0,9x_2 = 15$ — уравнение прямой линии.

Делим на 15 правую и левую части уравнения:

$$\frac{x_1}{10} + \frac{x_2}{16,6} = 1.$$

Это каноническое уравнение прямой. Построим его (рис. 2).

Уравнение для (1.8) имеет вид

$$\frac{1}{8}x_1 + \frac{1}{25}x_2 = 1.$$

Проверяя допустимость какой-нибудь точки для неравенств (1.7) и (1.8), например точки $(0; 0)$, получаем точки по одну сторону от прямых и определяем область допустимых решений, удовлетворяющих (1.7) и (1.8). На рисунке 2 это заштрихованная область.

Изобразим линию постоянного уровня целевой функции $L = \text{const}$. Возьмем, например, $L = 170$, получим уравнение прямой $170 = 10x_1 + 8,5x_2$.

Приведем его к канонической форме и построим

$$\frac{x_1}{17} + \frac{x_2}{20} = 1.$$

Убеждаемся, что если $L \rightarrow \min$, то эта прямая движется параллельно самой себе к началу координат. Оптимальной точкой будет точка выхода этой прямой из области допустимых решений.

Очевидно, что точка A является оптимальной для случая, когда банки можно вскрывать (x_1 и x_2 могут быть нецелыми), а точка B — оптимальная для целого решения.

Найдем координаты точки A, решив совместно уравнения (1.7) и (1.8):

$$\begin{cases} 1,5x_1 + 0,9x_2 = 15; \\ \frac{1}{8}x_1 + \frac{1}{25}x_2 = 1. \end{cases}$$

Получим $x_1^* = 5,8$; $x_2^* = 7,1$; $L^* = 118,35$.

Заметим, что «округленное» решение $x_1 = 6$; $x_2 = 7$ не допустимо.

Оптимальное решение, которое является последним при выходе прямой L, находим из целых допустимых. Это точка B с координатами $x_1 = 4$; $x_2 = 10$; $L = 125$.

Если стоимость банки краски первого типа повысится до 17 руб., то целевая функция будет $L = 17x_1 + 8,5x_2 \rightarrow \min$.

Построим прямую для $L = 170$,

$$\frac{x_1}{10} + \frac{x_2}{20} = 1.$$

Оптимальной для такой L будет точка C: $x_1 = 0$; $x_2 = 16,6$.

Целое оптимальное решение будет $x_1 = 0$; $x_2 = 17$; $L' = 144,5$.

О т в е т:

1. Если банки можно вскрывать, то необходимо взять — 5,8 банки первого типа; — 7,1 банки второго типа, стоимость $L^* = 118,35$.
2. Если банки вскрывать нельзя, то необходимо взять — 4 банки первого типа; — 10 банок второго типа, стоимость $L = 125$.
3. Если стоимость банки первого типа станет 17 руб., то оптимальное решение изменится: $x_1 = 0$; $x_2 = 17$; $L' = 144,5$.

ГЛАВА 2

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ (ЛП)

Линейное программирование — наиболее распространенный и хорошо изученный раздел исследования операций. Изучение ЛП можно проводить в двух направлениях:

1) как развитие линейной алгебры (чисто математическое);

2) изучение конкретных моделей, приводящих к ЛП.

Впервые задача ЛП была сформулирована Л. В. Канторовичем, который применил математическую модель задачи ЛП в экономике (1939). Впоследствии, в 1947 г., американец Джон Данциг разработал алгоритм решения этой задачи. С этого момента ЛП стало основным методом в системном анализе, и в первую очередь в задачах экономики. В 1975 г. Л. В. Канторович получил Нобелевскую премию как основоположник ЛП.

Задача ЛП — такая задача исследования операций, когда целевая функция и все функции ограничений линейны, а все переменные — действительные числа:

$$L = \sum_{j=1}^n c_j x_j \rightarrow \min(\max),$$

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

$$x_j \in R, \quad j = \overline{1, n}.$$

Пример.

Задача о диете. Имеются четыре вида продуктов P_1, P_2, P_3, P_4 . Известна стоимость каждого $c_1, c_2,$

c_3, c_4 . В рационе должно быть белков не менее b_1 ; жиров не менее b_2 ; углеводов не менее b_3 . Известно удельное содержание каждого вещества в единице продукта. Необходимо рассчитать рацион, содержащий необходимое количество полезных веществ, при этом стоимость всего рациона должна быть минимальной. Построим математическую модель: j — номер продукта; i — номер вещества; a_{ij} — удельное количество в j -м продукте i -го вещества (например, $a_{23} = 30$ г/кг).

Введем величину x_j — количество j -го продукта в рационе (управляемая переменная), величины c_j, a_{ij}, b_i — неуправляемые параметры — константы. Выберем критерий L — общая стоимость рациона.

Тогда целевая функция и ограничения имеют вид

$$L = c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 x_4 \rightarrow \min, \quad (2.1)$$

$$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + a_{14} x_4 \geq b_1,$$

$$a_{21} x_1 + a_{22} x_2 + a_{23} x_3 + a_{24} x_4 \geq b_2, \quad (2.2)$$

$$a_{31} x_1 + a_{32} x_2 + a_{33} x_3 + a_{34} x_4 \geq b_3.$$

Математически эту задачу можно сформулировать так: найти такие значения x_1, x_2, x_3, x_4 , которые удовлетворяли бы системе линейных неравенств и при этом линейная функция L была бы минимальной. В компактном виде это можно записать так:

$$L = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad (2.3)$$

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = \overline{1, m}, \quad (2.4)$$

$$x_j \geq 0, \quad i = \overline{1, n}; \quad n = 4; \quad m = 3. \quad (2.5)$$

Термин ЛП буквально означает линейное планирование.

2.1.

ОБЩАЯ И ОСНОВНАЯ ЗАДАЧИ ЛП

Моделируя реальную задачу, мы можем получить задачу ЛП с некоторыми особенностями:

1) в одном случае $L \rightarrow \min$, в другом $L \rightarrow \max$;

- 2) в ограничениях могут быть знаки: $\geq, \leq, =, >, <$;
 3) переменные могут быть любыми (как положительными, так и отрицательными).

Рассмотрим стандартную (основную) форму задачи ЛП, будем сокращенно называть ее ОЗЛП. В ней:

- 1) целевая функция минимизируется: $L \rightarrow \min$;
 2) ограничения только равенства;
 3) все переменные неотрицательны: $x_j \geq 0, \forall j$.

Любую задачу ЛП в общем виде можно привести к ОЗЛП, для чего необходимо выполнить следующие действия:

1. Если $L \rightarrow \max$, необходимо ввести $L' = -L$, тогда если $L \rightarrow \max$, то $L' \rightarrow \min$ и при этом $\arg \min L' = \arg \max L$.

2. Чтобы привести неравенство к равенству, необходимо:

— перенести все части неравенства в одну сторону (ту, которая ≥ 0);

— полученную функцию обозначить новой дополнительной переменной, которая ≥ 0 .

Это обозначение является равенством, эквивалентным неравенству.

Пример.

$$\begin{aligned} 2x_1 - x_2 + x_3 &\geq 4, \\ x_1 &\leq x_2 + 3x_3. \end{aligned}$$

Введем дополнительные переменные $y_1 \geq 0; y_2 \geq 0$:

$$\begin{aligned} y_1 &= 2x_1 - x_2 + x_3 - 4, \quad y_1 \geq 0, \\ y_2 &= x_2 + 3x_3 - x_1, \quad y_2 \geq 0. \end{aligned}$$

Ясно, что за переход к равенству приходится платить увеличением числа переменных.

3. Пусть x_k — переменная с любым знаком. Представим ее разностью двух положительных переменных. Тогда если вместо x_k подставить разность $x_k = x_k' - x_k''$, то остаются только неотрицательные переменные:

Пример.

$$\begin{aligned} L &= x_1 - 3x_2 + x_3 \rightarrow \max, \\ x_1 + 2x_2 - x_3 &\leq 3, \\ x_1 &\geq x_2, \\ x_1 \geq 0; \quad x_2 \geq 0; \quad x_3 &\text{ — любое.} \end{aligned}$$

1. $L' = -L = -x_1 + 3x_2 - x_3 \rightarrow \min$,
 2. $y_1 = 3 - x_1 - 2x_2 + x_3, \quad y_2 = x_1 - x_2$,
 3. $x_3 = x_4 - x_5$.

Получили ОЗЛП:

$$\begin{aligned} L' &= -x_1 + 3x_2 - x_4 + x_5 \rightarrow \min, \\ y_1 &= 3 - x_1 - 2x_2 + x_4 - x_5, \quad y_2 = x_1 - x_2, \\ (x_1, x_2, x_4, x_5, y_1, y_2) &\geq 0. \end{aligned}$$

Таким образом, ОЗЛП в общем виде можно записать так:

$$L = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad (2.6)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m}, \quad (2.6)$$

$$E_j \geq 0, \quad j = \overline{1, n}. \quad (2.8)$$

В матричном виде $CX \rightarrow \min, AX = B, X \geq 0$.

Известно, что система линейных уравнений (2.7) совместна, когда ранг матрицы A равен рангу расширенной матрицы AB ; $A = \{a_{ij}\}, B = \{b_i\}$. Но допустимыми решениями задачи ЛП являются не все, а только неотрицательные решения, так как должно выполняться условие (2.8). Если $m = n$, то может быть только одно решение; если $m < n$, то может быть бесчисленное множество решений, но таких, которые удовлетворяют системе (2.7). Эти точки образуют так называемую область допустимых решений (ОДР).

2.2.

ГЕОМЕТРИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ЗАДАЧИ ЛП

Рассмотрим частный случай $n - m = 2$. Тогда m переменных выбираются базисными, а $(n - m)$ — свободными. Рассмотрим такую систему, где

x_1, x_2 — свободные, x_3, x_4, \dots, x_n — базисные,

$$\left. \begin{aligned} x_3 &= \alpha_{31}x_1 + \alpha_{32}x_2 + \beta_3; \\ x_4 &= \alpha_{41}x_1 + \alpha_{42}x_2 + \beta_4; \\ &\dots \\ x_n &= \alpha_{n1}x_1 + \alpha_{n2}x_2 + \beta_n; \end{aligned} \right\} \quad (2.9)$$

$$L = \gamma_1 x_1 + \gamma_2 x_2 + \gamma_0 \rightarrow \min.$$

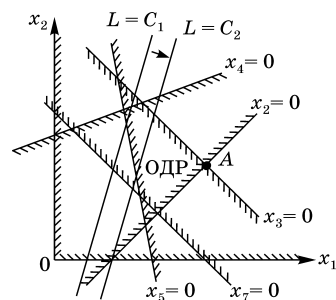


Рис. 3
Геометрическая интерпретация решения задачи ЛП

пустимых решений (ОДР, см. рис. 3).

ОДР содержит все точки, удовлетворяющие (2.9).

Рассмотрим теперь целевую функцию L .

Пусть $L = C_1: \gamma_1 x_1 + \gamma_2 x_2 = C_1 - \gamma_0$ — уравнение прямой.

Точки на прямой $L = C_2$ лучше, чем $L = C_1$ ($C_2 < C_1$), так как они дают меньшее значение целевой функции.

Оптимальное решение получается в точке выхода линии уровня целевой функции L (точка A) из ОДР. Отсюда ясно, что в задаче ЛП внутри ОДР не может быть оптимальной точки.

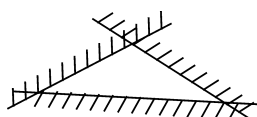


Рис. 4
Случай отсутствия допустимых решений

Выводы из геометрической интерпретации ОЗЛП:

1. ОЗЛП не обязательно имеет решение (рис. 4).

2. ОДР всегда выпуклый многоугольник (рис. 5).

3. Если есть допустимое решение, то обязательно, что есть оптимальное решение. ОДР может быть открытой, в этом случае оптимальной точки может не быть, если целевая функция изменяется, как показано в I (двигаемся в открытую область). Но если область открыта, то это не значит, что нет оптимального решения. Для целевой функции II оптимальное решение находится в точке A (рис. 6).

4. В задаче может быть бесчисленное множество оптимальных решений. Оптимальное решение будет в любой точке AB (рис. 7).

Рассмотрим систему (2.9).

Пусть $x_3 = 0$,

$$\alpha_{31}x_1 + \alpha_{32}x_2 + \beta_3 = 0$$

— уравнение прямой.

Значит, для $x_3 > 0$ точки лежат по одну сторону от этой прямой. Покажем это штриховкой. Аналогично для других равенств (2.9). Очевидно, точки, где все штриховки пересекаются, и есть область допустимых решений (ОДР, см. рис. 3).

5. Если оптимальное решение есть, то его необходимо искать в вершине многоугольника. Признаком вершины является пересечение двух прямых, т. е. равенство нулю двух переменных.

6. Если $n - m = 2$, т. е. в ОЗЛП две свободные переменные, тогда, чтобы попасть в вершину, можно приравнять к нулю две свободные переменные.

7. Если взять две переменные равными нулю (рис. 8), то мы можем попасть в недопустимую вершину (точка 1).

Точка 1: ($x_3 = 0$) и ($x_5 = 0$). Чтобы перейти в соседнюю вершину (точка 2), нужно вместо $x_5 = 0$ взять $x_4 = 0$, а $x_5 \neq 0$. Получаем точку 2 ($(x_3 = 0)$ и ($x_4 = 0$)). Допустимую вершину назовем опорной.

Рассмотрим случай, когда $n - m = 3$.

$$x_4 = \alpha_{41}x_1 + \alpha_{42}x_2 + \alpha_{43}x_3 + \beta_4,$$

...

$$x_4 = 0 \text{ — плоскость,}$$

$$x_4 \geq 0 \text{ — полупространство.}$$

Тогда ОДР — выпуклый многогранник.

Целевая функция $L = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3 \rightarrow \min$ (при $L = \text{const}$ — плоскость).

Допустимые решения находятся во внутренних и граничных точках многогранника, а оптимальное решение, очевидно,

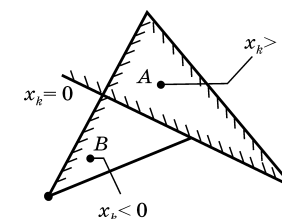


Рис. 5
Доказательство выпуклости ОДР

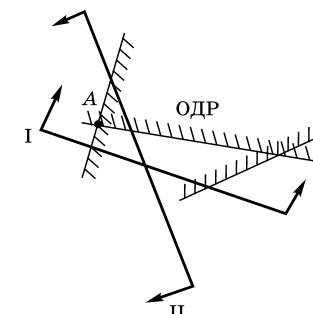


Рис. 6
Оптимальное решение задачи ЛП

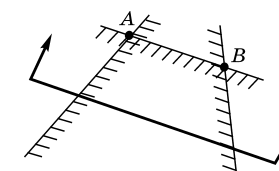


Рис. 7
Частный случай бесчисленного множества оптимальных решений

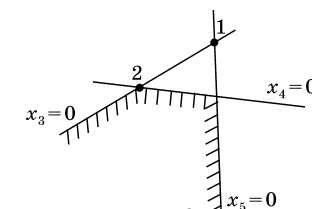


Рис. 8
Геометрический образ необходимого признака оптимального решения

может находиться только в вершине многогранника в точке «последнего» касания плоскости — целевой функции и многогранника, где пересекаются три плоскости, т. е. три переменных равны нулю.

Если в общем случае $n - m = k$, то у задачи k свободных переменных. Следовательно, нужно искать оптимальное решение там, где k переменных равны нулю. Так как в вершине в особом случае может пересекаться и большее число плоскостей, то в оптимальной точке не меньше k переменных должны быть равны нулю. Заметим, что это является необходимым условием оптимального решения для задачи ЛП.

2.3. ИДЕЯ СИМПЛЕКС-МЕТОДА РЕШЕНИЯ ЗАДАЧИ ЛП

Рассмотрим ОЗЛП, где базисные переменные $x_k + 1, \dots, x_n$ разрешены относительно свободных x_1, x_2, \dots, x_k . Например:

$$\begin{aligned} L &= 2x_1 + 3x_2 + \dots + 5x_k + 6 \rightarrow \min, \\ x_{k+1} &= 2x_1 + 3x_2 - x_3 + \dots + 6x_k + 4, \\ x_{k+2} &= x_1 - x_2 + x_3 + \dots + 7x_k - 7, \\ &\dots \\ x_n &= 2x_1 + 2x_2 - 3x_3 + \dots + x_k + 2; \\ (x_1 = x_2 = \dots x_k = 0) &\text{ — вершина,} \end{aligned}$$

при этом базисные переменные

$$x_{k+1} = 4, \quad x_{k+2} = -7, \quad x_n = 2.$$

В ОЗЛП все переменные должны быть ≥ 0 , следовательно, эта вершина недопустима. Признаком недопустимости вершины является наличие отрицательных свободных членов в ограничениях-равенствах. Если отрицательный свободный член есть, то надо перейти в другую вершину, для чего одну базисную переменную сделать свободной и приравнять к нулю, а какую-то свободную превратить в базисную. Следовательно, переход из одной вершины в другую осуществляется путем замены базисной переменной на свободную. Это называется *симплекс-преобразова-*

нием. Если мы будем переходить из вершины в вершину «в сторону ОДР», то за конечное число таких переходов попадем в допустимую вершину. Назовем ее опорной вершиной, а соответствующее решение *опорным решением*.

Пусть получено опорное решение, например, для следующей задачи. Проверим, оптимально оно или нет:

$$\begin{aligned} x_3 &= 2 + x_1 - x_2, \\ x_4 &= 4 - x_1 + x_2, \\ L &= 3 + 2x_1 - x_2 \rightarrow \min, \\ x_1 = x_2 &= 0; \quad x_3 = 2; \quad x_4 = 4; \quad L = 3. \end{aligned}$$

Видно, что если все коэффициенты при свободных переменных в целевой функции положительны, то никакие увеличения свободных переменных от 0 вверх не уменьшают L , следовательно, это L является оптимальным значением. Если же есть отрицательный коэффициент при свободной переменной (в нашем примере при x_2), то ее надо сделать базисной, так как если она будет положительной, то L будет меньше. Но x_2 может увеличиваться не бесконечно, так как в ограничениях есть отрицательный коэффициент при x_2 . Видно, что увеличивать x_2 можно только до $x_2 = 2$, так как в противном случае x_3 будет отрицательной, а это невозможно. Так мы нашли базисную переменную x_3 , которую надо поменять с свободной x_2 . Поиск этой пары для симплекс-преобразования и является основным элементом *симплекс-метода*.

Таким образом, симплекс-метод — это метод решения задач ЛП, который основывается на процедуре перехода от одной вершины к другой до тех пор, пока не придем в оптимальную вершину. Он состоит из трех алгоритмов:

1. Алгоритм перехода из одной вершины в другую, когда известна пара преобразования. Он пересчитывает коэффициенты системы уравнений для нового базиса. Назовем его *алгоритмом симплекс-преобразования*.

2. Алгоритм нахождения такой пары преобразования, чтобы при переходе в новую вершину мы приближались к ОДР. Это *алгоритм отыскания опорного решения*.

3. Алгоритм нахождения такой пары преобразования, чтобы новая опорная вершина давала лучшее значение

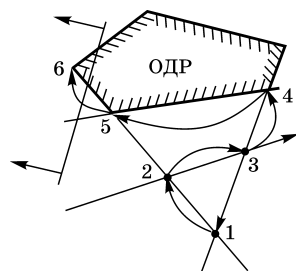


Рис. 9
Геометрическая
интерпретация идеи
симплекс-метода

целевой функции. Это алгоритм отыскания оптимального решения.

В симплекс-методе начинают со стандартной вершины (все свободные переменные равны нулю), т. е. с точки 1 (это недопустимая вершина, см. рис. 9). Далее последовательно переходим из одной вершины в другую (2, 3) и попадаем в опорную вершину 4.

Найдя опорное решение, мы движемся по вершинам ОДР так, чтобы целевая функция улучшалась и при этом эти вершины были опорными (5 и 6). Шестая вершина оптимальная.

Эти три алгоритма гарантируют либо отыскание оптимального решения, либо доказательство отсутствия допустимого решения, либо доказательство отсутствия оптимального решения.

2.4. СИМПЛЕКС-ТАБЛИЦА, СТАНДАРТНЫЙ АЛГОРИТМ СИМПЛЕКС-ПРЕОБРАЗОВАНИЯ

Существует много форм симплекс-таблиц (СТ). Мы рассмотрим наиболее компактную, в которой строки — базисные переменные, а столбцы — свободные переменные. Каждая СТ содержит коэффициенты модели задачи ЛП. Для записи СТ необходимо представить ОЗЛП в стандартной форме:

$$\begin{aligned} L &= \gamma_0 - (\gamma_1 x_1 + \gamma_2 x_2 + \dots + \gamma_k x_k), \\ y_1 &= b_1 - (\alpha_{11} x_1 + \alpha_{12} x_2 + \dots + \alpha_{1k} x_k), \\ y_2 &= b_2 - (\alpha_{21} x_1 + \alpha_{22} x_2 + \dots + \alpha_{2k} x_k), \\ &\dots \\ y_n &= b_n - (\alpha_{n1} x_1 + \alpha_{n2} x_2 + \dots + \alpha_{nk} x_k), \end{aligned}$$

т. е. на первом месте свободный член, а коэффициенты при свободных переменных меняют знак.

Пример.

$$\begin{aligned} L &= 2x_1 - x_3 + 4, \\ y_1 &= 2x_1 + x_2 + x_3, \\ y_2 &= 3x_1 + x_2 - x_3 + 1, \\ y_3 &= 2x_1 - x_2 - 2. \end{aligned}$$

Стандартная форма:

$$\begin{aligned} L &= 4 - (-2x_1 + 0x_2 + 1x_3), \\ y_1 &= 0 - (-2x_1 - 1x_2 - 1x_3), \\ y_2 &= 1 - (-3x_1 - 1x_2 + 1x_3), \\ y_3 &= -2 - (-2x_1 + 1x_2 - 0x_3). \end{aligned}$$

Записываем коэффициенты этой формы в левый верхний угол соответствующей клетки СТ. В нижней части будем писать промежуточные результаты вычислений (табл. 1).

Таблица 1

	1	x_1	x_2	x_3
L	4 0	-2 0	0 0	1 0
y_1	0 -1	-2 3	-1 -1	-1 -1
y_2	1 -1	-3 3	-1 -1	1 -1
y_3	-2 1	-2 -3	1 1	0 1

Каждое решение, соответствующее СТ, выглядит так: свободные переменные равны 0, базисные — свободным членам:

$$\begin{aligned} x_1 &= x_2 = x_3 = 0; \\ y_1 &= 0; \quad y_2 = 1; \quad y_3 = -2; \\ L &= 4. \end{aligned}$$

Рассмотрим порядок вычислений для перехода в новую вершину.

Пусть нам необходимо выполнить симплекс-преобразование $x_i \leftrightarrow x_j$. Тогда нужно получить новую симплекс-таблицу. Для этого выполним алгоритм.

Алгоритм 1.

1. Отыскиваем в СТ элемент a_{ij} и объявляем его *ведущим*. Тогда i -я строка и j -й столбец — ведущие.

2. Вычисляем величину, обратную ведущему элементу, и записываем ее в нижней части ведущей клетки:

$$\lambda = \frac{1}{a_{ij}}.$$

3. Все элементы ведущей строки умножаем на λ . Результат записываем в нижней части соответствующих клеток.

4. Все элементы ведущего столбца умножаем на $-\lambda$. Результат записываем в нижней части клеток.

5. Отмечаем верхние элементы ведущей строки и нижние элементы ведущего столбца.

6. В каждой клетке, не принадлежащей ни ведущему столбцу, ни ведущей строке, в нижней части записываем произведение отмеченных элементов, стоящих в том же столбце и той же строке, что и данная клетка.

7. Переписываем СТ, заменяя в соответствии с табл. 2:

Т а б л и ц а 2

	1	x_1	y_2	x_3
L	4	-2	0	1
y_1	-1	1	-1	-2
y_2	-1	3	-1	-1
y_3	-1	-5	1	1

- обозначение базисной переменной x_i обозначением свободной переменной x_j , и наоборот;
- элементы ведущей строки и ведущего столбца нижними элементами;
- все оставшиеся элементы суммой нижнего и верхнего числа.

В приведенных СТ показано преобразование $y_2 \leftrightarrow x_2$.

2.5. АЛГОРИТМ ОТЫСКАНИЯ ОПОРНОГО РЕШЕНИЯ ЗАДАЧИ ЛП

Этот алгоритм выполняется сразу после записи симплекс-таблицы.

Алгоритм 2.

1. Просматриваем столбец свободных членов (несмотря на свободный член строки L). Если все свободные члены ≥ 0 , то данная таблица уже соответствует опорному решению, иначе переходим к п. 2.

2. В столбце свободных членов выбираем первый по порядку отрицательный элемент и в этой строке находим еще один отрицательный элемент. Если такого элемента нет, то задача ЛП не имеет решения, если есть, то переходим к п. 3.

3. Столбец с найденным отрицательным элементом объявляем ведущим.

4. В ведущем столбце фиксируем элементы, имеющие такой же знак, что и знак соответствующих свободных членов, и среди них в качестве ведущего элемента выбираем тот, отношение к которому соответствующего свободного члена минимально:

$$\min_i \left(\frac{b_i}{a_{ij}} \right), \text{ знак } b_i = \text{знак } a_{ij}.$$

5. По выбранному ведущему элементу выполняем стандартное симплекс-преобразование (*алгоритм 1*) и переходим к п. 1.

Если при вычислениях нет ошибок, то количество отрицательных свободных членов должно уменьшаться или должна уменьшаться их абсолютная величина.

Примечание к пункту 2. Задача ЛП не имеет решения, когда соответствующая строка СТ, например, имеет вид: $y_4 = -1 - (2x_1 + x_2 + 4x_3)$. Это выражение не удовлетворяется ни при каких неотрицательных переменных.

2.6. АЛГОРИТМ ОТЫСКАНИЯ ОПТИМАЛЬНОГО РЕШЕНИЯ ЗАДАЧИ ЛП

Этот алгоритм выполняется после того, как найдено опорное решение, т. е. когда в симплекс-таблице все свободные члены ≥ 0 .

Алгоритм 3.

1. Просматриваем элементы строки L (несмотря на свободный член). Если все элементы ≤ 0 , то данная симплекс-таблица уже соответствует оптимальному решению, иначе переходим к п. 2.

2. В строке L выбираем наибольший положительный элемент и соответствующий столбец объявляем ведущим.

3. В ведущем столбце находим положительные элементы. Если таких нет, то задача не имеет оптимального решения, иначе переходим к п. 4.

4. Среди положительных элементов ведущего столбца выбирается тот, отношение к которому соответствующего свободного члена минимально:

$$\min_i \left(\frac{b_i}{a_{ij}} \right), \quad b_i \geq 0, \quad a_{ij} > 0.$$

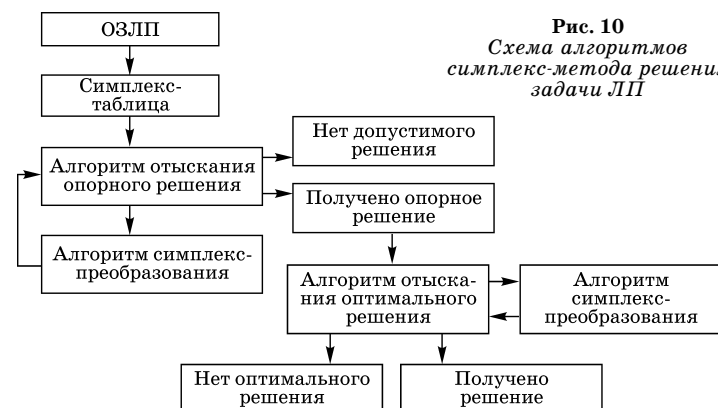
5. По выбранному ведущему элементу осуществляем симплекс-преобразование (алгоритм 1) и переходим к п. 1.

Примечание к пункту 3. Если в ведущем столбце нет положительных элементов, то задача не имеет оптимального решения — целевая функция уменьшается неограниченно (линия уровня целевой функции движется в открытую область). Это значит, что в исходной модели не учтено какое-то важное ограничение.

Схема алгоритмов решения задач ЛП показана на рис. 10:

Пример. Решить задачу ЛП:

$$\begin{aligned} L &= x_1 + x_2 \rightarrow \max; \\ x_2 &\leq x_1 + 2; \\ x_2 &\geq 1; \\ x_1 &\leq 4; \\ (x_1, x_2) &\geq 0. \end{aligned}$$



Решение.

Приведем задачу к ОЗЛП:

$$\begin{aligned} L' &= -x_1 - x_2 \rightarrow \min; \\ y_1 &= x_1 - x_2 + 2; \\ y_2 &= x_2 - 1; \\ y_3 &= 4 - x_1; \\ L' &= 0 - (x_1 + x_2); \\ y_1 &= 2 - (-x_1 + x_2); \\ y_2 &= -1 - (0x_1 - x_2); \\ y_3 &= 4 - (x_1 + 0x_2). \\ \frac{2}{1} &= 2; \quad \frac{-1}{-1} = 1 (\min). \end{aligned}$$

Таблица 3

	1	x_1	x_2
L'	0	1	1
y_1	2	-1	1
y_2	-1	0	-1
y_3	4	1	0

Таблица 4

	1	x_1	y_2
L'	-1	1	1
y_1	1	-1	1
x_2	1	0	-1
y_3	4	1	0

Построим симплекс-таблицу (см. табл. 3).

В табл. 4 получено опорное решение:

$$x_1 = y_2 = 0, \\ y_1 = 1, x_2 = 1, y_3 = 4, L' = -1.$$

В табл. 4 и 5 выполняем алгоритм отыскания оптимального решения:

Таблица 5

	1	y_3	y_2
L'	-5	-1	1
y_1	5	1	1
x_2	1	0	-1
x_1	4	1	0

Таблица 6

	1	y_3	y_1
L'	-10	-2	-1
y_2	5	1	1
x_2	6	1	1
x_1	4	1	0

В табл. 6 получено оптимальное решение:

$$y_3 = y_1 = 0, y_2 = 5, x_2 = 6, x_1 = 4, L' = -10, L_{\max} = 10.$$

Так как в данной задаче всего две свободных переменных, то можно рассмотреть ее геометрически на плоскости (рис. 11).

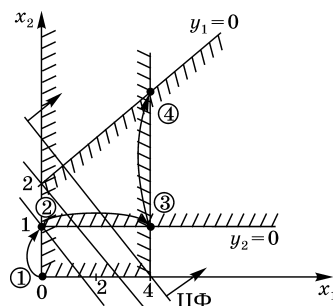


Рис. 11
Геометрическая интерпретация примера решения задачи ЛП

сти (рис. 11).

Поскольку $y_1 = 0, y_3 = 0$, то решение находится в точке, в которой выполняются равенства: $x_1 - x_2 + 2 = 0$ и $4 - x_1 = 0$.

На рис. 11 показана последовательность перехода от вершины к вершине в соответствии с работой алгоритма симплекс-метода. Цифрами обозначены соответствующие симплекс-таблицы.

2.7. АЛГОРИТМ ПОЛУЧЕНИЯ ПЕРВОГО БАЗИСНОГО РЕШЕНИЯ С ИСПОЛЬЗОВАНИЕМ СИМПЛЕКС-ПРОЦЕДУРЫ (МЕТОД ИСКУССТВЕННОГО БАЗИСА)

Было показано, что алгоритмы симплекс-метода требуют, чтобы исходная система ограничений-равенств была приведена к базису, т. е. какие-нибудь базисные переменные должны быть выражены через свободные. Оказывается, что первый базис можно также получить, используя алгоритм симплекс-метода. Для каждого ограничения-равенства введем искусственную переменную z_i ($i = 1, m$). Каждое ограничение-равенство $\beta_i - (g_{i1} \cdot x_1 + \dots + g_{in} \cdot x_n) = 0$ формально записывается так:

$$z_i = \beta_i - (g_{i1} \cdot x_1 + \dots + g_{in} \cdot x_n), \quad i = 1, m,$$

т. е. переменные z_i тождественно равны нулю. Последнее равенство определяет так называемый искусственный базис.

Вводим дополнительную целевую функцию

$$L_0 = z_1 + z_2 + \dots + z_m \rightarrow \min,$$

для чего суммируем коэффициенты при переменных x_j всех равенств z_i и получаем ОЗЛП, готовую для решения симплекс-методом.

Решая эту задачу на основе приведенных алгоритмов, последовательно выводим из базиса все искусственные переменные z_i . При этом, как только переменная z_r становится свободной, то столбец z_r исключается из симплекс-таблицы.

Пример.

Пусть исходную задачу ЛП мы привели к форме ОЗЛП и получили систему ограничений-равенств

$$x_1 + 2x_2 = 4, \\ x_1 - x_2 + 2x_3 = 2.$$

Найдем первый базис, т. е. разрешим базисные переменные (их всего будет 2) относительно свободных ($n - m = 3 - 2 = 1$). Для этого запишем

$$\begin{aligned} z_1 &= 4 - (x_1 + 2x_2), \\ z_2 &= 2 - (x_1 - x_2 + 2x_3), \\ L_0 &= z_1 + z_2 = 6 - (2x_1 + x_2 + 2x_3) \rightarrow \min. \end{aligned} \quad (2.10)$$

Запишем симплекс-таблицу. В табл. 7 находим ведущий элемент по алгоритму нахождения оптимального решения, получаем пару для симплекс-преобразования и переходим к табл. 8, в которой исключаем столбец z_2 .

Таблица 7

	1	x_1	x_2	x_3
L_0	6 -4	2 -2	1 2	2 -4
z_1	4 -2	1 -1	2 1	0 -2
z_2	2 2	1 1	-1 -1	2 2

Таблица 8

	1	x_2	x_3
L_0	2 -2	3 -1	-2 2
z_1	2 2/3	3 1/3	-2 -2/3
x_1	2 2/3	-1 1/3	2 -2/3

Таблица 9

	1	x_3
L	0	0
x_2	2/3	-2/3
x_1	8/3	4/3

В табл. 8 находим ведущий элемент и пару $z_1 \leftrightarrow x_2$, переходим к табл. 9.

Табл. 9 соответствует оптимальному решению вспомогательной задачи (2.10) и дает выражение базисных переменных x_1 , x_2 относительно свободной переменной x_3 , а именно:

$$\begin{aligned} x_2 &= \frac{2}{3} + \frac{2}{3}x_3; \\ x_1 &= \frac{8}{3} - \frac{4}{3}x_3. \end{aligned}$$

Теперь можно приступить к выполнению симплекс-алгоритма, записав вместо строки L_0 целевую функцию задачи, выраженную через свободные переменные. Чтобы такая запись получалась автоматически, можно в табл. 7 кроме строки L_0 записать строку целевой функции исходной ОЗЛП и выполнять описанный выше алгоритм, помня, что ведущий элемент не может быть ни в строке L_0 , ни в строке ЦФ.

2.8. ВЫРОЖДЕННАЯ ЗАДАЧА ЛП

При использовании симплекс-метода некоторые свободные члены могут быть равны нулю. Это значит, что в вершине, которой соответствует СТ, равны нулю не k переменных, а больше (равна нулю и базисная переменная). В этом случае при выборе ведущего элемента отношение b_i/a_{ij} будет минимальным именно для этой строки. Ясно, что соответствующую свободную переменную увеличить никак нельзя и целевая функция при переходе в новую вершину не меняется. Мы остаемся фактически в этой же вершине, хотя формально перешли в новую.

Такая задача называется *вырожденной*. Вырождение отрицательно сказывается на эффективности вычисления. Признаком вырожденности является равенство нулю некоторых свободных членов. В этом случае может произойти два отрицательных явления:

1. «Пробуксовка». Переходим в новую вершину, где равна нулю другая совокупность переменных, а на самом деле остаемся в той же точке. Значение ЦФ не меняется.

2. «Зацикливание» алгоритма. Через некоторое количество операций мы можем прийти к первоначальной таблице, следовательно, если алгоритм не менять, то зацикливаемся.

Существует несколько способов борьбы с зацикливанием:

- использование степени свободы алгоритма. Например, если в алгоритме сказано: «берем первый по порядку...», то при обнаружении заикливания можно взять «второй по порядку». Аналогично, если сказано «берем любой...»;
- «зашумление» коэффициентов задачи ЛП. Прибавляем матрицу малых случайных величин к матрице A . Получаем $A + \xi$, где A — матрица коэффициентов; ξ — матрица очень малых случайных величин. Тогда матрица коэффициентов СТ не будет содержать одинаковых чисел и после вычислений появление нулей будет маловероятным. Подробнее этот сложный вопрос освещен в специальной литературе.

2.9. ДВОЙСТВЕННАЯ ЗАДАЧА ЛП

Важным вопросом анализа полученного решения ЛП является определение чувствительности решения к параметрам модели (коэффициентам целевой функции, свободным членам ограничений, коэффициентам a_{ij}). Теория этого вопроса тесно связана с так называемой двойственной задачей ЛП. Двойственная задача ЛП получается из прямой, и она имеет физический смысл, когда прямая задача есть задача об использовании ресурсов. Имеются ресурсы m типов в количестве b_1, \dots, b_m . Для изготовления одного изделия j -го типа расходуется a_{ij} сырья i -го типа. Каждое j -е изделие продается по цене c_j . Ставится задача максимизации стоимости проданных изделий.

$$L = \sum_{j=1}^n c_j x_j \rightarrow \max, \quad (2.11)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad (2.12)$$

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

Можно рассмотреть эту проблему по-другому. Пусть некоторые величины Δ_i — стоимость единицы i -го ресурса. Нужно определить эту стоимость так, чтобы выполнялись неравенства, выражающие то, что стоимость ресур-

са, затраченного на j -е изделие, была бы не меньше стоимости изделия и при этом общие затраты на ресурсы были минимальны:

$$L^* = \sum_{i=1}^m b_i \Delta_i \rightarrow \min, \quad (2.13)$$

$$\sum_{i=1}^m a_{ij} \Delta_i \geq c_j, \quad j = \overline{1, n}. \quad (2.14)$$

Задача (2.13), (2.14) называется двойственной по отношению к задаче (2.11), (2.12). Она получается из прямой задачи (2.11), (2.12) так: коэффициенты целевой функции двойственной задачи — это правые части прямой задачи и наоборот; а коэффициенты a_{ij} суммируются по другому индексу (матрица коэффициентов двойственной задачи — это транспонированная матрица прямой задачи). Направление оптимизации целевой функции меняется на противоположное.

Основное свойство задачи ЛП: оптимальные значения целевой функции прямой и двойственной задачи ЛП совпадают:

$$L_{\text{опт}}^* = \sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i \Delta_i^*.$$

Оптимальное решение задачи ЛП находится в *седловой точке*:

$$\max L \text{ по } x_j = \min L^* \text{ по } \Delta_i.$$

Существует и двойственный симплекс-метод решения задачи ЛП, который в ряде случаев эффективнее прямого [1, 10].

Экономический смысл двойственной задачи: какова цена единицы каждого ресурса Δ_i , чтобы при заданных количествах ресурсов b_i и стоимости единицы продукции c_j обеспечить минимум общей стоимости затрат.

Δ_i называется *двойственной оценкой*, или *теневой ценой*. Если решение прямой задачи найдено, то Δ_i — есть коэффициенты, стоящие в строке L в последней оптимальной СТ.

Когда $\Delta_i > 0$, то она показывает, насколько увеличивается значение целевой функции прямой задачи, если соответствующие запасы сырья i -го типа увеличиваются на единицу.

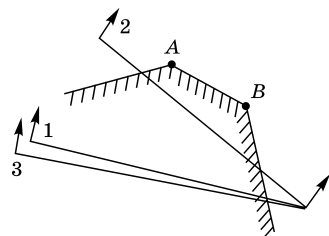


Рис. 12
Геометрическая интерпретация устойчивости оптимального решения задачи ЛП

Соответствующая дополнительная переменная y_i при этом равна нулю. Это означает, что i -е ограничение выполняется точно (как равенство) и весь ресурс b_i тратится полностью.

При малом изменении коэффициентов модели оптимальное решение задачи ЛП не меняется, такое свойство

называется *устойчивостью*. На рис. 12 показано, что для ЦФ1 решением является точка A, для ЦФ3 (при малом изменении коэффициентов c_j) также точка A, а для ЦФ2 решением уже будет точка B.

Существуют специальные программы, которые позволяют проанализировать пределы изменения коэффициентов c_j , при которых оптимальное решение не изменяется, а значение целевой функции меняется. Аналогично проводится анализ по правым частям b_i .

Современные алгоритмы ЛП способны с использованием современных компьютеров решать задачи ЛП очень большой размерности. Многие из них учитывают архитектуру компьютера.

Наиболее широко известны блочные методы ЛП. При большой размерности матрица A будет содержать много нулей. В этом случае задачу можно решать по блокам. Вот почему параметрами сложности решаемой задачи ЛП являются: *число переменных*, *число ограничений* и *число ненулевых коэффициентов* ограничений. Число итераций симплекс-метода, за которое находится решение, в большей степени зависит от числа ограничений. Современные программы ЛП имеют удобный интерфейс, позволяющий моделировать и решать небольшие задачи в удобной «математической» или табличной форме. Для решения задач ЛП большой размерности используются режимы работы с крупными базами исходных данных. Это, например, программа LINDO, а также приложение «Поиск решения» (SOLVER) программы Microsoft EXCEL.

ГЛАВА 3 ТРАНСПОРТНЫЕ ЗАДАЧИ (ТЗ)

Симплекс-метод является общим для любой задачи ЛП. Но существуют специальные задачи ЛП, которые имеют более простой и наглядный способ решения.

Постановка транспортной задачи.

Имеется m пунктов $A_1 \dots A_m$ отправления, в каждом из которых сосредоточено определенное количество груза ($a_1 \dots a_m$). Груз однородный, величина a_i называется *запасом*. Имеются n пунктов назначения $B_1 \dots B_n$, где требуется количество груза ($b_1 \dots b_n$); величина b_j — *заявка*. Известна стоимость перевозки единицы груза c_{ij} из i -го в j -й пункт.

Требуется так перевезти грузы, чтобы запасы не были превышены, заявки выполнены и общая стоимость перевозки была бы минимальной.

3.1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ТЗ ПО КРИТЕРИЮ СТОИМОСТИ

Введем дополнительное требование — так называемое условие правильного баланса:

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

Модель: x_{ij} — количество груза, перевозимого из i -го в j -й пункт:

$$L = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (3.1)$$

$$\left. \begin{aligned} \sum_{j=1}^n x_{ij} &= a_i, \quad i = \overline{1, m}; \\ \sum_{i=1}^m x_{ij} &= b_j, \quad j = \overline{1, n}; \\ x_{ij} &> 0, \quad \forall i, j, \end{aligned} \right\} \quad (3.2)$$

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (3.3)$$

Особенности модели ТЗ:

1. Наличие условия баланса (3.3) приводит к тому, что задача ТЗ всегда имеет решение.

2. Просуммировав левые части (3.2), получим уравнение (3.3), следовательно, одно уравнение в (3.2) линейно зависимо. Можно показать, что ранг системы (3.2) будет $n + m - 1$.

3. Коэффициенты при переменных в системе (3.2) все равны единице.

4. Подсчитаем число свободных и базисных переменных. Всего переменных $m \cdot n$, базисных $m + n - 1$. Свободных переменных будет $k = m \cdot n - m - n + 1 = (m - 1) \cdot (n - 1)$, т. е. практически все переменные свободные. Так как ТЗ является задачей ЛП, то в оптимальном решении все свободные переменные должны быть равны нулю, т. е. большинство маршрутов не будут задействованы. Если $m = 100$ и $n = 100$, всего маршрутов (переменных) 10 000, но не более 199 будут задействованы.

3.2. НАХОЖДЕНИЕ ОПОРНОГО ПЛАНА ТРАНСПОРТНОЙ ЗАДАЧИ

Все условия транспортной задачи можно записать в так называемой транспортной таблице. В ней выполняются и все необходимые вычисления.

Пример. В табл. 10 заданы условия ТЗ. Назовем x_{ij} «перевозкой».

Решение ТЗ также, как и ЛП, состоит из двух этапов:

Т а б л и ц а 10

I	B_1	B_2	B_3	a_i
A_1	4 5	1 6	3	11
A_2	2	5 7	2	7
A_3	3	7 3	4 5	8
A_4	2	4	5 4	4
b_j	5	16	9	30

Стоимость перевозки — стрелка на ячейку (1,1) с числом 4.
Перевозка (количество единиц в маршруте) — стрелка на ячейку (1,1) с числом 5.

1. Находится опорный план, т. е. допустимый план, соответствующий условию «вершины». В нем отличных от нуля перевозок не более $m + n - 1$.

2. Путем последовательного перехода от одного опорного плана к другому получаем оптимальный план.

Метод «северо-западного угла» (СЗУ) для нахождения опорного плана ТЗ. Он заключается в заполнении плана перевозок, начиная с самых верхних (северных) пунктов отправления, за счет самых левых (западных) пунктов назначения. Величина перевозки определяется выражением

$$x_{ij} = \min\{a'_i, b'_j\}, \quad (3.4)$$

где a'_i и b'_j — остатки запаса и заявки после предыдущей итерации. Этот способ обеспечивает линейную независимость столбцов. Свойство допустимого плана: сумма перевозок по строке равна запасу, а по столбцу — заявке. План называется невырожденным, если отличных от нуля перевозок ровно $m + n - 1$.

В приведенном примере $m + n - 1 = 4 + 3 - 1 = 6 = 6$, т. е. план невырожденный.

Выполненные действия показаны в табл. 10.

Определим стоимость плана:

$$L = 5 \cdot 4 + 6 \cdot 1 + 7 \cdot 5 + 3 \cdot 7 + 5 \cdot 4 + 4 \cdot 5 = 122.$$

Часто бывают такие транспортные задачи, где количество ненулевых перевозок (базисных клеток) меньше, чем $m + n - 1$.

Пример. В табл. 11 базисных клеток 4, а должно быть 5. Задача вырожденная. Будем дополнять план до невырожденного: в определенных местах про-

Таблица 11

	B_1	B_2	B_3	a_i
A_1	3	4	(0)	7
A_2			1	1
A_3			2	2
b_j	3	4	3	

ставим символическую бесконечно малую величину (0), чтобы выполнялись линейная независимость столбцов и условие невырожденности.

Чтобы автоматически определять место для дополнения плана, будем избавляться от вырожденности в процессе заполнения плана в соответствии с выражением (3.4). Вырожденность появляется тогда, когда в формуле (3.4) будет $a'_i = b'_j$. В этом случае будем считать, что в строке $a'_i = a'_i + (0)$.

Метод минимального элемента. В методе СЗУ мы не использовали стоимость перевозок. Существуют методы, которые позволяют получить первый план сразу близким к оптимальному (т. е. с достаточно малой общей стоимостью перевозок).

Метод минимального элемента состоит в том, что порядок заполнения плана соответствует возрастанию стоимости. Но при этом запас исчерпывается полностью, т. е. перевозка ставится в соответствии с формулой (3.4) в клетке с минимальной стоимостью в строке, если не исчерпали запас, или в столбце, если не выполнили заявку.

Метод не обязательно гарантирует лучшее решение, чем метод СЗУ, но, как правило, дает лучшие результаты.

Таблица 12

	B_1	B_2	B_3	B_4	a_i
A_1	5	4	3	1	9
A_2	2	4	2	9	3
A_3	8	7	1	3	8
A_4	4	2	3	2	5
b_j	4	6	5	10	25

Способ борьбы с вырождением такой же, как и в предыдущем методе.

Пример. Рассмотрим ТЗ (табл. 12). Начинаем с самого дешевого маршрута (A_1B_4), далее в столбце B_4 ищем самый дешевый (A_4B_4), затем в строке A_4 минимальная стоимость 2 в маршруте (A_4B_2) и т. д.

$$L = 9 \cdot 1 + 1 \cdot 2 + 2 \cdot 4 + 3 \cdot 8 + 5 \cdot 1 + 4 \cdot 2 + 1 \cdot 2 = 58.$$

3.3. ОПТИМИЗАЦИЯ ПЛАНА ТЗ, РАСПРЕДЕЛИТЕЛЬНЫЙ МЕТОД

Если в плане есть свободные клетки, для которых стоимость меньше, чем в базисных клетках, то можно передвинуть какую-либо перевозку в эту свободную клетку, но, чтобы баланс не нарушался, нужно из строки или столбца, где будет новая базисная клетка, убрать такое же количество груза. Будем ставить (+) там, где перевозка увеличивается, и (-) там, где уменьшается.

Таблица 13

	10	1
5	→	
	3	5
4	← 6	

Таблица 14

-	10	+	1
5			
+	3	-	5
4		6	

Пример. Рассмотрим фрагменты ТЗ, показанные в табл. 13, 14.

Необходим перенос по циклу. Назовем циклом замкнутый маршрут, который показывает перенос груза по некоторым клеткам таблицы. В цикле всегда четное число вершин.

Назовем ценой цикла γ алгебраическую сумму стоимостей перевозок c_{ij} , стоящих в вершинах цикла с соответствующими знаками. Цена показывает, насколько изменится стоимость перевозок в плане, если по данному циклу переносится единица груза. Если $\gamma < 0$, то стоимость изменяется. При переносе по циклу величины груза k стоимость уменьшается на $\Delta L = k \cdot \gamma$.

Будем рассматривать только правильные циклы: одна вершина в свободной клетке, а все остальные в базисных. В свободной клетке знак всегда «+», в остальных вершинах цикла знаки чередуются.

Для невырожденной транспортной таблицы с независимыми столбцами справедлива теорема о том, что для любой свободной клетки существует правильный цикл и при том только один.

Таким образом, для любой свободной клетки можно построить правильный цикл и вычислить его цену. Если она больше нуля, то переходим к другой свободной клетке. Если $\gamma < 0$, то строим цикл.

Очевидно, по циклу нужно перенести наибольшее количество груза. Эта величина k определяется как наименьшая из перевозок, стоящих в вершинах цикла со знаком «-».

Для приведенного далее (табл. 15, 16) примера цена цикла $1 - 5 + 3 - 10 = -11$. Величина $k = 5$. Значит, перенеся 5 единиц, мы изменяем общую стоимость на величину $5 \cdot (-11) = -55$ единиц.

Таблица 15

-	10	+	1
5			
+	3	-	5
4		6	

Таблица 16

	10		1
		5	
	3		5
9		1	

Перенос груза по циклу соответствует симплекс-преобразованию.

Метод непосредственного отыскания циклов с отрицательной ценой называется распределительным. Рассмотрим пример решения ТЗ этим методом.

Пример. В табл. 17–20 показаны действия для распределительного метода.

Для табл. 17

$$m + n - 1 = 3 + 4 - 1 = 6;$$

$$L = 2 \cdot 4 + 1 \cdot 2 + 3 \cdot 4 + 2 \cdot 3 + 1 \cdot 1 + 5 \cdot 4 = 49.$$

Таблица 17

T_1	B_1	B_2	B_3	B_4	a_i
A_1	4	2	1	1	3
A_2	5	4	3	2	5
A_3	1	2	1	4	6
b_j	2	4	3	5	

Таблица 18

T_2	B_1	B_2	B_3	B_4	a_i
A_1	4	2	1	1	3
A_2	5	4	3	2	5
A_3	1	2	1	4	6
b_j	2	4	3	5	

Строим правильный цикл для свободной клетки (3, 1).

$$\gamma_{31} = 1 - 4 + 2 - 4 + 3 - 1 = -3;$$

$$k_{31} = 1; \Delta L = -3.$$

Для табл. 18:

$$L = 4 + 4 + 8 + 9 + 1 + 20 = 46.$$

Строим цикл для клетки (1, 4), получаем табл. 19, далее строим цикл для клетки (3, 3) и переходим к табл. 20. В табл. 20 строим цикл для клетки (2, 4).

$$\gamma_{24} = 2 - 4 + 1 - 3 = -4;$$

$$k_{24} = 1; \Delta L = -4; L = 34.$$

Дальнейшее решение предлагается проделать самостоятельно.

Очевидно, для того чтобы процесс остановился, необходимо проверить все свободные клетки и убедиться, что цена циклов для каждой такой клетки больше или равна нулю. С учетом того, что количество свободных клеток равно $(m - 1) \cdot (n - 1)$, этот процесс очень трудоемок.

Таблица 19

T_3	B_1	B_2	B_3	B_4	a_i
A_1	4	2	1	1	3
		2	—	1	+
A_2	5	4	3	2	5
		2	+	3	—
A_3	1	2	1	4	6
	2		+	4	—
b_j	2	4	3	5	

Таблица 20

T_4	B_1	B_2	B_3	B_4	a_i
A_1	4	2	1	1	3
				3	
A_2	5	4	—	3	2
		4	1		+
A_3	1	2	+	1	4
	2		2	2	—
b_j	2	4	3	5	

Недостатком распределительного метода является необходимость перебирать циклы до встречи цикла с отрицательной ценой. Для доказательства оптимальности нужно перебрать и построить правильные циклы для всех свободных клеток. Если $n = 100$, $m = 100$, то свободных клеток будет 9801.

Рассмотрим метод, в котором автоматически находятся циклы с отрицательной ценой.

3.4. МЕТОД ПОТЕНЦИАЛОВ РЕШЕНИЯ ТЗ

Пусть каждый пункт отправления A_i платит за вывоз единицы груза платеж α_i . И каждый пункт назначения B_j платит за привоз единицы груза β_j . Эти платежи передаются посреднику. Таким образом, на маршруте ij он получает $\alpha_i + \beta_j = \tilde{c}_{ij}$. Назовем эту величину *псевдостоимостью*. Оказывается, что суммарная псевдостоимость плана перевозок

закон есть величина постоянная (не меняется с изменением плана). В этом заключается *теорема о платежах*.

Для доказательства запишем формулу псевдостоимости и перегруппируем слагаемые:

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

Теорема о потенциальном плане ТЗ.

Пусть в базисных клетках стоимость перевозок равна псевдостоимости перевозок, а в свободных клетках стоимость не меньше псевдостоимости. Тогда такой план является оптимальным и улучшен быть не может

$$\begin{aligned}c_{ij} &= \tilde{c}_{ij}, \quad x_{ij} > 0, \\ c_{ij} &\geq \tilde{c}_{ij}, \quad x_{ij} = 0.\end{aligned}\quad (*)$$

Доказательство:

$$L^* = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^*.$$

Знак (*) означает, что выполняется условие (*). Необходимо показать, что стоимость плана X^* является нижней границей и определяется только исходными данными задачи. Действительно,

$$L^* = \sum \sum_{\{x_{ij}\}} \tilde{c}_{ij} x_{ij}^* = \text{const},$$

так как в базисных клетках $c_{ij} = \tilde{c}_{ij}$, а в свободных клетках $x_{ij}^* = 0$.

Перейдем к другому произвольному плану:

$$\{x_{ij}^*\} \rightarrow \{x'_{ij}\}: \quad L' = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x'_{ij}.$$

В этом плане перевозка из базисной клетки попала в свободную, где $c_{ij} \geq \tilde{c}_{ij}$, поэтому

$$L' \geq L^* = \sum_{i=1}^m \sum_{j=1}^n \tilde{c}_{ij} x_{ij}^* = \text{const},$$

т. е. любое изменение плана от $\{x_{ij}^*\}$ приводит к увеличению стоимости, следовательно, план $\{x_{ij}^*\}$ является оптимальным.

Идея метода потенциалов основывается на этой теореме и заключается в следующем:

1. Составим любой опорный план, и если это необходимо, то дополним до невырожденного.

2. Потребуем в базисной клетке $c_{ij} = \tilde{c}_{ij}$. Запишем в левом верхнем углу базисной клетки величины псевдостоимостей, равные стоимостям, стоящим в правом верхнем углу. Таких клеток $m + n - 1$.

3. Находим неизвестные нам платежи α_i и β_j из условий, что их сумма известна для каждой базисной клетки:

$$\left. \begin{array}{l} \alpha_i + \beta_j = c_{ij}, \\ \text{неизв.} - \text{изв.}, \\ \dots \end{array} \right\} m + n - 1.$$

В такой системе в каждой строке два неизвестных, а всего их $m + n$. Так как равенств всего $m + n - 1$, то решений бесконечное множество. Поэтому будем брать произвольное α_1 и полагать, например, $\alpha_1 = 0$. Тогда можно начать решение со строки, где содержится α_1 , и так, последовательно переходя в строки с известным платежом, найти все платежи.

4. Найдя все платежи, подсчитываем $\alpha_i + \beta_j = \tilde{c}_{ij}$ для свободных клеток.

5. Проверяем условие (*) для свободных клеток. Там, где $\tilde{c}_{ij} > c_{ij}$, свободная клетка дает цикл с отрицательной ценой, так как $\gamma_{ij} = c_{ij} - \tilde{c}_{ij}$.

6. Выбираем цикл с наименьшим (наиболее отрицательным) γ_{ij} и составляем цикл. По циклу переносим k единиц груза и т. д. до выполнения условия (*). Величина k определяется так же, как и в рассмотренном выше распространенном методе.

Пример. Опорный план построим по методу минимального элемента, вычисляем платежи и определяем псевдостоимости в свободных клетках (табл. 21). Видим, что в клетке (3,2) цена цикла отрицательная. Строим цикл, переходим в табл. 22.

Таблица 21

	B_1	B_2	B_3	B_4	a_i	α_i
A_1	-2 4	3 2	-2 1	1 1 3	3	0
A_2	-1 5	4 4 4	-1 3	2 2 1 +	5	1
A_3	1 1 2	6 2 +	1 1 3	4 4 1 -	6	3
b_j	2	4	3	5		
β_j	-2	3	-2	1		

Таблица 22

	B_1	B_2	B_3	B_4	a_i	α_i
A_1	2 4	3 2 +	2 1	1 -1 3 +0	3	0
A_2	3 5	4 4 3 -	3 3	2 2 2 +	5	1
A_3	1 1 2	2 2 1	1 1 3	0 4	6	-1
b_j	2	4	3			
β_j	2	3	2	1		

Таблица 23

	B_1	B_2	B_3	B_4	a_i	α_i
A_1	1 4	2 2 3	1 1	1 1 0	3	0
A_2	2 5	3 4	2 3	2 2 5	5	1
A_3	1 1 2	2 2 1	1 1 3	1 4	6	0
b_j	2	4	3	5		
β_j	1	2	1	1		

$$\gamma_{32} = -4, \\ k_{32} = 1; \Delta L = -3.$$

Заметим, что если при вычислении k минимальных перевозок две и более, то при переносе по циклу появятся дополнительные свободные клетки, т. е. план вырождается. В этом случае считаем, что в одной такой клетке величина перевозки k , а в других $k + (0)$. Ноль лучше поставить там, где дешевле маршрут (табл. 23).

План табл. 23 оптимальный, так как выполняется условие (*).

3.5. РЕШЕНИЕ ТЗ С НЕПРАВИЛЬНЫМ БАЛАНСОМ

До сих пор мы считали, что в ТЗ выполняется условие правильного баланса, т. е. $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$. На практике это никогда не выполняется. Но можно ТЗ с неправильным балансом привести к правильному балансу и показать, что оптимальный план, найденный для ТЗ с правильным балансом, является оптимальным и для ТЗ с неправильным балансом.

Рассмотрим два случая:

$$1. \text{ ТЗ с избытком запаса: } \sum_{i=1}^m a_i > \sum_{j=1}^n b_j.$$

Будем считать, что излишки запасов якобы перевозятся в некоторый фиктивный пункт назначения.

Чтобы учесть фиктивность перевозки, достаточно считать, что стоимость перевозки в фиктивный пункт B_Φ равна нулю, $c_{i\Phi} = 0$. Тогда, если какие-то перевозки $x_{i\Phi} > 0$, то $\sum_{n=1}^m c_{i\Phi} E_{i\Phi} = 0$.

Таким образом, чтобы решать ТЗ с избытком запаса, нужно к таблице прибавить дополнительный столбец B_Φ , заявка которого $b_\Phi = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$, а $c_{i\Phi} = 0, i = \overline{1, m}$. Тогда для новой таблицы будет выполняться условие правильного баланса. После этого задача решается любым способом.

$$2. \text{ ТЗ с избытком заявок: } \sum_{i=1}^m a_i < \sum_{j=1}^n b_j.$$

Необходимо ввести фиктивный пункт отправления (A_Φ),

в котором якобы есть недостающий запас $a_\Phi = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$,

при этом $c_{j\Phi} = 0, j = \overline{1, n}$. После этого задача решается обычным способом.

Заметим, что при таком подходе в итоговом оптимальном плане не все заявки будут выполнены.

Иногда задача решается по-другому.

Пусть $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$. Тогда можно вычислить коэффициент дефицита:

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

После этого каждую заявку можно скорректировать:

$$b'_j = b_j k, \\ \sum_{j=1}^n b'_j = \sum_{i=1}^m a_i.$$

При этом можно решать задачу с учетом приоритетов на заявку. Величина приоритетов j -го пункта назначения (d_j) может быть от 0 до 1 ($0 \leq d_j \leq 1$):

$$\sum_{j=1}^n d_j = 1.$$

Чем меньше d_j , тем выше приоритет.

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

где Δ — величина дефицита.

$$b'_j = b_j - d_j \Delta.$$

Пример. Рассмотрим ТЗ с избытком запасов (см. табл. 24) ($58 < 68$), поэтому введем фиктивный столбец, найдем опорный план методом минимального элемента, а далее находим цикл с отрицательной ценой (2,1).

Т а б л и ц а 24

T_1	B_1	B_2	B_3	a_i	B_Φ
A_1	6	5	1	35	0
		11	14		10
A_2	4	3	2	21	0
	+	21	—		
A_3	7	3	4	12	0
	8	—	4	+	
b_j	8	36	14		10

Т а б л и ц а 25

T_2	B_1	B_2	B_3	B_Φ	A_i	α_i
A_1	6 6	5 5	1 1	0 0	35	0
		11	14	10		
A_2	4 4	3 3	—1 2	—2 0	21	—2
	8	13				
A_3	4 7	3 3	—1 4	—2 0	12	—2
		12				
b_j	8	36	14	10		
β_j	6	5	1	0		

Если цикл с отрицательной ценой виден сразу, то не обязательно применять метод потенциалов. По таблице находим $k = 8, \gamma = -3$.

Для табл. 25 применим метод потенциалов. Видно, что во всех свободных клетках псевдостоимости $\leq c_{ij}$, следовательно, нет цикла с отрицательной ценой. План — оптимальный.

3.6.

ТЗ ПО КРИТЕРИЮ ВРЕМЕНИ

Пусть в ТЗ задано время t_{ij} , за которое груз можно перевезти из i -го в j -й пункт. Оно не зависит от количества перевозимого груза. Заданы запасы a_i , заявки b_j и необходимо составить такой план перевозок, чтобы общее время окончания перевозок было бы минимальным.

Так как перевозки осуществляются параллельно, окончание всех перевозок определяется длительностью маршрута с самой большой величиной t_{ij} среди занятых маршрутов (базисных клеток).

Таким образом, критерий в нашей задаче

$$T = (\max_{(i,j): x_{ij} > 0} t_{ij}) \rightarrow \min. \quad (3.5)$$

Ограничения те же:

$$\left. \begin{aligned} \sum_{i=1}^m x_{ij} &= b_j, \quad j = \overline{1, n}; \\ \sum_{j=1}^n x_{ij} &= a_i, \quad i = \overline{1, m}; \end{aligned} \right\} \quad (3.6)$$

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (3.7)$$

Здесь (3.5)–(3.7) — математическая модель ТЗ по критерию времени. Заметим, что целевая функция $T(x_{ij})$ имеет нелинейный вид. При изменении плана $\{x_{ij}\}$ критерий T может не меняться. Этот критерий требует оптимизации по так называемому узкому месту. В этом смысле рассмотренный в предыдущих параграфах критерий общей суммарной стоимости перевозок является *интегральным*. Таким образом ТЗ по критерию стоимости — это пример задачи с интегральным критерием, а ТЗ по критерию времени — пример задачи с минимаксным критерием.

Так как целевая функция $T(x_{ij})$ имеет нелинейный вид, то для решения ТЗ по критерию времени нельзя применить теорию линейного программирования. Для ее решения можно построить последовательность нескольких задач линейного программирования.

Мы рассмотрим более простой эвристический алгоритм.

Эвристическими называют такие алгоритмы, которые обеспечивают разумность поиска оптимального решения, но не основываются на строгой математической теории и в связи с этим не вполне гарантируют результат (не доказана сходимость к оптимальному решению, не изучена точность и т. п.). Тем не менее *эвристические* процедуры

чрезвычайно распространены, например при игре в шахматы, при составлении компьютерных программ и т. д.

Метод запрещенных клеток.

1. Находим любой опорный план (необязательно невырожденный).

2. Вычисляем величину критерия для полученного плана

$$T_1 = T(x_{ij}^1).$$

3. Запрещаем все маршруты в свободных клетках, у которых время $t_{ij} \geq T_1$.

4. Зафиксируем в плане занятую клетку с максимальным t_{ij} (узкое место).

5. Ищем цикл, при помощи которого можно «разгрузить» эту клетку, и переносим по этому циклу величину груза k . Циклы обязательно должны быть правильными.

Таблица 26

T_1	B_1	B_2	B_3	a_i
A_1	— 3 19	1 1	0 +	20
A_2	2 +	1 10	2 30 —	40
A_3	3	2	1 15	15
b_j	19	11	45	

$$T_1 = 3$$

Таблица 27

T_2	B_1	B_2	B_3	a_i
A_1	3	1 1	0 19	20
A_2	2 19	1 10	2 11	40
A_3	3	2	1 15	15
b_j	19	11	45	

$$T_2 = 2$$

ми. Причем может быть, что освободить такую клетку можно только за несколько циклов.

6. Получаем план с новым значением

$$T_2 = T(x_{ij}^2).$$

Запрещаем свободные клетки, для которых $t_{ij} \geq T_2$, и переходим к п. 4.

Повторяем до тех пор, пока будет невозможно составить ни одного цикла, разгружающего занятую клетку с максимальным временем.

Пример. Для нижеприведенной ТЗ по критерию времени (табл. 26) составляем первый опорный план, получаем $T_1 = 3$. Запрещаем все свободные клетки, у которых $t_{ij} \geq 3$. Используем неправильный цикл, получаем новый план, в котором $T_2 = 2$ (табл. 27).

Видно, что перевозку со временем $t_{ij} = 2$ уже нельзя перенести ни в одну строку, ни в один столбец, поэтому табл. 27 является оптимальной.

ГЛАВА 4

ДИСКРЕТНОЕ ПРОГРАММИРОВАНИЕ

При моделировании различных задач требуется, чтобы какие-то управляемые переменные принадлежали специальному множеству:

1. Множество целых чисел — число людей, датчиков, самолетов и т. д.

2. Множество дискретных значений — это стандартные размеры деталей, возможные мощности электродвигателей и т. п.

В таких задачах приходится искать значения переменных, которые не являются любыми действительными числами. Искомые переменные должны принадлежать заранее зафиксированному множеству дискретных значений D .

Такого рода задачи принадлежат классу задач *дискретного программирования*:

$$k = f(x) \rightarrow \min, \quad (4.1)$$

$$\varphi_i(x) \leq 0, \quad i = \overline{1, m}, \quad (4.2)$$

$$x \in D. \quad (4.3)$$

В связи с чрезвычайной сложностью решения таких задач в общем виде мы рассмотрим только наиболее важные методы:

1. Универсальные методы решения задач линейного целочисленного программирования на основе «классических» алгоритмов.

2. Специальные модели и методы дискретной оптимизации, которые имеют особое практическое значение.

Задачей линейного целочисленного программирования (ЛЦП) называют задачу вида

$$L = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad (4.4)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad (4.5)$$

$$x_j \in Z, \quad j = \overline{1, n}. \quad (4.6)$$

где Z — множество целых чисел.

Если только часть переменных должны быть целыми, то задача называется частично целочисленной. Тогда в формуле (4.6) вместо n будет $n_1 < n$.

Во многих задачах дискретность проявляется при моделировании задачи. Это относится к так называемым булевым переменным. В этом случае переменные принадлежат множеству $B = \{0, 1\}$. Ограничения также могут иметь логический характер. Но при этом все действия с этими переменными, определяемые математической моделью, подчиняются обычным правилам арифметики (т. е., например, $1 + 1 = 2$, а не 0, как в булевой алгебре). Таким образом, вместо формулы (4.6) или в дополнение к ней в модели может присутствовать требование

$$x_j \in \{0, 1\}. \quad (4.7)$$

Если все переменные булевы, то задача (4.4), (4.5), (4.7) — задача линейного *булева программирования* (ЛБП).

К этим задачам иногда сводят задачи целочисленного и дискретного программирования, так как для ЛБП разработаны эффективные быстродействующие алгоритмы (в том числе и с «аппаратной поддержкой»).

Чтобы привести задачу ЛЦП к ЛБП, достаточно каждую переменную представить в виде

$$x_j = 2^0 x_{j1} + 2^1 x_{j2} + 2^2 x_{j3} + \dots + 2^k x_{jk+1}, \\ x_{jk} \in \{0, 1\}.$$

Если x_j ограничено, то можно найти k , т. е. количество двоичных переменных, которыми можно описать исходные целочисленные переменные.

Задачу дискретного программирования можно привести к задаче булева программирования так. Пусть

$$x_j \in \{k_{j1}, k_{j2}, k_{j3}, \dots, k_{jp}\}.$$

Тогда можно ввести булевы переменные y_r и заменить $x_j = y_1 k_{j1} + y_2 k_{j2} + \dots + y_p k_{jp}$, $y_r \in \{0, 1\}$, $r = \overline{1, p}$; $\sum_{r=1}^p y_r = 1$.

4.1. ОСОБЕННОСТИ ЗАДАЧ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ

Рассмотрим геометрический образ задачи ЛЦП для двух переменных (рис. 13).

ОДР этой задачи представляет собой только целочисленные точки внутри ОДР соответствующей задачи ЛП.

Решение задачи ЛП без требования целочисленности будет находиться в точке А (рис. 13).

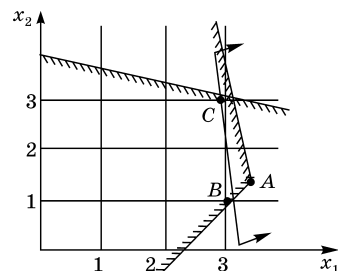


Рис. 13
Геометрическая интерпретация примера решения задачи линейного целочисленного программирования

Возникает вопрос, нельзя ли найти решение задачи ЛП (назовем его «непрерывным» решением), а затем округлить до ближайшего целого? Покажем, что ответ на этот вопрос отрицательный.

1. При округлении решения до ближайшего целого может быть получено недопустимое решение.
2. Даже если при округлении решение является допустимым, может оказаться, что существует другое допустимое целочисленное решение, которое лучше округленного. Из рис. 13 видно, что ближе к непрерывному решению А целочисленная допустимая точка В, оптимальная точка С. Оптимальное решение: $x^* = (3, 3)$. Иногда кроме целочисленности переменной требуется, чтобы целой была величина целевой функции. Тогда тем более округление не дает правильного результата.

Пример.

$$z = \frac{1}{6}x_1 + x_2 \rightarrow \max,$$

$$-\frac{1}{2}x_1 + x_2 \leq 2, \quad \frac{1}{5}x_1 + x_2 \leq 3, \quad x_1 - \frac{1}{3}x_2 \leq 4.$$

О т в е т:

$$x_1 = 1\frac{3}{7}, \quad x_2 = 2\frac{5}{7}.$$

Округлив эти значения, получаем $x_1 = 1$, $x_2 = 3$ — это не удовлетворяет решению!

О т в е т (целочисленный): $x_1 = 4$, $x_2 = 2$.

Чтобы z было целым: $x_1 = 0$, $x_2 = 0$.

Таким образом, для нахождения целочисленных оптимальных решений требуется непосредственно находить их на множестве целых (дискретных) чисел.

4.2. ПРИМЕРЫ МОДЕЛЕЙ ЗАДАЧ ДИСКРЕТНОГО ПРОГРАММИРОВАНИЯ

Заметим, что обычная ТЗ при целых коэффициентах a_i , b_i дает целочисленное решение. Рассмотрим некоторые наиболее распространенные задачи дискретного программирования.

1. Задача о назначениях (о выборе).

Имеется n работ и n кандидатов на эти работы. Известен эффект того, что i -я работа будет занята j -м работником c_{ij} . Необходимо распределить работы по работникам так, чтобы суммарный эффект был максимальным.

Управляемые переменные:

$$x_{ij} = \begin{cases} 1, & i \rightarrow j, \\ 0, & i \not\rightarrow j. \end{cases} \quad (4.8)$$

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max. \quad (4.9)$$

Так как каждая работа может быть занята только одним работником и каждый работник может занимать только одну работу, то

$$\left. \begin{aligned} \sum_{j=1}^n x_{ij} &= 1, \quad i = \overline{1, n}; \\ \sum_{i=1}^n x_{ij} &= 1, \quad j = \overline{1, n}. \end{aligned} \right\} \quad (4.10)$$

Задача (4.8)–(4.10) принадлежит к классу задач ЛБП. Заметим, что она является также частным видом ТЗ. Любой опорный план будет сильно вырожден — в нем будет только n отличных от нуля базисных переменных (в невырожденном плане ТЗ должно быть $2n - 1$). Эту задачу можно свести к выбору n нулевых элементов, по одному в каждом столбце и каждой строке («правильный выбор») в некоторой матрице с неотрицательными элементами, у которой в каждой строке и каждом столбце есть нули.

Нам удобно будет предварительно перейти от данной задачи выбора на максимум к задаче выбора с теми же условиями, но на минимум, т. е. от матрицы $C = (c_{ij})$ перейти к матрице $-C = (-c_{ij})$ и искать выбор, дающий минимальную сумму элементов.

Теперь перейдем от задачи на минимум с матрицей $-C$ к задаче на минимум с эквивалентной ей матрицей, которая имела бы только неотрицательные элементы, и в каждом столбце, каждой строке которой было хотя бы по одному нулевому элементу. Для этого сначала прибавим к каждому столбцу матрицы $-C$ наибольший из элементов соответствующего столбца матрицы C (или, что то же, вычтем элементы каждого столбца матрицы C из наибольшего элемента этого столбца). Получится неотрицательная матрица C_1 , в каждом столбце которой есть хотя бы один нуль. Теперь вычтем из каждой строки матрицы $-C_1$ минимальный элемент этой строки. Полученная матрица D и будет неотрицательной матрицей, в каждом столбце и каждой строке которой есть хотя бы один нуль.

Итак, преобразуем следующие два последовательных преобразования матрицы $-C$:

$$\begin{aligned} C &= (c_{ij}) \xrightarrow{(1)} C_1 = (c'_{ij}) = (\max_i c_{ij} - c_{ij}) \xrightarrow{(2)} \\ &\rightarrow D = (d_{ij}) = (c'_{ij} - \min_j c'_{ij}). \end{aligned}$$

Назовем их *предварительными преобразованиями*. Если в каждой строке матрицы $C = c'_{ij}$, полученной после первого преобразования матрицы C , есть хотя бы один нуль, то второе преобразование, конечно, не изменит матрицу C и практически не производится.

Наименьшее возможное значение суммы n элементов неотрицательной матрицы равно, очевидно, нулю. Таким образом, наша задача сводится теперь к выбору в матрице D или в эквивалентной ей матрице с неотрицательными элементами n нулевых элементов — по одному в каждой строке и каждом столбце. Покажем, как это можно сделать. Неформальный смысл приводимого ниже алгоритма заключается в последовательных переходах от одного правильного неполного выбора нулей к другому, содержащему на один нуль больше, чем предыдущий, до тех пор, пока не получится полный правильный выбор. При этом на отдельных этапах может потребоваться переход к новой матрице, эквивалентной предыдущей.

Приведем алгоритм решения задачи о назначениях. Пусть уже проделаны предварительные преобразования матрицы эффективности C данной задачи и получена неотрицательная матрица D , содержащая хотя бы по одному нулевому элементу в каждой строке и в каждом столбце.

1. Отмечаем (например, звездочкой) какой-нибудь нуль в первом столбце матрицы $D(0^*)$; отмечаем звездочкой какой-нибудь нуль во втором столбце, не лежащий в той строке, в которой находится 0^* из первого столбца (если такой нуль во втором столбце найдется); отмечаем звездочкой один из нулей третьего столбца, лежащий в строке, где нет еще нулей со звездочкой (если такой нуль в третьем столбце найдется); и так далее, пока не пройдем все столбцы матрицы.

Если число отмеченных звездочкой нулей равно n , то процесс окончен: места, занимаемые нулями со звездочкой, соответствуют n переменным x_{ij} , равным 1 в оптимальном решении исходной задачи.

Если нулей со звездочкой меньше n , то переходим к п. 5.

2. Помечаем (например, знаком «+» сверху) столбцы матрицы, в которых есть 0^* , и считаем эти столбцы занятыми.

В ходе процесса будут появляться и занятые строки. Элементы, стоящие на пересечении незанятого столбца и незанятой строки, будем считать незанятыми, остальные элементы — занятыми.

Если в матрице нет незанятых нулей, то переходим к п. 5.

Если незанятые нули есть, то выбираем первый из них (просматривая поочередно строки матрицы слева направо). Отмечаем его каким-нибудь промежуточным значком (например, штрихом — $0'$). Если в его строке нет нуля со звездочкой, то переходим к п. 4; если в его строке 0^* есть, то переходим к п. 3.

3. Освобождаем, т. е. снимаем знак «+» и считаем снова незанятым столбец, в котором находится 0^* , лежащий в той же строке, что и отмеченный только что штрихом ноль. Помечаем (например, знаком «+» справа) строку, в которой находится наш $0'$, и считаем ее занятой. Возвращаемся ко второй части п. 2 (третий абзац п. 2).

4. Начиная с только что отмеченного $0'$, строим цепочку из нулей: от этого $0'$ по столбцу к 0^* , от него по строке к $0'$ и т. д., пока это возможно. Цепочка оборвется (возможно, на первом же $0'$) на некотором $0'$. Снимаем звездочки у нулей из цепочки и заменяем звездочками штрихи у нулей из цепочки. Новый набор нулей со звездочками содержит на один ноль больше, чем предыдущий, и является также правильным.

Снимаем все пометки, кроме звездочек, и возвращаемся ко второй части п. 1 (второй абзац п. 1).

5. Отыскиваем минимальный элемент среди незанятых элементов матрицы (пусть он равен h) и вычитаем его из всех незанятых строк, а затем прибавляем ко всем занятым столбцам. Никакие пометки при этом не снимаются. Получается матрица, эквивалентная предыдущей и содержащая незанятые нули. Возвращаемся к третьей части п. 2 (четвертый абзац п. 2).

Рассмотрим пример использования алгоритма.

Пример. Найти оптимальный вариант назначений, если матрица эффективностей такова:

$$\begin{pmatrix} 2 & 3 & 3 & 5 & 4 \\ 4 & 2 & 4 & 6 & 2 \\ 2 & 2 & 2 & 4 & 3 \\ 4 & 3 & 4 & 3 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{pmatrix}.$$

Приводим цепочку матриц, получающихся в процессе решения задачи с соответствующими пометками. Снятие значка отмечено включением его в прямоугольник. Над стрелками переходов от матрицы указаны пункты алгоритма, которые использовались при соответствующих преобразованиях:

$$\begin{aligned} & \begin{pmatrix} 2 & 3 & 3 & 5 & 4 \\ 4 & 2 & 4 & 6 & 2 \\ 2 & 2 & 2 & 4 & 3 \\ 4 & 3 & 4 & 3 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{pmatrix} \xrightarrow{\text{ПРЕДВАРИТ. ПРЕОБР. (1)}} \begin{pmatrix} 2 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 3 \\ 2 & 1 & 2 & 2 & 2 \\ 0 & 0 & 0 & 3 & 0 \\ 4 & 2 & 4 & 4 & 5 \end{pmatrix} \xrightarrow{\text{ПРЕДВАРИТ. ПРЕОБР. (2) П. 1, П. 2, П. 3}} \\ & \begin{pmatrix} 2 & 0^* & 1 & 1 & 1 \\ 0^* & 1 & 0 & 0' & 3 \\ 1 & 0 & 1 & 1 & 1 \\ 0' & 0 & 0^* & 3 & 0 \\ 2 & 0 & 2 & 2 & 3 \end{pmatrix} \xrightarrow{\text{П. 5 (h=1), П. 2, П. 3, П. 4}} \begin{pmatrix} 1 & 0^* & 0' & 0 & 0 \\ 0^* & 1 & 0 & 0^* & 3 \\ 0^* & 0 & 0 & 0 & 0 \\ 0' & 1 & 0^* & 3 & 0 \\ 1 & 0 & 1 & 1 & 2 \end{pmatrix} \xrightarrow{\text{П. 1, П. 2, П. 3, П. 4}} \\ & \begin{pmatrix} 1 & 0^* & 0 & 0 & 0^* \\ 0 & 2 & 0 & 0^* & 3 \\ 0^* & 0' & 0 & 0 & 0 \\ 0' & 1 & 0^* & 3 & 0 \\ 1 & 0^* & 1 & 1 & 2 \end{pmatrix} \xrightarrow{\text{РЕЗУЛЬТАТ}} \begin{pmatrix} 1 & 0 & 0 & 0 & 0^* \\ 0 & 2 & 0 & 0^* & 3 \\ 0^* & 0 & 0 & 0 & 0 \\ 0 & 1 & 0^* & 3 & 0 \\ 1 & 0^* & 1 & 1 & 2 \end{pmatrix}. \end{aligned}$$

Процесс окончен, так как получилось $n = 5$ нулей со звездочкой. Оптимальный вариант назначений: $x_{15} = x_{24} = x_{31} = x_{43} = x_{52} = 1$, остальные $x_{ij} = 0$, т. е. первый механизм

назначается на пятую работу, второй — на четвертую, третий — на первую, четвертый — на третью, пятый — на вторую.

Изложенный алгоритм получил в литературе название венгерского (венгерский метод) решения задачи о назначениях. Если в исходной задаче матрица C не квадратная, то необходимо дополнить ее до квадратной нулевыми элементами.

4.2.1. ЗАДАЧА О ПОКРЫТИИ

Примером такой задачи является задача об определении минимального числа спутников, которые бы обеспечивали устойчивую связь для заданных городов. Другой пример — это задача о переводчиках и туристических группах. В группе известны языки, которые они понимают, а у переводчиков — какие языки знают. Нужно обеспечить все группы минимальным количеством переводчиков.

Классическая задача о покрытии формулируется следующим образом.

Задан граф с ребрами и узлами. Найти такое подмножество ребер, которое инцидентно любой вершине этого графа и при этом, чтобы количество ребер в этом подмножестве (называемым покрытием) было бы минимальным. Тогда если j — номер ребра, а Π — множество ребер покрытия, то

$$x_j = \begin{cases} 1, & j \in \Pi, \\ 0, & j \notin \Pi. \end{cases}$$

Построим математическую модель. Целевая функция имеет вид

$$z = \sum_{j=1}^n x_j \rightarrow \min. \quad (4.11)$$

A — матрица инцидентности исходного графа, элементы которой

$$a_{ij} = \begin{cases} 1, & \text{вершина } i \text{ инцидентна ребру } j, \\ 0, & \text{в противном случае.} \end{cases}$$

Каждая вершина должна быть инцидентна хотя бы одному ребру покрытия

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = \overline{1, m}. \quad (4.12)$$

Таким образом, мы получили задачу ЛБП (4.11), (4.12). Решение такой задачи рассмотрим ниже.

4.2.2. ЗАДАЧА О КОММИВОЯЖЕРЕ

Задана сеть городов с расстояниями между ними. Торговец должен обойти все города, вернувшись в начальный город, по минимальному пути. Необходимо найти порядок обхода городов. Эта ставшая классической задача находит чрезвычайно широкое применение при оптимизации порядка выполнения операций на станках, выполнении заданий операционной системой компьютера и т. д.

Одна из математических моделей этой задачи имеет вид

$$x_{ij} = \begin{cases} 1, & i \rightarrow j, \\ 0, & i \nrightarrow j. \end{cases} \quad (4.13)$$

Здесь $i \rightarrow j$ означает, что из города i осуществляется переход в город j . Пусть c_{ij} — расстояние между городами i и j ; может быть, что $c_{ij} \neq c_{ji}$. Длина пути должна быть минимальной, поэтому

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min. \quad (4.14)$$

Из каждого города торговец должен выйти один раз и лишь один раз войти, поэтому

$$\left. \begin{aligned} \sum_{j=1}^n x_{ij} &= 1, \quad i = \overline{1, n}; \\ \sum_{i=1}^n x_{ij} &= 1, \quad j = \overline{1, n}; \end{aligned} \right\} \quad (4.15)$$

Необходимо вернуться в начальный город. Это условие цикличности можно записать так

$$u_i - u_j + n x_{ij} \leq n - 1, \quad (1 \leq i \neq j \leq n), \quad (a)$$

где u_i и u_j — произвольные целые и неотрицательные числа.

Поясним условие (а). Пусть имеется другой цикл μ с числом ребер $k \leq n$, не проходящий через начальный город. Складывая все неравенства (а) при $x_{ij} = 1$ вдоль цикла μ , получаем противоречивое неравенство: $nk \leq (n-1)k$, так как все разности $u_i - u_j$ уничтожаются. Таким образом, условие (а) не допускает цикла, не проходящего через начальный город.

Покажем, что для любого цикла, начинающегося из начального города, можно найти u_i , удовлетворяющие условию (а).

Пусть $u_i = p$, если коммивояжер посещает город i на p -м этапе. Отсюда следует, что $u_i - u_j \leq p-1$ для всех i и j , и, таким образом, условие выполняется при $x_{ij} = 0$. При $x_{ij} = 1$ условие (а) выполняется как строгое равенство:

$$u_i - u_j + nx_{ij} = p - (p+1) + n = n-1.$$

Условие цикличности маршрута (а) приводит к тому, что задача коммивояжера имеет резко выраженные комбинаторные свойства и принадлежит к классу чрезвычайно сложных «переборных» задач. При некоторых коэффициентах c_{ij} решение этой задачи приводит к полному перебору вариантов. Далее мы рассмотрим общий метод решения подобных задач, а также алгоритм и пример решения задачи коммивояжера (см. п. 4.6).

4.2.3. ЗАДАЧА О РАСКРОЕ МАТЕРИАЛА

Эта задача впервые была поставлена и изучена нобелевским лауреатом Л. В. Канторовичем. Задачу о раскрое материалов необходимо рассматривать для производства комплектов продукции.

Заданы заготовки некоторого материала в виде нескольких партий различного размера; a_i — количество заготовок i -й партии (i -я партия имеет свой размер $i = 1, m$). Под размером можно понимать в общем случае различные величины, в простейшем случае «стержневого раскроя» это длина стержня.

Из заготовок необходимо выкроить детали (рис. 14). Всего деталей n , $j = 1, n$.

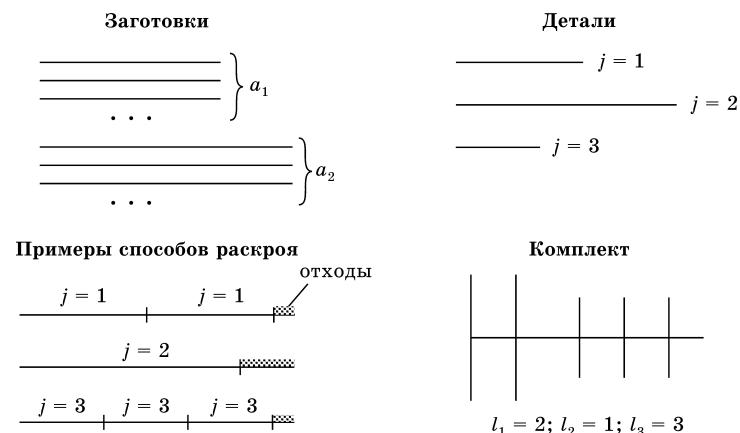


Рис. 14

Заготовки и детали в задаче оптимального раскроя материалов

Из деталей делается комплект. Известно количество деталей j -го типа, входящих в комплект l_j .

Необходимо так раскроить все заготовки, чтобы количество полученных из них комплектов было максимальным.

Существует несколько способов моделирования такой задачи (двух- и трехиндексная модели).

Двухиндексная модель предполагает, что мы сначала рассматриваем различные способы раскроя (при этом можно рассматривать раскрой деталей сложной формы). Для фиксированного способа можно подсчитать количество деталей j -го типа для каждой i -й заготовки при k -м способе раскроя: a_{ijk} . Например,

$$\begin{aligned} a_{111} &= 2, & a_{121} &= 0; \\ a_{112} &= 0, & a_{122} &= 1. \end{aligned}$$

Целесообразно отбросить способы раскроя, при которых величины a_{ijk} одинаковы, чтобы уменьшить размерность.

Интересно, что при помощи предварительного вычисления величин a_{ijk} можно решать не только «стержневые» задачи, но и «плоскостные»; для этого достаточно перечислить различные способы раскроя плоской заготовки i -го типа на множество плоских деталей из j -х типов. Далее

сама задача оптимального раскроя решается универсальным способом.

Рассмотрим величины x_{ik} — количество заготовок i -го типа, раскраиваемых k -м способом — это управляемые переменные.

Обозначим z_j — количество деталей j -го типа, полученных из всех типов заготовок при всех способах раскроя:

$$\sum_{i=1}^m \sum_{k=1}^K a_{ijk} \cdot x_{ik} = z_j,$$

где $[z_j/l_j]$ — число комплектов, которые можно изготовить из деталей j -го типа. Знак $[\cdot]$ обозначает ближайшее целое с недостатком.

Выберем тип деталей, которых меньше всего входит в комплект, в качестве основного, тогда можно максимизировать общее их количество

$$r = \arg \min_j l_j, \\ z_r = \sum_{i=1}^m \sum_{k=1}^K a_{irk} \cdot x_{ik} \rightarrow \max,$$

при выполнении условия комплектности:

$$\left. \begin{array}{l} \frac{z_1}{l_1} = \frac{z_2}{l_2} \\ \frac{z_2}{l_2} = \frac{z_3}{l_3} \\ \dots \\ \frac{z_m}{l_m} = \frac{z_n}{l_n} \end{array} \right\}.$$

Кроме этого, количество заготовок ограничено:

$$\sum_{k=1}^K x_{ik} \leq a_i, \quad i = \overline{1, m}.$$

Приведенная модель требует строгой пропорциональности деталей для производства комплектов. Но это может оказаться слишком строгим ограничением: задача или не будет иметь решения, или же решение будет не самым лучшим. Поэтому можно поставить задачу в более свободной форме.

Пусть z — число комплектов:

$$z \rightarrow \max, \quad (4.16)$$

$$\frac{\sum_{i=1}^m \sum_{k=1}^K a_{ijk} \cdot x_{ik}}{l_j} \geq z, \quad j = \overline{1, n}, \quad (4.17)$$

$$\sum_{k=1}^K x_{ik} \leq a_i, \quad i = \overline{1, m}. \quad (4.18)$$

В этой модели z — дополнительная переменная, причем, так как ограничения (4.17) более «слабые», ОДР более широкая и оптимальное количество комплектов может быть большим, чем в первой модели. Видно, что обе модели задачи раскроя материалов — это задачи ЛЦП, решение которых подробно рассмотрим ниже. Задача получения величин a_{ijk} решается отдельно для каждого типа заготовок и деталей. Для простых заготовок и деталей (стержень, прямоугольники) существуют простые алгоритмы, для изделий сложной формы — эвристические алгоритмы и специализированные программные системы.

4.2.4. ЗАДАЧА О РАНЦЕ

В ранец объема b необходимо упаковать предметы из имеющегося множества предметов разного типа так, чтобы суммарный вес ранца был максимальным. Пусть j — тип предмета, который занимает объем a_j и весит c_j .

Введем управляемые переменные x_j — количество предметов j -го типа, которые будут в ранце. Общий вес ранца

$$z = \sum_{j=1}^n c_j x_j \rightarrow \max. \quad (4.19)$$

Ограничение по объему ранца

$$\sum_{j=1}^n a_j x_j \leq b, \quad (4.20)$$

$$x_j — \text{целое}. \quad (4.21)$$

Это задача ЛЦП при единственном ограничении.

4.3. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ О РАНЦЕ

Рассмотрим специальный метод решения задачи о ранце, так как в ней лишь одно ограничение. При этом введем важнейшие понятия, необходимые для разработки общего метода решения задач ЛЦП.

Любое решение задачи о ранце — это вектор, например: $\bar{x} = (6, 0, 2, 4)$. Возьмем предметы с большим удельным весом в качестве приоритетных. Упорядочим компоненты по удельному весу:

$$\frac{c_j}{a_j} \mid \frac{c_1}{a_1} > \frac{c_2}{a_2} > \dots > \frac{c_n}{a_n}.$$

Старшая компонента — с большим удельным весом.

Определим лексикографический порядок таких векторов — это упорядочение векторов по первой компоненте, если они равны, то по второй и т. д. Большой вектор тот, у которого старшая компонента больше.

Лексикографический максимальный вектор для задачи о ранце определяется так:

$$\bar{x}^0 = (x_1^0, x_2^0, \dots, x_n^0),$$

$$x_1^0 = \left[\frac{b}{a_1} \right],$$

$$x_2^0 = \left[\frac{b - a_1 \cdot x_1^0}{a_2} \right],$$

...

$$x_k^0 = \left[\frac{b - \sum_{j=1}^{k-1} a_j \cdot x_j^0}{a_k} \right].$$

Следующий после максимального в лексикографическом ряду будет вектор, который получается так: последнюю ненулевую компоненту уменьшают на единицу, а следующую увеличивают на максимально возможную величину.

Несмотря на то, что лексикографически максимальный вектор дает близкий к оптимальному значению вес ранца, целевая функция для векторов, лежащих ниже, чем \bar{x}^0 , мо-

жет быть больше, чем целевая функция для \bar{x}^0 . В этом проявляется специфика целочисленности, так как если из ранца убрать один «тяжелый» предмет, то в сумме два более «легких» могут дать больший вес (они плотнее упакуются).

Если пройти все лексикографически упорядоченные векторы, то можно найти лучшее значение z .

Рассмотрим подход, позволяющий отсеять некоторое подмножество вариантов. Следует отметить, что все методы дискретной оптимизации направлены на сокращение перебора вариантов решения. Для этого мы должны быть уверены, что все эти отсеиваемые варианты ни в каком случае не дадут значение целевой функции лучше, чем уже полученное. Для этого используется понятие *оценки* множества вариантов.

Оценка подмножества вариантов — это такая величина, имеющая размерность целевой функции, что ни один вариант этого подмножества не даст значения целевой функции лучше, чем данная оценка (δ).

Грубую оценку сделать легко. Вопрос в том, чтобы получить оценку как можно ближе к границе допустимых значений целевой функции. Чем точнее оценка, тем больше будем отсеивать вариантов.

Рассмотрим в лексикографическом ряду вектор \bar{x}^i , стоящий на i -м месте. Его последняя ненулевая компонента x_s^i . Рассмотрим вектор \bar{y}^i , который получается из вектора \bar{x}^i так: последняя ненулевая компонента уменьшается на единицу $y_s^i = x_s^i - 1$, все дальнейшие — нули.

Сравним \bar{y}^i и \bar{x}^{i+1} .

Пример:

$$\bar{x}^i = (6, 0, 5, 3, 0, 0),$$

$$\bar{x}^{i+1} = (6, 0, 5, 2, 9, 8),$$

...

$$\bar{y}^i = (6, 0, 5, 2, 0, 0).$$

Дадим оценку всех вариантов, находящихся от \bar{x}^i до \bar{y}^i . Для этого определим оценку i -го множества в виде

$$\delta^i = \sum_{j=1}^n c_j y_j^i + \frac{c_{s+1}}{a_{s+1}} \left(b - \sum_{j=1}^n a_j y_j^i \right).$$

Первая сумма — это вес упаковки \bar{y}^i , а выражение в скобках — объем, оставшийся после упаковки \bar{y}^i , поэтому в оценке учитывается, что этот оставшийся объем заполняется плотно веществом с удельным весом, наибольшим для оставшихся предметов. Поэтому величина δ^i ни при каких целых предметах с номерами от $s+1$ до n превышена быть не может — это и есть свойство оценки.

Теперь, если для всех векторов от \bar{x}^0 до \bar{x}^i получено некоторое наибольшее, так называемое *рекордное* значение целевой функции z^* и $\delta^i \leq z^*$, это означает, что не надо рассматривать векторы от \bar{x}^i до \bar{y}^i , а сразу взять для анализа вектор \bar{y}^i . Если $\delta^i > z^*$, то ничего сказать нельзя и берем для анализа следующий вектор \bar{x}^{i+1} , хотя в сохраненном подмножестве может и не быть лучшего значения целевой функции.

Алгоритм решения задачи о ранце можно сформулировать так.

1. Проверим соотношения c_s/a_s и переобозначим переменные.

2. Найдем лексикографический максимальный вектор.

3. Строим вспомогательный вектор \bar{y}^i , вычисляем оценку, сравниваем ее с последним рекордом и проверяем возможность отсеивания векторов.

4. Если $\delta^{io} \leq z$ — переходим к \bar{y}^i , а если $\delta^{io} > z$, то переходим к следующему в лексикографическом ряду вектору \bar{x}^{i+1} .

Таким образом, доходим до нулевого вектора \bar{x}^i . Последний рекорд и определяет оптимальное решение.

Пример:

$$z = 2x'_1 + 7x'_2 + 4,5x'_3 + x'_4 \rightarrow \max,$$

$$6x'_1 + 3x'_2 + 2x'_3 + x'_4 \leq 19.$$

Проверим соотношения c_s/a_s и переобозначим переменные.

Ниже приведены все вычисления без комментариев.

$$\frac{7}{3} > \frac{4,5}{2} > \frac{1}{1} > \frac{2}{6};$$

$$\begin{cases} 7x_1 + 4,5x_2 + x_3 + 2x_4 \rightarrow \max, \\ 3x_1 + 2x_2 + x_3 + 6x_4 \leq 19; \end{cases}$$

$$\bar{x}^0 = (6, 0, 1, 0); \quad z = 6 \cdot 7 + 1 \cdot 1 = 43 = z^*; \quad s = 3;$$

$$\delta^0 = 42 + \frac{1}{3}(19 - 18) = 42\frac{1}{3} < z^*;$$

$$\bar{x}^1 = (6, 0, 0, 0) = \bar{y}^0; \quad \bar{y}^1 = (5, 0, 0, 0); \quad s = 1;$$

$$\delta^1 = 35 + \frac{4,5}{2}(19 - 15) = 35 + 9 = 44 > z^*;$$

$$\bar{x}^2 = (5, 2, 0, 0); \quad z = 5 \cdot 7 + 2 \cdot 4,5 = 44 = z^{**};$$

$$\bar{y}^2 = (5, 1, 0, 0); \quad s = 2; \quad \delta^2 = 39,5 + 1(19 - 17) = 41,5 < z^*;$$

$$\bar{x}^3 = \bar{y}^2 = (5, 1, 0, 0); \quad \bar{y}^3 = (5, 0, 0, 0); \quad s = 2;$$

$$\delta^3 = 35 + 1(19 - 15) = 39 < z^{**};$$

$$\bar{x}^4 = \bar{y}^3 = (5, 0, 0, 0); \quad \bar{y}^4 = (4, 0, 0, 0); \quad s = 1;$$

$$\delta^4 = 28 + \frac{4,5}{2}(19 - 12) = 28 + \frac{4,5 \cdot 7}{2} = 28 + 15\frac{3}{4} = 43\frac{3}{4} < z^{**};$$

$$\bar{x}^5 = \bar{y}^4 = (4, 0, 0, 0); \quad \bar{y}^5 = (3, 0, 0, 0); \quad s = 1;$$

$$\delta^5 = 21 + \frac{4,5}{2}(19 - 9) = 21 + \frac{90}{4} = 21 + 22\frac{1}{2} = 43\frac{1}{2} < z^{**};$$

$$\bar{x}^6 = \bar{y}^5 = (3, 0, 0, 0); \quad \bar{y}^6 = (2, 0, 0, 0); \quad s = 1;$$

$$\delta^6 = 14 + \frac{4,5}{2}(19 - 6) = 43\frac{1}{4} < z^{**};$$

$$\bar{x}^7 = \bar{y}^6 = (2, 0, 0, 0); \quad \bar{y}^7 = (1, 0, 0, 0); \quad s = 1; \quad B$$

$$\delta^7 = 7 + \frac{4,5}{2}(19 - 3) = 43 < z^{**};$$

$$\bar{x}^8 = \bar{y}^7 = (1, 0, 0, 0); \quad \bar{y}^8 = (0, 0, 0, 0); \quad s = 1;$$

$$\delta^8 = 0 + \frac{4,5}{2} \cdot 19 = 42\frac{3}{4} < z^{**};$$

$$\bar{x}^9 = (0, 0, 0, 0).$$

Ответ: (5, 2, 0, 0) с учетом переобозначения $\bar{x}^i = (0, 5, 2, 0)$.

Примечание. Если коэффициенты целевой функции целые, то истинное значение z может быть только целым, поэтому если оценка получилась дробной, то ее можно усилить (приблизить к истинной величине z : для $z \rightarrow \max$ уменьшить до ближайшего целого, а для $z \rightarrow \min$ увеличить до ближайшего целого).

4.4. РЕШЕНИЕ ЗАДАЧ ЛЦП МЕТОДОМ ОТСЕЧЕНИЙ ГОМОРИ

Алгоритмы решения задач ЛЦП используют в качестве подалгоритма симплекс-метод решения задач ЛП. Рассмотрим идею методов отсечения. Пусть дана задача ЛЦП (рис. 15). Отбросим требование целочисленности и решаем обычную задачу ЛП. Если решение получается целым, то оно является оптимальным. Если решение не удовлетворяет требованиям целочисленности, то применяют условие правильного отсечения. Для этого формируют новое ограничение, которое отсекает только что полученное оптимальное «непрерывное» решение, оставляя допустимыми все целочисленные решения.

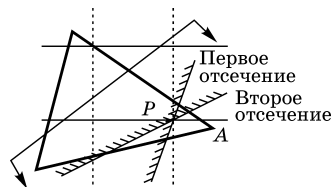


Рис. 15
Метод отсечений
решения задачи ЛЦП

Точка A — оптимальное «непрерывное» решение. Если его не отсечь, то оно появится на следующем этапе. Первое отсечение оставляет все целочисленные решения и отсекает точку A.

Точка B — оптимальное целочисленное решение.

Как построить отсечение, чтобы получить такое решение?

Решая задачу ЛП, мы получаем в симплекс-таблице выражение базисных переменных через свободные:

$$x_i = b_i - \sum_{j \in S} \alpha_{ij} x_j, \quad i = \overline{1, m}. \quad (4.22)$$

Пусть $x_j \in Z, j = \overline{1, n}, S$ — множество свободных переменных.

Коэффициенты α_{ij} — любые (целые, дробные).

Обозначим $[a]$ — ближайшее целое, не превосходящее a (называемое иногда целым с недостатком). Тогда $\{a\}$ — дробная часть числа a . Заметим, что для положительных и отрицательных нецелых чисел a выполняется:

$$\{a\} = a - [a] > 0;$$

$$\begin{aligned} x_i &= [b_i] + \{b_i\} - \sum_{j \in S} [\alpha_{ij}] x_j - \sum_{j \in S} \{\alpha_{ij}\} x_j = \\ &= [b_i] - \sum_{j \in S} [\alpha_{ij}] x_j + \{b_i\} - \sum_{j \in S} \{\alpha_{ij}\} x_j. \end{aligned} \quad (4.23)$$

Покажем, что если в решении x_i целое, то выражение

$$\{b_i\} - \sum_{j \in S} \{\alpha_{ij}\} x_j \text{ целое и обязательно } \leq 0. \quad (4.24)$$

Формула (4.22) представляет собой запись симплекс-таблицы для оптимального решения, но решение не целое. Рассмотрим строку с нецелым b_i :

$$\{b_i\} \neq 0, \quad 0 \leq \{b_i\} \leq 1.$$

Так как $\{\alpha_{ij}\} > 0, x_j > 0$, то отсюда следует, что величина, обозначенная в формуле (4.24), может быть только отрицательным целым числом.

Это необходимый признак целочисленности x_i . Условие (4.24) можно считать правильным отсечением. Неравенство (4.24) приведем к равенству, введя дополнительную переменную y_{n+1} :

$$y_{n+1} = -\{b_i\} + \sum_{j \in S} \{\alpha_{ij}\} x_j. \quad (4.25)$$

Если это ограничение добавить к предыдущим ограничениям, то оно отсекает полученное непрерывное решение. Так как в симплекс-таблице появляется отрицательный свободный член, то надо искать новое опорное и новое оптимальное решение.

Алгоритм Гомори:

1. Решаем задачу ЛП. Находим оптимальное решение. Если в оптимальной симплекс-таблице все свободные члены не меньше нуля и целые, то получено оптимальное целочисленное решение. Иначе переходим ко второму пункту.

2. Находим строку с наибольшей дробной частью $\{b_i\}$. Для этой строки по формуле (4.25) составляем правильное отсечение и записываем дополнительной строкой в симплекс-таблицу; переходим к п. 1.

При целых коэффициентах задача ЛЦП сходится за конечное число шагов к точному решению.

Если в исходной задаче коэффициенты дробные и есть ограничения неравенства, то дополнительные переменные y_i могут быть дробными и задача не является полностью целочисленной. В этом случае может получиться, что если потребовать $y_i \in Z$ и построить отсечение (4.25), то целых решений может не быть. Для таких задач можно предварительно избавиться от дробных коэффициентов, умножив неравенства на наименьшее общее кратное знаменателей дробей, тогда все y_i будут целыми. Если же в исходной задаче

$$x_j \in Z \text{ для } j = 1, n_1; \quad x_j \in R \text{ для } j = n_1 + 1, n, \quad (4.26)$$

то вместо (4.25) применяют другое условие правильного отсечения (см. подробно [7]).

Если требуется, чтобы значение целевой функции было целым, то строим условие правильного отсечения по строке L .

Пример. Решить задачу:

$$\begin{aligned} L &= -x_1 + 3x_2 \rightarrow \min, \\ 2x_1 + x_2 &\leq 4, \\ 4x_1 - 3x_2 &\leq 2, \\ -3x_1 + 2x_2 &\leq 3, \\ x_1, x_2 &\geq 0 \text{ и целые числа.} \end{aligned}$$

Сначала решим данную задачу без требования целочисленности симплекс-методом. Для этого запишем ее в виде основной задачи линейного программирования (ОЗЛП) в стандартной форме:

$$\begin{aligned} L &= 0 - (x_1 - 3x_2) \rightarrow \min, \\ y_1 &= 4 - (2x_1 + x_2), \\ y_2 &= 2 - (4x_1 - 3x_2), \\ y_3 &= 3 - (-3x_1 + 2x_2). \end{aligned}$$

Далее заполним симплекс-таблицу (табл. 28).

Это опорное решение, так как в столбце свободных членов при базисных переменных все коэффициенты положительны. Но решение не оптимально, так как в строке функции цели есть положительный коэффициент при свободной переменной. Ищем оптимальное решение.

Таблица 28

	1	x_1	x_2
L	0	1	-3
y_1	4	2	1
y_2	2	4	-3
y_3	3	-3	2

→

Таблица 29

	1	y_2	x_2
L	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{9}{4}$
y_1	3	$-\frac{1}{2}$	$\frac{5}{2}$
x_1	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{3}{4}$
y_3	$\frac{9}{2}$	$\frac{3}{4}$	$-\frac{1}{4}$

Делаем все необходимые преобразования в симплекс-таблице для замены y_2 на x_1 , и переходим к табл. 29.

Это оптимальное решение задачи без требования целочисленности, так как в столбце свободных членов нет отрицательных коэффициентов при базисных переменных, а в строке функции цели L нет положительных коэффициентов при свободных переменных. Таким образом,

$$x_{1\text{опт}} = \frac{1}{2}; \quad x_{2\text{опт}} = 0; \quad L = \frac{1}{2}.$$

Но это решение не удовлетворяет требованию целочисленности. Для построения правильного отсечения и ввода нового ограничения берется первая по номеру нецелочисленной переменная. В данном случае это x_1 .

Выпишем x_1 из табл. 29:

$$\begin{aligned} x_1 &= \frac{1}{2} - \left(\frac{1}{4}y_2 - \frac{3}{4}x_2 \right), \\ \left\{ \frac{1}{2} \right\} &= \frac{1}{2} - \left[\frac{1}{2} \right] = \frac{1}{2} - 0 = \frac{1}{2}, \\ \left\{ \frac{1}{4} \right\} &= \frac{1}{4} - \left[\frac{1}{4} \right] = \frac{1}{4} - 0 = \frac{1}{4}, \\ \left\{ -\frac{3}{4} \right\} &= -\frac{3}{4} - \left[-\frac{3}{4} \right] = -\frac{3}{4} - (-1) = \frac{1}{4}. \end{aligned}$$

Применяя формулу (4.25), получим

$$y_4 = -\frac{1}{2} + \left[\frac{1}{4}y_2 + \frac{1}{4}x_2 \right]$$

— это новое ограничение, которое определяет правильное отсечение. Перепишем это ограничение в стандартном виде

$$y_4 = -\frac{1}{2} - \left(-\frac{1}{4}y_2 - \frac{1}{4}x_2 \right)$$

и допишем его в симплекс-таблицу 30.

Для решения этой расширенной задачи без требования целочисленности необходимо найти опорное решение, так как в столбце свободных членов есть отрицательные элементы при базисных переменных. Делаем необходимые преобразования в табл. 30 для нахождения опорного решения и переходим к табл. 31.

Таблица 30

	1	y_2	x_2
L	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{9}{4}$
y_1	3	$-\frac{1}{2}$	$\frac{5}{2}$
x_1	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{3}{4}$
y_3	$\frac{9}{2}$	$\frac{3}{4}$	$-\frac{1}{4}$
y_4	$-\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{1}{4}$

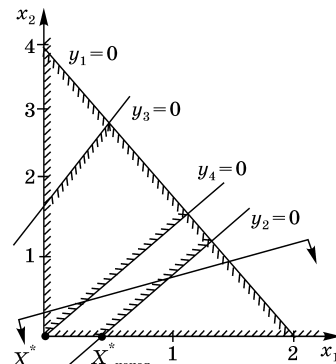


Рис. 16

Геометрическая интерпретация примера решения задачи ЛЦП методом отсечений

Таблица 31

	1	y_4	x_2
L	0	-1	-2
y_1	4	-2	3
x_1	0	1	-1
y_3	3	3	-1
y_2	2	-4	1

Решение оптимально и целочисленно: $x_{1\text{опт}} = 0$; $x_{2\text{опт}} = 0$; $L = 0$.

Геометрическая интерпретация данного примера приведена на рис. 16.

4.5.

МЕТОД ВЕТВЕЙ И ГРАНИЦ (МВГ)

Метод ветвей и границ (МВГ) — это наиболее широко применяемый метод дискретного программирования. Он относится к комбинаторным методам; МВГ — метод последовательного анализа вариантов.

Широкое распространение МВГ получил, когда с его помощью был разработан алгоритм решения задачи коммивояжера (алгоритм Литтла), а также алгоритм решения задачи ЛЦП (алгоритм Лэнда-Дойга).

Название МВГ раскрывает основное содержание метода, он состоит из двух процедур:

- 1) ветвление вариантов решения;
- 2) установление границ возможного для каждого анализируемого подмножества решений.

В любой задаче все множество решений представляется в виде дерева вариантов, т. е. вводится принцип разбиения множества вариантов на подмножества вариантов.

В каждой задаче может быть свой принцип разбиения. Для МВГ необходимо, чтобы, во-первых, был удобный способ перехода от множества к подмножеству, во-вторых, чтобы варианты не повторялись и чтобы ни один вариант не был упущен.

Переход от одного варианта к другому осуществляется в соответствии с некоторым порядком (например, в лексикографическом порядке).

Исходя из свойств задачи и физического смысла целевой функции, устанавливается граница возможного (иногда называемого «обещанием») для каждого подмножества, т. е. придумывается формула оценки для заданного подмножества.

Оценкой множества вариантов называется такое значение ЦФ, которое не может быть улучшено ни на одном из вариантов оцениваемого множества. Оценка множества

всегда не хуже оценки входящего в него подмножества. В частности, если в некоторой задаче добавляется ограничение, то оценка для этой задачи может только ухудшиться или остаться неизменной.

Оценка должна быть как можно более точной, т. е. быть как можно ближе к истинным значениям. Чем точнее оценка, тем чаще отсеиваются варианты. Действительно, если получено некоторое рекордное значение ЦФ z^* , а наилучшая из оценок, оставшихся для проверки подмножеств, хуже z^* , то, очевидно, процесс дальнейшего улучшения z можно прекратить. Чтобы это произошло как можно раньше, во-первых, надо получить достаточно «хороший» первый рекорд, а во-вторых, оценки должны быть как можно более точными.

Рассмотрим МВГ применительно к любой задаче дискретного программирования:

$$z = f(\bar{x}) \rightarrow \min, \quad (4.27)$$

$$\bar{x} \in G, \quad (4.28)$$

где G — некоторая область, определяемая системой ограничений неравенств, а также типом переменных.

На первом этапе вычисляют нижнюю границу целевой функции. Для этого снимают ограничение на тип переменных. Целевая функция может быть при этом только лучше (или равной), поэтому ее можно принять за оценку.

Если $\xi(G)$ — оценка множества G , то для задачи на минимум $f(\bar{x}) \geq \xi(G)$.

На втором этапе производится разбиение (ветвление) множества G . Само разбиение идет в процессе решения и зависит от результатов, которые получены при разбиении на непересекающиеся подмножества.

Рассмотрим k -ю итерацию. Для дальнейшего разбиения выбирают так называемое перспективное подмножество (например то, которое имеет наилучшую оценку). Заметим, что при разбиении на следующем уровне получаемые оценки могут быть только хуже (или те же).

Если $G_1 \subset G_0$, то $\xi(G_1) \geq \xi(G_0)$.

Как только на каком-то подмножестве вариантов получено решение, удовлетворяющее всем ограничениям

(в том числе специальному требованию на тип переменной), проверяется условие оптимальности МВГ.

Условие оптимальности МВГ. Если на каком-либо подмножестве $G_v^{(k)}$ на k -й итерации получено такое решение \bar{x}^* , что значение целевой функции равно оценке этого подмножества $f(\bar{x}^*) = \xi(G_v^{(k)})$, $\bar{x}^* \in G_v^{(k)}$ и при условии сама оценка является наилучшей среди всех оценок еще не разбитых подмножеств на этой итерации $\xi(G_v^{(k)}) = \min_i \{\xi(G_i^{(k)})\}$, при этом $\bigcup_i G_i^{(k)} = G_0$, то решение \bar{x}^* является оптимальным и улучшено быть не может.

Действительно, в самом множестве G_v лучшего решения быть не может, а остальные оценки дают «обещание» решений хуже, чем \bar{x}^* .

В практических задачах можно ограничиться приближенным решением с точностью до ε .

Пусть получено некоторое решение \bar{x}' и задана некоторая абсолютная точность решения ε :

$$|f(\bar{x}') - f(\bar{x}^*)| < \varepsilon,$$

где \bar{x}^* — неизвестное нам точное решение (рис. 17).

Если $\Delta = |f(\bar{x}') - \min_i \{\xi(G_i^{(k)})\}| < \varepsilon$, то разница между точным оптимальным значением и приближенным тем более меньше ε .

Если $\Delta < \varepsilon$, то $\delta < \varepsilon$.

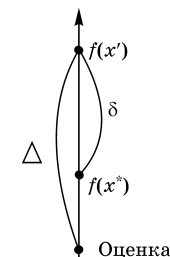


Рис. 17
Точное и приближенное оптимальные решения и оценка решений

4. 6. МВГ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА

Рассмотрим применения МВГ для решения задачи коммивояжера, сформулированной в п. 4.2.2. Исторически первым появился так называемый алгоритм Литтла. Данный алгоритм используется для поиска оптимального гамильтонова контура в графе, имеющем n вершин, причем каждая вершина i связана с любой другой вершиной j двуправленной дугой. Каждой дуге приписан вес c_{ij} ,

причем веса дуг строго положительны ($c_{ij} \geq 0$). Все элементы по диагонали матрицы приравнивают к бесконечности ($c_{ii} = \infty$).

В случае, если пара вершин i и j не связана между собой (граф не полносвязный), то соответствующему элементу матрицы приписываем вес, равный длине минимального пути между вершинами i и j . Если в итоге дуга (i, j) войдет в результирующий контур, то ее необходимо заменить соответствующим ей путем.

Алгоритм Литтла является частным случаем применения рассмотренного МВГ для конкретной задачи. В соответствии с МВГ нужно разделить огромное число перебираемых вариантов на подмножества в виде дерева подмножеств (ветвей) и получить оценки (снизу — в задаче минимизации, сверху — в задаче максимизации) для этих ветвей, чтобы иметь возможность отбрасывать варианты не по одному, а целыми ветвями. Трудность состоит в том, чтобы найти такое разделение на подмножества (ветви) и такие оценки ветвей (границы возможных значений целевой функции), чтобы процедура была эффективной, т. е. чтобы как можно большее число вариантов отсеивалось. Теоретическое обоснование алгоритма Литтла выходит за рамки настоящего учебного пособия, его можно найти в [30], [39]. Дадим краткое описание алгоритма Литтла и пример, демонстрирующий особенности МВГ для этой задачи.

Алгоритм Литтла.

1. В каждой строке матрицы c_{ij} найдем минимальный элемент и вычтем его из всех элементов строки. Сделаем это и для столбцов, не содержащих нуля. Получим матрицу, каждая строка и каждый столбец которой содержат хотя бы один нулевой элемент.

2. Для каждого нулевого элемента матрицы c_{ij} рассчитаем коэффициент $\theta(i, j)$, который равен сумме наименьшего элемента i -ой строки (исключая элемент $c_{ij} = 0$) и наименьшего элемента j -го столбца.

3. Из всех коэффициентов $\theta(i, j)$ выберем такой, который является максимальным $\theta(k, l) = \max\{\theta(i, j)\}$. В гамильтонов контур вносится соответствующая дуга (k, l) .

4. Удаляем k -ую строку и l -й столбец и меняем на бесконечность значение элемента ckl (поскольку дуга (k, l) включена в контур, то обратный путь из l в k недопустим).

5. Повторяем алгоритм, пока порядок матрицы не станет равным двум.

6. В текущий ориентированный граф вносим две недостающие дуги, определяющиеся однозначно матрицей порядка 2. Получаем гамильтонов контур.

В ходе решения ведется постоянный подсчет текущего значения нижней границы-оценки соответствующего подмножества циклов. Нижняя граница равна сумме всех вычтенных элементов в строках и столбцах. Итоговое значение нижней границы должно совпасть с длиной результирующего контура.

Пример. Пусть матрица «расстояний» в задаче коммивояжера имеет вид

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

Приведем матрицу, вычитая сначала из строк, затем из столбцов минимальные элементы. Из строк вынесены (3, 3, 2, 1), далее из столбцов (0, 1, 0, 0).

Получим матрицу (сверху и справа матрицы указаны номера строк и столбцов)

$$C^0 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} \infty & 2 & 2 & 0^* \\ 0 & \infty & 1 & 1 \\ 0 & 0 & \infty & 1 \\ 1 & 1 & 0 & \infty \end{pmatrix} \end{matrix} \begin{matrix} 1 \\ 2 \\ 3^* \\ 4 \end{matrix}$$

Оценка исходного множества циклов будет равна сумме вынесенных элементов-констант приведения:

$$\xi(G^{(0)}) = 3 + 3 + 2 + 1 + 0 + 1 + 0 + 0 = 10.$$

Выбираем 0 в C^0 так, чтобы сумма θ минимальных, кроме него, коэффициентов его строки и столбца была максимальна. Это будет для $c_{1,4} = 0$, а именно $\theta(1, 4) = 3$.

Пусть $G_1^{(1)}$ — множество циклов, включающих переход $1 \rightarrow 4$.

Такие циклы определяет матрица C_1^1 :

$$C_1^1 = \begin{pmatrix} 1 & 2 & 3 \\ 0 & \infty & 1 \\ 0 & 0 & \infty \\ \infty & 1 & 0 \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}.$$

Так как C_1^1 уже приведена, то

$$\xi(G^0) = \xi(G_1^1) = 10.$$

Второе множество G_2^1 — множество циклов, не имеющих перехода $1 \rightarrow 4$.

Ему соответствует матрица

$$C_2^1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ \infty & 2 & 2 & \infty \\ 0 & \infty & 1 & 1 \\ 0 & 0 & \infty & 1 \\ 1 & 1 & 0 & \infty \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix};$$

$$\xi(G_2^{(1)}) = \xi(G^0) + \theta(1, 4) = 10 + 3 = 13.$$

Рассмотрим подмножество циклов с лучшей оценкой — перспективное подмножество — это $G_1^{(1)}$. В матрице C_1^1 выбираем 0 с максимальной θ . Получаем $\theta(4, 3) = 2$.

Пусть $G_1^{(2)}$ — множество циклов, включающих переходы $1 \rightarrow 4 \rightarrow 3$. Ему соответствует приведенная матрица

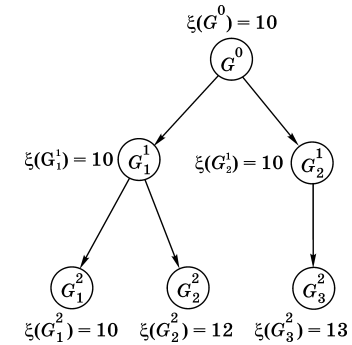
$$C_1^2 = \begin{pmatrix} 1 & 2 \\ 0 & \infty \\ \infty & 0 \end{pmatrix} \begin{matrix} 1 \\ 2 \end{matrix};$$

$$\xi(G_1^{(2)}) = \xi(G_1^{(1)}) = 10.$$

Рассмотрим $G_2^{(2)}$ — множество циклов, содержащих переход $1 \rightarrow 4$, но не содержащих $4 \rightarrow 3$. Ему соответствует матрица

$$C_2^2 = \begin{pmatrix} 1 & 2 & 3 \\ 0 & \infty & 1 \\ 0 & 0 & \infty \\ \infty & 1 & \infty \end{pmatrix} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix};$$

Рис. 18
Схема ветвления
решения задачи
коммивояжера



$$\xi(G_2^{(2)}) = \xi(G_1^{(1)}) + \theta(4, 3) = 12;$$

$$G_3^{(2)} = G_2^{(1)}.$$

Множество $G_1^{(2)}$ содержит единственный цикл $(1 \rightarrow 4 \rightarrow 3 \rightarrow 2)$, длина которого равна наилучшей оценке, т. е. выполняется условие оптимальности МВГ:

$$\xi(G_1^{(2)}) = \min\{\xi(G_1^{(2)}), \xi(G_2^{(2)}), \xi(G_3^{(2)})\} = \min\{10, 12, 12\} = 10.$$

Следовательно, замкнутый цикл $(1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1)$ является кратчайшим.

$$c_{14} + c_{43} + c_{32} + c_{21} = 3 + 1 + 3 + 3 = 10.$$

Схема ветвления решенной задачи показана на рис. 18.

4.7. АЛГОРИТМ МВГ ДЛЯ ЗАДАЧИ ЛЦП

Рассмотрим задачу ЛЦП:

$$z = \sum_{j=1}^n c_j x_j \rightarrow \min, \quad (4.29)$$

$$\sum_{j=1}^n a_j x_j \leq b_i, \quad i = \overline{1, m}, \quad (4.30)$$

$$x_j — \text{целое}, \quad j = \overline{1, n_1}, \quad n_1 \leq n. \quad (4.31)$$

Для задачи ЛЦП (4.29), (4.30), (4.31) оценкой множества решений, определяемых формулами (4.30), (4.31), является оптимальное значение для задачи ЛП (4.29),

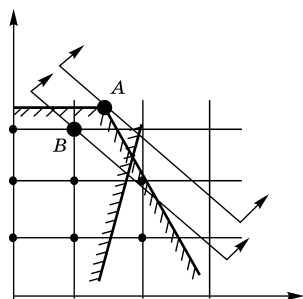


Рис. 19
Геометрическая
интерпретация
понятия оценки
для задачи ЛЦП

(4.30). На рис. 19 точка A — решение задачи ЛП, а точка B — решение задачи ЛЦП. Кроме этого, для ЛЦП можно использовать простое правило ветвления вариантов.

Получаем следующий алгоритм (*алгоритм Ленда — Дойга*):

1. Отбросим условие (4.31) и найдем решение задачи ЛП. Если решение автоматически удовлетворяет условию (4.31), то оно является оптимальным. Иначе переходим ко второму пункту.

2. Рассмотрим какую-нибудь нецелочисленную переменную полученного решения $x_l = d_l$, $l \in \{1, n_1\}$. По этой переменной производится ветвление, все множество решений разбивается на два и кроме ограничений (4.30) добавляются:

$$G_1^{(1)}: x_l \geq [d_l] + 1,$$

$$G_2^{(1)}: x_l \leq [d_l].$$

Пример.

$$x_5 = 4\frac{4}{7} \rightarrow \begin{cases} x_5 \leq 4, \\ x_5 \geq 5. \end{cases}$$

Снова решаем задачу на каждом подмножестве (ветви). Оценкой ветви будет значение целевой функции, полученное для оптимального решения соответствующей задачи ЛП. Если оба решения не удовлетворяют формуле (4.31), то в качестве перспективного выбирают то подмножество, в котором оценка лучше. Оценки сравниваются для всех еще не разбитых на данной итерации подмножеств (концевых вершин дерева ветвления). Как только

на каком-нибудь подмножестве получено целочисленное решение, проверяем *условие оптимальности* МВГ. Если оно выполняется, то получено оптимальное решение. Если не выполняется, то снова выбирают перспективное подмножество и продолжают поиск решения.

Примечания. 1. Если решения задачи ЛП нет, то соответствующее ее множество $G_l = \emptyset$ пустое. Ему необходимо приписать оценку $+\infty$ (для $z \rightarrow \min$).

2. Если все коэффициенты целевой функции целые, то оценку множества можно заменить на более сильную. Для задачи на \min — на величину $\lceil \xi(G_l) \rceil$ — ближайшее целое с избытком. Для $z \rightarrow \max$ — на $\lfloor \xi(G_l) \rfloor$.

Пример.

$$z = -x_1 - x_2 \rightarrow \min, \quad (4.32)$$

$$\left. \begin{aligned} 2x_1 + 11x_2 &\leq 38; \\ x_1 + x_2 &\leq 7; \\ 4x_1 - 5x_2 &\leq 5; \end{aligned} \right\} \quad (4.33)$$

$$x_1, x_2 — \text{целые}.$$

Решим задачу (4.32), (4.33). Получим

$$\bar{x}_0 = \left(\frac{40}{9}, \frac{23}{9} \right); \quad \xi(G_0) = z(\bar{x}_0) = -7,$$

$$G_1^{(1)} = \{x_1 \leq 4\},$$

$$G_2^{(1)} = \{x_1 \geq 5\}.$$

Решив эти задачи, получим:

$$x_1 = 4; \quad x_2 = 2\frac{8}{11}; \quad \xi(G_1^{(1)}) = \left\lceil -6\frac{8}{11} \right\rceil = -6; \quad G_2^{(1)} = \emptyset; \quad \xi(G_2) = +\infty.$$

Таким образом, $G_1^{(1)}$ — перспективное множество. Разбив его на два подмножества по x_2 , получаем:

$$G_{1,1}^{(1)}: (x_2 \leq 2) \dots$$

$$G_{1,2}^{(2)}: (x_2 \geq 3) \dots$$

Переобозначим:

$$G_{1,1}^{(1)} = G_1^{(2)}; \quad G_{1,2}^{(1)} = G_2^{(2)}; \quad G_2^{(1)} = G_{(3)}^{(2)}.$$

Решим задачу на $G_1^{(2)}$. Получим

$$\bar{x}_1^2 = \left(3\frac{3}{4}; 2\right); \quad \xi(G_1^{(2)}) = \left\lfloor -5\frac{3}{4} \right\rfloor = -5.$$

На $G_2^{(2)}$ получим:

$$G_2^{(2)}: \bar{x}_2^2 = \left(2\frac{1}{2}; 3\right), \quad \xi(G_2^{(2)}) = \left\lfloor -5\frac{1}{2} \right\rfloor = -5,$$

$$G_3^{(2)} = \emptyset; \quad \xi = +\infty.$$

Сравниваем оценки $(-5, -5, \infty)$, выбираем перспективное подмножество $G_1^{(2)}$ и разбиваем его на два:

$$G_{1,1}^{(2)}: (x_1 \leq 3),$$

$$G_{1,2}^{(2)}: (x_1 \geq 4).$$

Переобозначим (получим 4 подмножества).

Решив задачу на

$$G_1^{(3)} \text{ получаем } \bar{x}_1^{(3)} = (3; 2); \quad \xi(G_1^{(3)}) = -5,$$

$$G_2^{(3)} = \emptyset; \quad \xi(G_2^{(3)}) = +\infty;$$

$$\text{на } G_3^{(3)} \text{ получаем } \bar{x}_3^{(3)} = \left(2\frac{1}{2}; 3\right); \quad \xi(G_3^{(3)}) = \left\lfloor -5\frac{1}{2} \right\rfloor = -5,$$

$$G_4^{(3)} = \emptyset; \quad \xi(G_4^{(3)}) = +\infty.$$

Получено целочисленное решение $\bar{x}_1^{(3)} = (3; 2)$. Применим условие оптимальности МВГ:

$$z(3, 2) = -5 = \min\{-5, +\infty, -5, +\infty\} = -5.$$

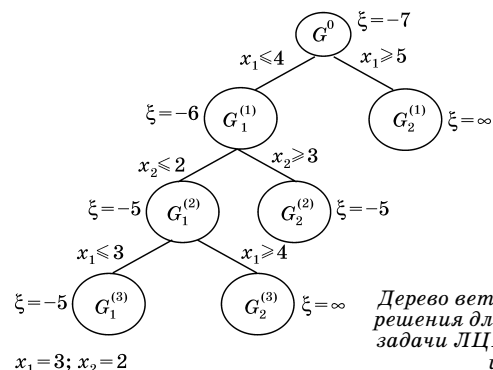


Рис. 20
Дерево ветвления вариантов
решения для примера решения
задачи ЛЦП методом ветвей
и границ

Условие оптимальности выполняется.

Решение $\bar{x}_1^{(3)} = (3; 2)$ является оптимальным $z^* = -5$. Изобразим дерево решений (рис. 20).

4.8. АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ БУЛЕВА ПРОГРАММИРОВАНИЯ

Задача булева программирования (БП) наиболее часто встречается при проектировании технических средств и автоматизированных систем управления производством.

Как было показано, любую задачу дискретного программирования, в свою очередь, можно привести к задаче БП.

Для решения задач БП используется большое число методов прямого поиска. Все они похожи на метод ветвей и границ, так как в них происходит перебор вариантов и анализ подмножеств в целях отсеивания целых подмножеств вариантов.

Большинство алгоритмов используют частичный вид булевой формы. Из-за важности приложений существуют специализированные процессоры для решения таких задач.

Рассмотрим задачу линейного булева программирования (ЛБП):

$$z = \sum_{j=1}^n c_j x_j \rightarrow \max, \quad (4.34)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}, \quad (4.35)$$

$$x_j \in \{0, 1\}, \quad i = \overline{1, n}, \quad (4.36)$$

Существует 2^n вариантов решения, часть из которых допустима.

При большом n придется проверять все m ограничений, а проверка каждого ограничения — это трудоемкая операция.

Рассмотрим процедуру перебора, положенную в основу аддитивного алгоритма Балаша [8].

Так как внутри метода всегда будет сравнение с рекордом, то необходимо, чтобы первое рекордное значение было

достаточно большим. Поэтому переменные исходной задачи преобразовывают в соответствии с возрастанием коэффициентов целевой функции (для задачи на максимум):

$$c_1 \leq c_2 \leq c_3 \leq \dots \leq c_n.$$

В этом случае мы будем перебирать векторы, начиная с нулевого, и добавляя единицы справа. Чтобы недопустимость вектора проявлялась сразу, можно выстроить ограничения в таком порядке: сначала самое жесткое из них, а далее менее жесткие.

Затем выбирают первый вектор списка и проверяют ограничения, если какое-то из них не выполняется, то переходим к следующему вектору и т. д. Если для какого-то вектора удовлетворяются все ограничения, то подсчитывается величина целевой функции и она объявляется рекордом. После этого процедура меняется.

Э. Балаш предложил добавить некоторое фильтрующее ограничение. Идея его в том, что незачем проверять вектор на допустимость, если он не превосходит рекорда z^* :

$$\sum_{j=1} c_{ij} x_j > z^*. \quad (4.37)$$

Если ограничение (4.37) не выполняется, то переходим к следующему вектору, в противном случае — к ограничению (4.35). Если новый вектор удовлетворяет всем ограничениям, то, подставляя его в (4.34), получим новый рекорд z^* и формируем новое фильтрующее ограничение, заменяя z^* на z^{**} . Если пройден весь список, то последний рекорд дает решение. При этом мы экономим на проверке ограничений.

Пример.

$$\begin{aligned} z &= 3x_1 - x_2 + 2x_3 + x_4 \rightarrow \max, \\ 2x_1 - x_2 - 4x_3 + x_4 &\leq 10, \\ -3x_1 + 2x_2 + x_3 + 2x_4 &\geq 1, \\ 4x_1 - x_2 - 2x_3 &\leq 4, \\ x_j &\in \{0, 1\}. \end{aligned}$$

Преобразуем переменные и пронумеруем целевую функцию и ограничения:

Таблица 32

\bar{x}	1	2	3	(0)	z
0 0 0 0	+	—			2
0 0 0 1	+	—			
0 0 1 0	+	+	+		
0 0 1 1	+	—		+	
0 1 0 0				—	3
0 1 0 1	+	—		+	
0 1 1 0	+	+	+	+	
0 1 1 1	+	—		+	
1 0 0 0				—	
1 0 0 1				—	
1 0 1 0				—	
1 0 1 1	+	—		+	
1 1 0 0				—	5
1 1 0 1				—	
1 1 1 0				—	
1 1 1 1	+	+	+	+	

$$z = -x_2 + x_4 + 2x_3 + 3x_1 \rightarrow \max,$$

$$-x_2 + x_4 - 4x_3 + 2x_1 \leq 10, \quad (1)$$

$$2x_2 + 2x_4 + x_3 - 3x_1 \geq 1, \quad (2)$$

$$-x_2 + 0 \cdot x_4 - 2x_3 + 4x_1 \leq 4. \quad (3)$$

Составим табл. 32 и получим первый допустимый вектор (0010); $z = 2$. Составляем фильтрующее ограничение:

$$-x_2 + x_4 + 2x_3 + 3x_1 > 2. \quad (0)$$

Находим новый рекордный вектор (0110); $z = 3$, новое фильтрующее ограничение:

$$-x_2 + x_4 + 2x_3 + 3x_1 > 3.$$

Наконец, находим новое рекордное значение (1111); $z = 5$. Так как список исчерпан, то последний рекорд — это оптимальное решение: $z_{\max} = 5$.

Эффективность процесса можно оценить количеством значков в таблице. В современных алгоритмах (например, в алгоритме Лемке — Шпильберга) порядок перехода от одного вектора к другому меняется в соответствии с анализом коэффициентов модели задачи:

1. Критерий недопустимости запрещает двигаться к векторам, которые заведомо недопустимы.

2. Проверяются целевые функции — множеству дается оценка (подробнее см. [7], [10], [23]).

В алгоритме Балаша используются и другие ограничения, позволяющие отсечь некоторые решения (см. [7]). Обратимся к некоторым из них.

Рассмотрим правило, по которому можно отсеивать варианты. Пусть N — множество индексов всех двоичных векторов. Подмножество множества N , состоящее из таких элементов j , что каждому j поставлено в соответствие $x_j \in \{0, 1\}$, называется частичным решением, а множество индексов обозначим V .

Переменные x_j , номера которых не принадлежат этому подмножеству, называются свободными, множество индексов свободных переменных обозначим S .

Пример. Вектор переменных $(x_1, x_2, x_3, x_4, x_5)$

$N = \{1, 2, 3, 4, 5\}$,

$(x_1, 0, 1, x_4, x_5)$ — частичное решение, $V = \{2, 3\}$,

$S = N/V$,

$S = \{1, 4, 5\}$ — номера свободных переменных.

Если частичное решение содержит k переменных, то существует 2^{n-k} различных дополнений частичного решения.

Пусть каким-то образом выбрано частичное решение, содержащее k переменных, тогда проверка 2^{n-k} его дополнений на допустимость по условию (4.35) называется *зондированием*.

Если до этого получен рекорд z^* и удастся установить, что для какого-то частичного решения не существует допустимого дополнения, приводящего к лучшему рекорду, то говорят, что частичное решение прозондировано, т. е. что это подмножество можно отсеять.

Балаш установил, что если V определяет переменные зондированного решения, а z^* — ранее достигнутый ре-

корд, то зондируемое решение x не имеет допустимого дополнения, улучшающего z^* , если выполняется условие

$$\sum_{j \in S} \min(a_{ij}, 0) > b_i - \sum_{j \in V} a_{ij} x_j, \quad i = \overline{1, m}. \quad (4.38)$$

Например, пусть задача содержит ограничение

$$-x_1 + 2x_2 + 6x_3 + 4x_4 - x_5 \leq 0,$$

$$x_1 = 1, \quad x_3 = 1, \quad x_4 = 0,$$

$$V = \{1, 3, 4\}, \quad N = \{1, 2, 3, 4, 5\}, \quad S = \{2, 5\}.$$

Проверим наличие допустимых дополнений для выбранного частичного решения:

$$\sum_{j \in S} \min(a_{1j}, 0) = -1; \quad \sum_{j \in V} a_{1j} x_j = 5; \quad b_1 - \sum_{j \in V} a_{1j} x_j = -5; \quad -1 > -5.$$

Значит, не существует допустимых дополнений частичного решения $x_1 = 1; x_3 = 1; x_4 = 0$.

Решение считается прозондированным.

Рассмотрим другое правило отсеивания вариантов. Если для некоторой свободной переменной x_k имеет место соотношение

$$\sum_{j \in S} \min(a_{ij}, 0) + |a_{ik}| > b_i - \sum_{j \in V} a_{ij} x_j, \quad i = \overline{1, m},$$

то

$$x_k = \begin{cases} 0, & \text{если } a_{ik} \geq 0, \\ 1, & \text{если } a_{ik} < 0. \end{cases} \quad (4.39)$$

Кроме этих двух правил можно использовать так называемый признак составных ограничений, позволяющий установить факт наличия или отсутствия допустимых дополнений некоторого частичного решения, путем анализа составного ограничения.

В общем случае это комбинация всех строк ограничений.

Например, используется суммирование левых и правых частей ограничений.

Пример. Пусть задача состоит из такого решения x_1, x_2, \dots, x_5 , которое имеет ограничения:

$$-x_1 - x_2 - x_3 \leq -1,$$

$$2x_2 - 2x_4 + 2x_5 \leq 0,$$

$$x_j = \{0, 1\}.$$

Максимизируется функция

$$z = 2x_1 + x_2 + x_3 \rightarrow \max.$$

Пусть при некотором частичном решении рекорд составлял $z = 2$. Определить, имеет ли частичное решение $x_1 = 0$ допустимое дополнение, улучшающее достигнутый рекорд.

Прибавим к ограничениям фильтрующее ограничение, т. е.

$$2x_1 + x_2 + x_3 \geq 3,$$

получим новую систему ограничений:

$$\begin{aligned} -x_1 - x_2 - x_3 &\leq -1, \\ 2x_2 - 2x_4 + 2x_5 &\leq 0, \\ -2x_1 - x_2 - x_3 &\leq -3. \end{aligned}$$

Сформируем составное ограничение

$$-3x_1 - 2x_3 - 2x_4 + 2x_5 \leq -4$$

и применим к нему признак (4.38)

$$\sum_{j \in S} \min(a_{ij}, 0) = -4, \quad \sum_{j \in V} a_{ij}, x_j = 0, \quad -4 = -4.$$

Получаем $-4 = -4$, следовательно, ограничение выполняется, т. е. при $x_1 = 0$ не существует дополнения, улучшающего рекорд $z = 2$.

На основе полученных правил запишем алгоритм.

1. Пуст ли основной список задач? Если «да», то вычисления окончены и переходим к п. 5. Если «нет» — присвоим счетчику числа задач основного списка α значение $\alpha := \alpha - 1$, т. е. вычеркиваем одну задачу. Счетчику числа итераций $k := k + 1$ (в начале, очевидно, $k = 1$).

2. Существует ли допустимое дополнение при выбранном частичном решении, улучшающее значение рекорда z^* ? Проверим условие (4.38). Если «да», то определяем эти значения по формуле (4.39) и переходим к п. 3. Если «нет» — положим $z^{(k+1)} = z^k$ и переходим к п. 1.

3. Содержат ли допустимое дополнение частичного решения и зондируемое решение n переменных? Если «да»,

то запоминаем решение, значение рекорда z^* и переходим к п. 1. Если «нет» — переходим к п. 4.

4. Выбираем любую свободную переменную x_i , не входящую в число допустимых переменных. Вводим две задачи в основной список (в одной $x_i = 0$, в другой $x_i = 1$). Положим $z^k = z^{(k-1)}$ и переходим к п. 1.

Результат решения: рекорд и значения переменных либо «задача решения не имеет».

Конец алгоритма.

Ручное решение подобных задач занимает слишком много места (подробнее см. в [6], [39]).

ГЛАВА 5

ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ (ДП)

Это особый метод, который специально приспособлен для оптимизации динамических задач, в которых операция состоит из элементов, сильно влияющих друг на друга. Динамическое программирование связано с именем Ричарда Беллмана, который сформулировал *принцип оптимальности Беллмана*. Он позволяет существенно сократить перебор вариантов решений в многоэтапных нелинейных задачах.

Рассмотрим экономическую задачу распределения ресурсов: пусть есть начальный капитал k_0 . Его можно потратить на несколько предприятий P_1, P_2, \dots, P_n (рис. 21). X_{ij} — количество средств вкладываемых в i -м году в j -е предприятие. В результате получается эффект:

$$\omega_{ij} = f_{ij}(X_{ij}).$$

В общем случае f_{ij} — нелинейная функция. Необходимо распределить начальный капитал (ресурс) так, чтобы суммарный эффект от всех предприятий за все годы был максимальным:

$$W = \sum_{i=1}^m \sum_{j=1}^n f_{ij}(X_{ij}) \rightarrow \max,$$

$$\sum_{i=1}^m \sum_{j=1}^n X_{ij} \leq k_0.$$

Так как функция W нелинейная, то получаем задачу нелинейного программирования при очень большом числе переменных (см. главу 6). Решать ее сложно, кроме

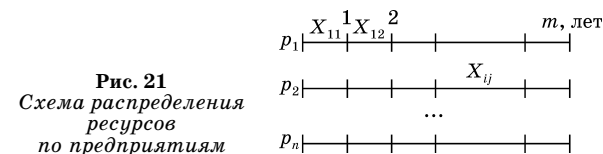


Рис. 21
Схема распределения
ресурсов
по предприятиям

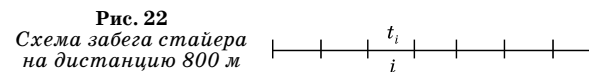


Рис. 22
Схема забега стайера
на дистанцию 800 м

того, часто X_{ij} — дискретные значения. Возникает вопрос: нельзя ли решить задачу последовательно, т. е. найти оптимальное вложение для первого года, второго и т. д., т. е. провести последовательную оптимизацию? В большинстве задач так делать нельзя, так как решение, принимаемое на одном шаге, оказывает влияние на последующие шаги. Рассмотрим простой пример.

Проанализируем процесс забега стайеров на 800 м (рис. 22).

Каждый бегун имеет запас энергии x_0 , x_i — затрата энергии на каждые 100 м; t_i — время бега стайера на отдельном участке i ; T — общее время забега на 800 м:

$$T = \sum_{i=1}^8 t_i(x_i) \rightarrow \min.$$

Казалось бы, чтобы сумма времен t_i была минимальной, нужно минимизировать каждое t_i , т. е. бежать как можно быстрее первую стометровку, затем как можно быстрее вторую и т. д. Читатель, вспомнив свой опыт бега, конечно, понял, что такой способ оптимизации неверен.

Если уменьшать каждое t_i отдельно, то на 800 м стайер затратит очень много времени, у него не будет сил продолжать бег. Значит, нужно бежать так, чтобы оставались силы на всю дистанцию.

Очевидно, каждый бегун решает задачу оптимального распределения ресурса своей энергии на дистанции:

$$T = \sum_{i=1}^8 t_i(x_i) \rightarrow \min, \quad \sum_{i=1}^8 x_i \leq x_0.$$

Оказывается, оптимальной величиной x_i и соответственно t_i будет такая, которая обеспечит минимальное общее время бега на все 800 м. Таким образом мы сформулировали принцип «действуй с прицелом на будущее». Другими словами оптимальность «в малом» необходимо понимать с точки зрения оптимальности «в большом». Это чрезвычайно важный принцип системного подхода к оптимизации систем.

5.1. ПРИНЦИП ОПТИМАЛЬНОСТИ Р. БЕЛЛМАНА

Метод ДП является наиболее общим методом решения задач оптимального управления. Он применим как для задач с линейной ЦФ, так и с нелинейной, а также в случае, когда управляемые переменные целые числа, при этом сама ЦФ может быть задана таблицей, что наиболее распространено в практических задачах.

Процессы называют динамическими, если результаты на одном участке процесса влияют на другие шаги.

Рассмотрим принцип оптимального управления Р. Беллмана. Он связан с проблемой оптимизации сложной системы, состоящей из многих взаимосвязанных элементов. Элементами могут быть экономические единицы, которые входят в единую (более общую) систему; узлы сложной технической системы; отдельные участки производства, строительства, боевых операций и т. д.

Возникает вопрос: как нужно управлять отдельными элементами системы, чтобы показатель эффективности всей системы был максимальным?

Мы на интуитивном уровне показали, что для оптимизации в целом недостаточно оптимизировать каждый элемент отдельно, это приводит к неверному результату.

Р. Беллман впервые сформулировал принцип оптимальности для такой задачи: оптимизируя отдельный шаг, необходимо задумываться о его последствиях, приводящих к общему результату. Для решения подобных задач был разработан метод ДП, основывающийся на уравнении Беллмана, которое математически выражает его принцип.

Назовем состоянием системы S один или несколько параметров системы. Например, деньги на лицевом счете предприятия. Обозначим управление на i -м шаге U_i — это некоторое воздействие, которое испытывает система и изменяет свое состояние S .

Если перед i -м шагом состояние системы S и мы принимаем управление U_i , то за i -й шаг мы можем получить некоторый выигрыш, который обозначается $\omega_i(S, U_i)$, при этом состояние S переходит в S' (рис. 23):

$$S \rightarrow S' = \varphi_i(S, U_i).$$

Считается, что функции $\omega_i(S, U_i)$ и $\varphi_i(S, U_i)$ известны.

Р. Беллман ввел понятие *условного оптимального выигрыша* $W_i(S)$. Эта функция показывает оптимальный выигрыш (наилучший результат), полученный за все шаги от i -го и до конца, если i -й шаг начинается с состояния S . Тогда согласно принципу оптимальности Беллмана, *принимая решение на i -м шаге, мы должны выбрать U_i так, чтобы выигрыш был максимальным от i -го шага и до конца.*

Принцип оптимальности Беллмана ставит вопрос о том, что такое оптимальность отдельного элемента системы с точки зрения оптимальности всей системы. Принимая решение на отдельном этапе, мы должны выбирать управление на этом этапе с прицелом на будущее, так как нас интересует результат в целом за все шаги.

Любой процесс имеет где-то окончание, т. е. говорят, что он имеет «горизонт планирования». Тогда последний этап «не имеет будущего». Вот именно его можно оптимизировать только из условий данного этапа. После этого приступают к оптимизации $(m-1)$ -го этапа. При этом мы задаем состояние, с которого начинается $(m-1)$ -й шаг (условие). Поэтому функцию $W_i(S)$ называют *условным оптимальным выигрышем*. Таким образом, процесс опти-

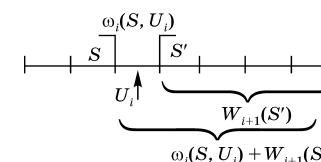


Рис. 23
Схема получения
условного оптимального
выигрыша

мизации по методу ДП разворачивается от конца к началу, а затем от начала к концу. В различных задачах может быть известно либо начальное состояние, либо конечное, либо то и другое. Принцип Беллмана нашел практическое применение в так называемом методе программно-целевого планирования (любое действие планируется как элемент некоторого проекта).

5.2. РЕШЕНИЕ ГРАФОВЫХ ЗАДАЧ НА ОСНОВЕ ПРИНЦИПА БЕЛЛМАНА

Задача о наборе высоты и скорости летательного аппарата.

Летательный аппарат находится на высоте h_0 и летит со скоростью v_0 . Необходимо перевести его на высоту h_1 , придав скорость v_1 (рис. 24). Причем $h_0 > h_1$, $v_0 > v_1$. Разобьем участок от h_0 до h_1 на n частей:

$$\Delta h = \frac{h_1 - h_0}{n}, \quad \Delta v = \frac{v_1 - v_0}{m}.$$

Известен расход горючего при переводе системы на Δh при $v = \text{const}$ и на Δv при $h = \text{const}$. Таким образом, из каждого состояния есть лишь два управления.

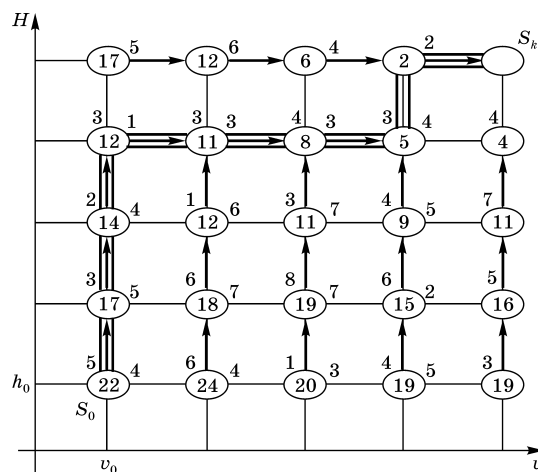


Рис. 24
Геометрическая
иллюстрация
решения задачи
о наборе высоты
и скорости
летательного
аппарата

Начиная с конца, помечаем все узлы (состояния) величинами условных (для данного узла) оптимальных расходов горючего от этого узла и до конца, а стрелками — условные оптимальные управления. Указанные действия в упрощенном виде демонстрируют процедуру решения на основе уравнения Беллмана. Дойдя от конечного состояния до начального и получив 22, мы будем иметь минимальную величину потери горючего. Идя по стрелкам от начального состояния до конечного, мы получаем безусловные оптимальные управления (показаны двойной линией).

Видно, что любая задача, сводящаяся к поиску минимального пути в графе, решается методом динамического программирования.

5.3. ФУНКЦИОНАЛЬНОЕ УРАВНЕНИЕ БЕЛЛМАНА

Назовем состоянием системы вектор координат: $S = (\xi_1, \xi_2, \dots, \xi_L)$. В некоторых задачах состояние — одна величина. Тогда работу системы можно представить как движение в фазовом пространстве — пространстве состояний. Назовем *шаговым управлением* управление на i -м шаге. Рассмотрим процесс управления системой за m шагов. Функция $\omega_i(S, U_i)$ называется *выигрышем на i -м шаге*. Здесь S — состояние перед i -м шагом, а U_i — управление на i -м шаге.

Величина $\omega_i(S, U_i)$ должна быть известна до начала динамического программирования. Если состояние перед i -м шагом было S и мы приняли какое-то управление U_i , то система перейдет в новое состояние $S' = \phi_i(S, U_i)$. Эта функция должна быть также известна. Если эти функции не заданы, то их надо сформулировать. Введем функцию $W_i(S)$ — *условный оптимальный выигрыш*. Это выигрыш на всех этапах от i до конца, если i -й шаг начинается с состояния S .

Рассмотрим m шагов. Пусть с $(i + 1)$ -го шага мы системой управляем оптимально, тогда величина выигрыша будет такая: $W_{i+1}(S')$. Применим на i -м шаге произвольное управление U_i , тогда $\bar{W}_i(S)$ — неоптимальный выигрыш, так как на i -м шаге мы применяем неоптимальное

управление U_i . Чтобы от i -го шага и до конца получить оптимальный выигрыш, нужно изменять U_i так, чтобы

$$W_i(S) = \max_{U_i} \{ \omega_i(S, U_i) + W_{i+1}(S') \}; \quad S' = \varphi_i(S, U_i),$$

$$W_i(S) = \max_{U_i} \left\{ \begin{array}{ll} \omega_i(S, U_i) & \text{известно} \\ W_{i+1}[\varphi_i(S, U_i)] & \text{неизвестно} \end{array} \right\}.$$

Это *функциональное уравнение Беллмана*. Для использования уравнения Беллмана начинают с конца:

1. $i = m$,

$$W_m(S) = \max_{U_m} \{ \omega_m(S, U_m) \};$$

2. $i = m - 1$,

$$W_{m-1}(S) = \max_{U_{m-1}} \{ \omega_{m-1}(S, U_{m-1}) + W_m[\varphi_{m-1}(S, U_{m-1})] \}.$$

Итак, идя от конца к началу, мы получаем последовательно:

$$W_m(S), W_{m-1}(S), \dots, W_1(S), \\ U_m(S), U_{m-1}(S), \dots, U_1(S).$$

На рис. 25, 26 поясняются выполняемые действия.

Придя в начальное состояние $W_1(S)$, мы можем подставить $S = S_0$ и $W_1(S_0) = W_{\max}$ — это безусловный выигрыш.

Теперь необходимо получить безусловные оптимальные уравнения, идя от начала к концу по цепочке:

$$S = S_0 \rightarrow U_1(S_0) = U_1^* \rightarrow \\ \rightarrow \varphi(S_0, U_1^*) = S_1^* \rightarrow U_2(S_1^*) = U_2^* \rightarrow \varphi(S_1^*, U_2^*) \dots$$

Итак, в результате мы получаем оптимальное решение:

$$U_1^*, U_2^*, \dots, U_m^*; W_{\max}.$$

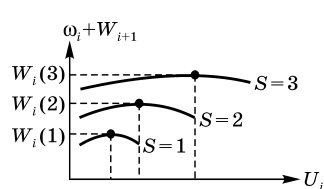


Рис. 25
Получение функций
Беллмана

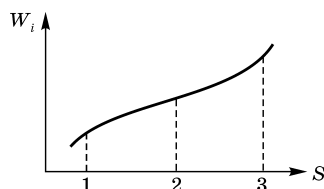


Рис. 26
Функция условного
оптимального выигрыша

5.4. ЗАДАЧИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ

5.4.1. КЛАССИЧЕСКАЯ ЗАДАЧА РАСПРЕДЕЛЕНИЯ РЕСУРСОВ

Распределение ресурсов — это едва ли не самая распространенная операция. Под ресурсом в общем случае понимают физическую или абстрактную величину, которую система использует для производства полезного продукта. Например, горючее, деньги, время, объем склада. Как правило, ресурс ограничен, поэтому встает задача так распределить ресурс между отдельными элементами системы, чтобы суммарный эффект был максимальным. Рассмотрим классическую задачу распределения ресурсов.

Имеется начальное количество ресурсов k_0 , которое необходимо распределить между двумя отраслями. Каждая отрасль работает в течение m лет. Если в первую отрасль в i -й год вкладываются средства X_i , то доход $f(X_i)$, если же во вторую вкладываются Y_i , тогда доход $g(Y_i)$. Средства тратятся, принося доход, а новых средств не поступает и полученный доход не вкладывается.

Нас интересует суммарный доход:

$$W = \sum_{i=1}^m f(X_i) + g(Y_i).$$

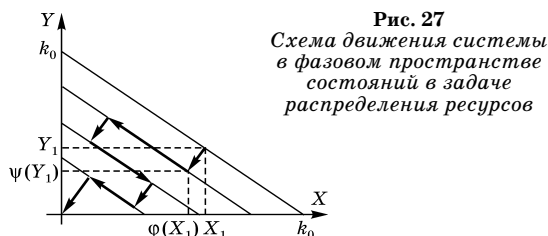
Суммарный выигрыш равен сумме выигрышей на каждом шаге. Состоянием системы является количество средств перед i -м шагом. Так как новых средств не поступает, то ресурсы «тают».

Управление Y_i может быть записано как $Y_i = k - X_i$. После i -го шага в первой отрасли остаются средства $\varphi(X_i)$, а во второй $\psi(Y_i) = \psi(k - X_i)$. Эти функции называются *функциями траты*. Мы можем составить уравнение Беллмана. В этой задаче на i -м шаге одно управление X_i и одно состояние k .

$$W_i(k) = \max_{X_i} \{ f(X_i) + g(k - X_i) + W_{i+1}[\varphi(X_i) + \psi(k - X_i)] \};$$

$$i = m: W_m(k) = \max_{X_m} \{ f(X_m) + g(k - X_m) \} \text{ и т. д.};$$

$$W_i(k), \quad k = k_0, \quad W_1(k_0) = W_{\max}; \quad X_1(k_0) = X_1^*, \quad Y_1^* = k_0 - X_1^*.$$



Используя функции траты, получим количество средств после i -го шага:

$$\varphi(X_1^*) + \psi(Y_1^*) = k_1^*; \quad X_2(k_1^*) = X_2^* \quad \text{и т. д.}$$

Задача о распределении ресурсов допускает геометрическую интерпретацию (рис. 27):

$$X_1 + Y_1 = k_0.$$

Распределение на первом шаге — указание точки на гипотенузе. После этого средства тратятся. Распределение средств — движение внутрь треугольника. Рассмотрим частные случаи задач о распределении ресурсов.

5.4.2. НЕОДНОРОДНЫЕ ЭТАПЫ И РАСПРЕДЕЛЕНИЕ РЕСУРСОВ ПО n ОТРАСЛЯМ

Распределение по неоднородным этапам. Мы считали, что все функции одинаковы на всех этапах. Во многих задачах они меняются от этапа к этапу: $f_i(X_i)$, $g_i(Y_i)$; $\varphi_i(X_i)$, $\psi_i(Y_i)$.

Покажем, что процедура динамического программирования принципиально не меняется. Запишем уравнение Беллмана:

$$W_i(k) = \max_{X_i} \{f_i(X_i) + g_i(k - X_i) + W_{i+1}[\varphi_i(X_i) + \psi_i(k - X_i)]\}.$$

Видно, что при решении достаточно на каждом этапе всего лишь подставлять разные функции.

Распределение ресурсов между тремя и более отраслями. В этом случае на каждом шаге будет уже n управлений, но одно из них может быть выражено как

$$X_i^n = k - \sum_{j=1}^{n-1} X_i^j.$$

Тогда в правой части уравнения Беллмана будет две или более переменных, по которым ищется максимум, и задача существенно усложняется.

5.4.3. РАСПРЕДЕЛЕНИЕ РЕСУРСОВ С РЕЗЕРВИРОВАНИЕМ

В такой модели если средства распределяются между двумя отраслями, то какое-то количество средств можно оставить до последующего распределения. В этом случае задача имеет смысл даже для одной отрасли.

Начальное количество средств разделяется на первом этапе на X_1 и на $k - X_1$ (резерв), на втором этапе подлежат распределению оставшиеся средства и из резерва. Такую задачу можно представить с одной реальной отраслью, а другой — фиктивной (не приносящей дохода и не расходующей средства). На рис. 28 показаны графики распределения ресурсов с резервированием (рис. 28а) и в дополнение к этому с нулевыми функциями трат (рис. 28б).

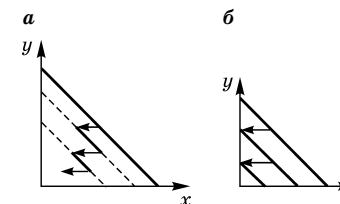


Рис. 28
Геометрическая интерпретация распределения ресурсов с резервированием

Решение такой задачи сводится к классическому решению путем задания функций дохода и трат в виде $f(X)$, $g(Y) = 0$ — функции дохода; $\varphi(X)$, $\psi(Y) = Y$ — функции трат.

Задача с резервированием в одной отрасли при нулевых функциях трат. Пусть x_i — количество средств, выделяемых в i -й этап, а все функции траты $\varphi_i(x_i) = 0$, т. е. вложенные средства тратятся полностью.

В этом случае задача сводится к более простой:

$$\varphi_i(x_i) = 0,$$

$$W = \sum_{i=1}^m f_i(x_i) \rightarrow \max, \quad \sum_{i=1}^m x_i \leq k_0.$$

Рассмотрим более частный случай — все функции одинаковые на всех шагах:

$$f_i(x) = f(x), \forall i.$$

Пусть эти функции неубывающие, тогда недоиспользование средств невыгодно и в ограничении можно поставить равенство:

$$W = \sum_{i=1}^m f(x_i) \rightarrow \max, \quad (5.1)$$

$$\sum_{i=1}^m x_i = k_0. \quad (5.2)$$

Эта задача имеет теоретическое решение:

1. Если функция $f(x)$ неубывающая и выпуклая, то оптимальным распределением является равномерное распределение: $x_i = k_0/m$.

2. Если функция $f(x)$ неубывающая и вогнутая, то оптимальное распределение будет таким: все распределение в один этап (элемент) и ничего в другие: $x_r = k_0, x_i = 0$ для всех i , кроме $i = r$.

Покажем это на графических примерах (рис. 29, 30).

1. Пусть $f(x)$ — выпуклая, например

$$f(x) = \sqrt{x}, \quad W = \sqrt{x_1} + \sqrt{x_2} \rightarrow \max, \quad x_1 + x_2 = k_0.$$

Тогда линии постоянного уровня W — это дуги линий астроида, симметричных относительно биссектрисы

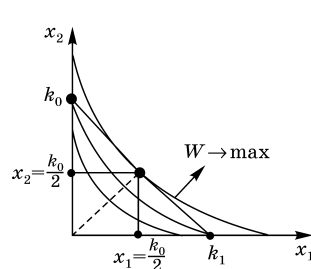


Рис. 29
Распределение ресурсов для случая функции дохода выпуклой вверх

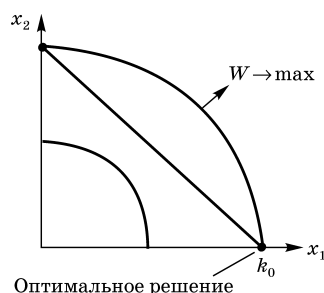


Рис. 30
Распределение ресурсов для случая функции дохода выпуклой вниз

координатного угла. Поэтому последнее касание астроида гипотенузы треугольника при $W \rightarrow \max$ будет в точке $x_i = k_0/2$. Очевидно, для m шагов будет $x_i = k_0/m$.

2. Пусть $f(x)$ — вогнутая, например

$$f(x) = x^2, \quad W = x_1^2 + x_2^2 \rightarrow \max, \quad x_1 + x_2 = k_1.$$

Из рис. 30 видно, что в этом случае решение будет лежать на оси координат, т. е. $x_1 = k_0, x_2 = 0$ либо $x_2 = k_0, x_1 = 0$.

5.4.4. РАСПРЕДЕЛЕНИЕ РЕСУРСОВ С «ВЛОЖЕНИЕМ ДОХОДОВ»

В классической задаче считается, что полученный на i -ом шаге доход в производство не вкладывается, т. е. он отчисляется и подсчитывается как эффект. Во многих задачах полученный эффект можно использовать как ресурс для следующего шага, объединив его с оставшимся ресурсом. Если ресурс не деньги, то средства нужно привести к единому эквиваленту с оставшимися средствами. Такая модель является развитием классической. Так как оставшиеся средства и доход объединяются, то можно ввести единую интегральную функцию — функцию изменения средств $F(X_i)$ — количество оставшихся средств плюс доход после i -го шага, если вложили X_i :

$$\begin{aligned} &\text{I. } F(X_i), \\ &\text{II. } G(Y_i) = G(k - X_i), \end{aligned}$$

где k — количество средств перед i -м шагом.

Выигрыш на i -м шаге зависит от того, как мы подсчитываем доход (эффект) от управления всеми ресурсами. Поставим задачу: максимизировать доход в конце m -го шага. Тогда на всех шагах $i = 1, m-1$, доход $= 0$, $\omega_i = 0$. На m -м шаге выигрыш $\omega_m = F_m(X_m) + G_m(k - X_m)$. Подставив эти выражения в уравнение Беллмана, мы программируем задачу от начала к концу, если имеется начальное количество средств k_0 . Здесь функция траты: $k' = F_i(X_i) + G_i(k - X_i)$.

Рассмотрим частный случай, когда F и G — неубывающие функции. В этом случае, чем больше значение до-

ход + средства получается в конце i -го шага, тем лучшим условием это будет для проведения $(i + 1)$ -го шага. Поэтому можно не заботиться о следующих шагах, достаточно обеспечить максимум на каждом шаге.

Таким образом, процедура оптимизации возможна в одном направлении — от начала к концу, т. е. задача динамического программирования вырождается в задачу последовательной оптимизации:

$$\begin{aligned} \max_{X_1} \{F_1(X_1) + G_1(k_0 - X_1)\} &= k_1^*, \\ \max_{X_2} \{F_2(X_2) + G_2(k_1^* - X_2)\} &= k_2^*, \\ \dots \\ \max_{X_m} \{F_m(X_m) + G_m(k_{m-1}^* - X_m)\} &= k_m^*, \quad W_{\max} = k_m^*. \end{aligned}$$

Рассмотрим задачу распределения ресурсов с вложением доходов в производство и отчислением. Это наиболее общий случай. Разделим функции дохода и функции траты:

$$\begin{aligned} f(X_i), \quad g(Y_i), \\ \varphi(X_i), \quad \psi(Y_i) \end{aligned}$$

и максимизируем суммарный отчисленный *доход + оставшиеся средства* после m -го шага. Введем функцию отчисления $r_i(D)$; D — доход. Тогда выигрыш на каждом шаге:

$$\begin{aligned} \omega_i &= r_i[f(X_i) + g(k - X_i)], \quad i = \overline{1, m-1}, \\ \omega_m &= r_m[f(X_m) + g(k - X_m)] + \varphi(X_m) + \psi(k - X_m), \\ W &= \sum_{i=1}^m \omega_i \rightarrow \max, \\ k' &= \varphi(X_i) + \psi(k - X_i) + f(X_i) + g(k - X_i) - r_i[f(X_i) + g(k - X_i)]. \end{aligned} \quad (5.3)$$

Уравнение Беллмана для i -го шага будет выглядеть так:

$$\begin{aligned} W_i(k) &= \max_{X_i} \{r_i[f(X_i) + g(k - X_i)] + W_{i+1}[\varphi(X_i) + \psi(k - X_i) + \\ &\quad + f(X_i) + g(k - X_i) - r_i[f(X_i) + g(k - X_i)]]\}, \quad i = \overline{1, m-1} \end{aligned}$$

для $i = m$ надо учесть (5.3).

Если $r_i(D) = D$, то мы получаем классическую задачу.

5.5. РАСШИРЕНИЕ МОДЕЛИ ЗАДАЧ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Учет предыстории процесса. Мы считали, что все функции, как выигрыша, так и траты, зависят от состояния перед i -м шагом, т. е. не зависят от более ранних состояний. Такие процессы называются *процессами без памяти*. Но иногда при рассмотрении некоторых процессов требуется помнить всю историю происходящего (например, в задачах ремонта и замены оборудования). Такая задача более сложная. Введем расширенное состояние:

$$S = (S, S_{i-1}, S_{i-2}, \dots, S_{i-L}),$$

S_{i-L} — состояние за L шагов до i -го. Тогда, если считать аргументы векторами, функции $\omega_i(S, U_i)$, $\varphi_i(S, U_i)$ не меняются. Но задача становится сложнее в вычислительном аспекте. Пусть S имеет p координат и предыстория распространяется на L шагов, тогда в результате число переменных будет $p \times L$. Вот почему подобные задачи практически можно решать, если $p \times L \leq 3$.

ДП для задач со многими ограничениями. Пусть в задаче распределения ресурсов есть два ограничения (два ресурса):

$$\left. \begin{aligned} W &= \sum_{i=1}^m f(X_i) \rightarrow \max; \\ \sum_{i=1}^m a_{1i} X_i &\leq b_1; \\ \sum_{i=1}^m a_{2i} X_i &\leq b_2. \end{aligned} \right\}$$

Введем два параметра состояния K_1 и K_2 . Тогда функции условных оптимальных выигрышей и условных оптимальных управлений будут иметь по два аргумента: $W(K_1, K_2)$; $X_i^*(K_1, K_2)$. Уравнение Беллмана для решения такой задачи имеет вид

$$W_i(K_1, K_2) = \max_{X_i} \{f_i(X_i) + W_{i+1}[K_1 - a_{1i} X_i; K_2 - a_{2i} X_i]\}.$$

При этом величина X_i на каждом этапе варьируется от нуля до d_i :

$$d_i = \min(K_1/a_{1i}; K_2/a_{2i}).$$

Видно, что для двух ограничений, если каждая из величин K_1, K_2 может принимать 10^2 значений, функцию $W_i(K_1, K_2)$ приходится табулировать в 10^4 точках. Таким образом, наиболее серьезным препятствием для практического применения ДП является число параметров задачи, что заставило Р. Беллмана заявить о так называемом «проклятии многомерности». Некоторые способы понижения размерности смотри в работе [10].

Задача с мультипликативным критерием. До сих пор мы считали, что суммарный выигрыш равен сумме выигрышей, полученных на i -х шагах. Но есть задачи, где общий критерий равен произведению критериальных величин на каждом шаге:

$$W = \prod_{i=1}^m \omega_i.$$

В этом случае также можно применить уравнение Беллмана, где вместо W можно взять функцию $W' = \ln W$. Оптимальные решения будут одинаковы ввиду монотонности логарифмической функции. Но можно при выводе уравнения Беллмана учесть, что

$$W_i(S) = \max_{U_i} \{\omega_i \cdot W_{i+1}(S')\}.$$

Таким образом, метод ДП применим в случае, когда аргументы в критериальной функции разделяются.

Пример.

Устройство состоит из n узлов. Имеется некоторый ресурс k_0 , который может использоваться для повышения надежности каждого узла. Необходимо так распределить ресурс, чтобы суммарная надежность была максимальной. Получаем задачу

$$Q = \prod_{i=1}^m q_i(X_i) \rightarrow \max,$$

$$\sum_{i=1}^m X_i \leq k_0,$$

где $q_i(X_i)$ — надежность каждого узла.

Операции, не связанные со временем. Во многих задачах распределение ресурсов не связано с временными шагами. Ресурс распределяется по объектам. Например, если рассмотреть распределение ресурсов между n объектами, когда на каждый объект задана функция выигрыша, то такая задача эквивалентна рассмотренной нами задаче о распределении ресурсов с резервированием в одной отрасли по n шагам. Это наиболее распространенная задача распределения ресурсов, поэтому приведем ее полное решение методом ДП.

5.6. ПРИМЕР РЕШЕНИЯ ЗАДАЧИ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ

Рассмотрим пример. Завод получил 4 новых станка, которые необходимо распределить между тремя цехами. Известна дополнительная прибыль, которая получится в i -м цехе, если туда поставят x_i станков — $f(x_i)$ (рис. 31).

Состоянием является оставшееся (текущее) количество станков (k). Пусть мы распределяем оставшиеся k станков. Начинаем с третьего (последнего) цеха:

$$W_3(k) = \max_{x_3 \leq k} \{f_3(x_3)\}.$$

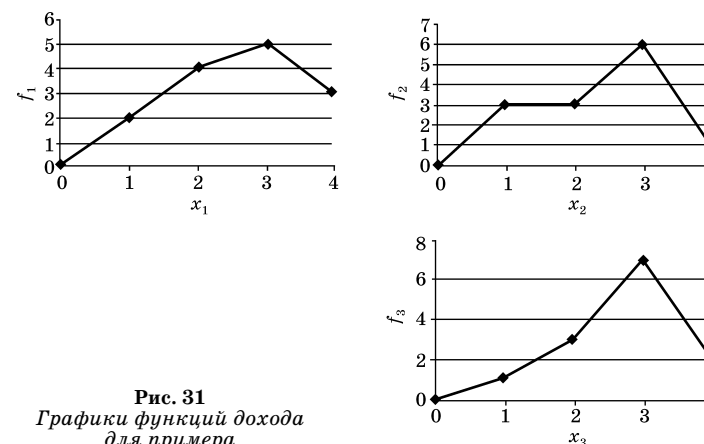


Рис. 31
Графики функций дохода
для примера

Составляем табл. 33, 34.

Переходим ко второму шагу:

$$W_2(k) = \max_{x_2 \leq k} \{f_2(x_2) + W_3(k - x_2)\}.$$

Составляем табл. 35, 36.

Переходим к первому шагу:

$$W_1(k) = \max_{x_1 \leq 4} \{f_1(x_1) + W_2(k - x_1)\}.$$

Составляем табл. 37.

Таблица 33

k	x_3	\tilde{W}_3	W_3
0	0	0	0
1	0 <u>1</u>	0 1	1
2	0 1 <u>2</u>	0 1 3	3
3	0 1 2 <u>3</u>	0 1 3 7	7
4	0 1 2 <u>3</u> 4	0 1 3 7 2	7

Таблица 35

k	x_2	\tilde{W}_2	W_2
0	0	0	0
1	0 <u>1</u>	1 3	3
2	0 <u>1</u> 2	3 4 3	4
3	<u>0</u> 1 2 3	7 6 4 6	7
4	0 <u>1</u> 2 3 4	7 10 6 7 1	10

Таблица 34

k	$x_3^{\text{опт}}$	W_3
0	0	0
1	1	1
2	2	3
3	3	7
4	3	7

Таблица 36

k	$x_2^{\text{опт}}$	W_2
0	0	0
1	1	3
2	1	4
3	0	7
4	1	10

Таблица 37

k	x_1	\tilde{W}_1	W_1
	<u>0</u>	10	
	1	9	
4	2	8	10
	3	8	
	4	3	

Так как начальное состояние известно, то для первого шага не нужно перебирать оставшееся число станков и берем $k = 4$.

О т в е т:

$$x_1^* = 0, \quad k_1^* = 4 - 0 = 4 \quad (\text{подставляем } k_1^* \text{ в табл. 36}),$$

$$x_2^* = 1, \quad k_2^* = 4 - 1 = 3 \quad (\text{подставляем } k_2^* \text{ в табл. 34}),$$

$$x_3^* = 3, \quad k_3^* = 4 - 3 = 1.$$

5.7. ЭФФЕКТИВНОСТЬ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Сравним по числу необходимых операций динамическое программирование с простым перебором всех вариантов для задачи

$$W = \sum_{i=1}^n f(x_i) \rightarrow \max,$$

$$\sum_{i=1}^n a_i x_i \leq b.$$

Для упрощения расчета примем $a_1 = a_2 = \dots = a_n = 1$.

При простом переборе число возможных вариантов решений (при условии целочисленности всех переменных) равно числу способов, которыми можно разместить b одинаковых шаров внутри n урн.

Оно составляет

$$C_{n+b-1}^b = \frac{(n+b-1)!}{b!(n-1)!}.$$

При $n = 5, b = 20$ имеем $C_{24}^{20} = 10\,626$.

Оценим число операций, требуемых для решения этой задачи, методом динамического программирования.

Для вычисления $W_i(k)$ при фиксированном k необходимо провести $k + 1$ вычислений. Поэтому для заполнения одной таблицы $W_i(k)$ необходимо проделать

$$\sum_{k=0}^b (k+1) = \frac{(b+1)(b+2)}{2} \quad \text{операций.}$$

Следовательно, для вычисления всех функций $W_1(k)$, ..., $W_{n-1}(k)$ необходимо

$$(n-1) \frac{(b+1)(b+2)}{2} \text{ операций.}$$

С учетом вычисления функции $W_n(b)$ общее число операций

$$N_{\Sigma} = \frac{(n-1)(b+1)(b+2)}{2} + (b+1) = \frac{(b+1)[(n-1)(n+2)+2]}{2}.$$

При $n = 5$, $b = 20$ имеем $N_{\Sigma} = 945$, что в 11 раз меньше, чем при простом переборе.

ГЛАВА 6 НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

6.1. ОСОБЕННОСТИ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Нелинейное программирование (НЛП) — это такая задача математического программирования, в которой либо целевая функция, либо функции ограничений, либо они вместе представляют собой нелинейные функции. Термин НЛП применяется только тогда, когда все переменные X — действительные числа. Решение задач НЛП намного сложнее решения задач ЛП при прочих равных условиях. Рассмотрим особенности задач НЛП, а именно то, где может находиться оптимальное решение. Для этого возьмем целевую функцию двух переменных. В НЛП используют геометрический образ целевой функции как некоторой поверхности, определенной на множестве управляемых переменных. Вид этой поверхности можно представить, изображая в области переменных так называемые линии уровня (см. рис. 32).

Особенности задач НЛП проиллюстрируем графически на примере задачи:

$$\begin{cases} F = \left(x_1 - \frac{1}{2}\right)^2 + (x_2 - 1)^2 \rightarrow \min, \\ x_1 x_2 \leq a, \\ x_1 + x_2 \geq b, \\ 0 \leq x_1 \leq 3; \quad 0 \leq x_2 \leq 3. \end{cases}$$

Изобразим графически область допустимых решений (ОДР) и найдем решение для различных значений a и b . Очевидно, что оптимальное решение будет находиться

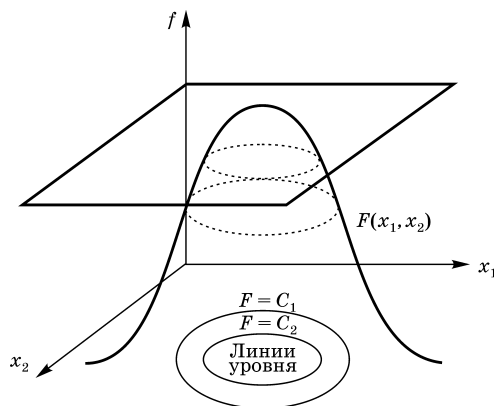


Рис. 32

График нелинейной целевой функции и ее линии уровня

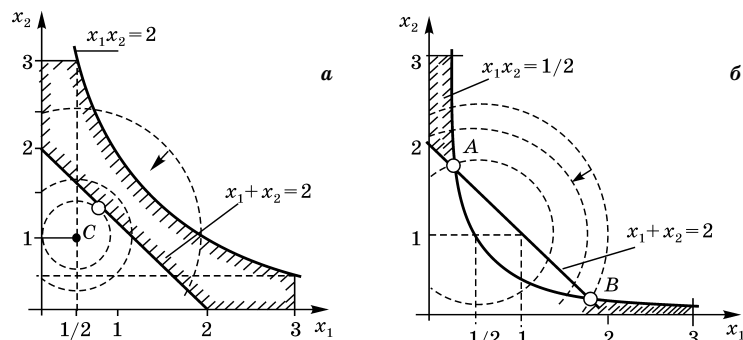


Рис. 33

Местоположение оптимального решения для задачи НЛП

в точке, где линия уровня имеет самую малую величину в области допустимых решений (ОДР). Это можно представить как «последнее касание» линии уровня ОДР по мере уменьшения этого уровня. Линии уровня в этой задаче представляют собой концентрические окружности, а $F \rightarrow \min$ соответствует уменьшению их радиуса.

1. $a = 2$, $b = 2$ (рис. 33а);

2. $a = \frac{1}{2}$, $b = 2$ (рис. 33б).

Решение задачи на рис. 33а находится в точке С, где линия наименьшего уровня целевой функции касается прямой $x_1 + x_2 = 2$. Координаты оптимального решения $x_1 = 3/4$, $x_2 = 5/4$ можно найти из простых геометрических соображений.

Решение задачи на рис. 33б качественно отличается от первого случая, так как здесь получаются две несвязанные ОДР, дающие две экстремальные точки А и В, являющиеся точками пересечения линий

$$x_1 x_2 = \frac{1}{2} \quad \text{и} \quad x_1 + x_2 = 2.$$

Точка В является точкой локального минимума, а точка А — точкой глобального минимума. Ее координаты найдем, решив систему уравнений:

$$\begin{cases} x_1 x_2 = \frac{1}{2}, \\ x_1 + x_2 = 2, \end{cases}$$

$$x_1^A = 1 - \frac{\sqrt{2}}{2} \approx 0,3, \quad x_2^A = 1 + \frac{\sqrt{2}}{2} \approx 1,7.$$

3. $a = 2$, $b = 1$.

Решение на рис. 34 находится в точке D с координатами центра концентрических окружностей (линий уровня целевой функции).

Из приведенных примеров ясно, что решение задачи НЛП может находиться как внутри ОДР (рис. 34), на одной из границ в произвольной точке (см. рис. 33а), так и в вершинах ОДР, являющихся точками пересечений функций ограничений (см. рис. 33б), причем может получиться несколько экстремальных точек.

Графическое изображение ОДР и линий уровня $F(x)$ позволяет решать задачи НЛП с двумя переменными.

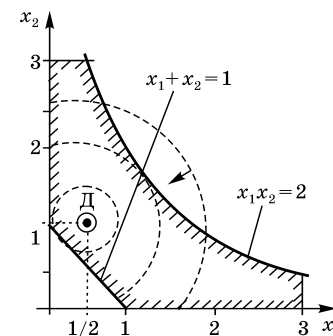


Рис. 34

Случай расположения оптимального решения внутри ОДР

6.2. ПРЯМЫЕ МЕТОДЫ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ НЕЛИНЕЙНЫХ ФУНКЦИЙ БЕЗ ОГРАНИЧЕНИЙ

Пусть на множестве $U \in R$ определена функция одной переменной $f(x)$. Под *минимизацией функции* $f(x)$ на множестве U будем понимать решение следующей задачи: найти хотя бы одну точку минимума x^* и минимум $f^* = f(x^*)$ этой функции на множестве U .

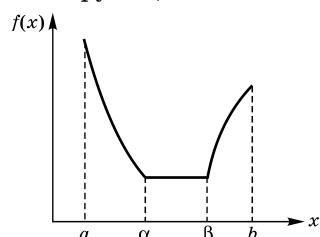


Рис. 35
График унимодальной функции

Многие приближенные методы минимизации применимы только тогда, когда любой локальный минимум $f(x)$ является одновременно и глобальным. Одним из классов функций, удовлетворяющих этому условию, является класс *унимодальных функций* (рис. 35).

Функция $f(x)$ называется *унимодальной* на отрезке $[a; b]$, если она непрерывна на $[a; b]$ и существуют числа α и β , $a \leq \alpha \leq \beta \leq b$, такие, что:

- 1) если $a < \alpha$, то на отрезке $[a; \alpha]$ $f(x)$ монотонно убывает;
- 2) если $\beta < b$, то на отрезке $[\beta; b]$ $f(x)$ монотонно возрастает;

$$3) \text{ при } x \in [\alpha; \beta] \quad f(x) = f^* = \min_{[a; b]} f(x).$$

Для проверки унимодальности функции $f(x)$ на практике обычно используют следующие критерии:

1) если функция $f(x)$ дифференцируема на отрезке $[a; b]$ и производная $f'(x)$ не убывает на этом отрезке, то $f(x)$ унимодальна;

2) если функция $f(x)$ дважды дифференцируема на отрезке $[a; b]$ и $f''(x) \geq 0$ при $x \in [a; b]$, то $f(x)$ также унимодальна.

Большую группу приближенных методов минимизации функций составляют прямые методы минимизации, основанные на вычислении только значений минимизируемой

функции в некоторых точках и не использующие значений ее производных.

Метод перебора (пассивный поиск) является простейшим из прямых методов минимизации. Пусть $f(x)$ унимодальна и требуется найти какую-либо из точек минимума x^* функции $f(x)$ на отрезке $[a; b]$ с абсолютной погрешностью $\varepsilon > 0$. Разобьем $[a; b]$ на n равных частей точками деления $x_i = a + i(b - a)/n$, $i = 0, 1, 2, \dots, n$, где $n \geq (b - a)/\varepsilon$. Вычислив значения $f(x)$ в этих точках, путем сравнения найдем точку x_m , для которой

$$f(x_m) = \min f(x_i); \quad 0 \leq i \leq n. \quad (6.1)$$

Далее полагаем $x^* = x_m$, $f^* = f(x_m)$. При этом максимальная погрешность ε_n определения точки x^* равна $\varepsilon_n = (b - a)/n$.

Метод деления отрезка пополам (метод дихотомии) является простейшим последовательным методом минимизации. Он позволяет для любой унимодальной функции $f(x)$ построить последовательность вложенных отрезков $[a; b] [a_1; b_1] \dots [a_{n-1}; b_{n-1}] [a_n; b_n]$, каждый из которых содержит хотя бы одну из точек минимума x^* функции $f(x)$.

Пусть $\varepsilon > 0$ — требуемая точность определения точки x^* . Выбрав $\delta = 2\varepsilon$, построим последовательности

$$\{a_n\}, \{b_n\}, \{x_1^{(n)}\}, \{x_2^{(n)}\}, n = 0, 1, \dots,$$

используя рекуррентные формулы

$$\begin{aligned} a_0 &= a, \quad b_0 = b; \\ x_1^{(n-1)} &= \frac{(a_{n-1} + b_{n-1} - \delta)}{2}, \quad x_2^{(n-1)} = \frac{(a_{n-1} + b_{n-1} + \delta)}{2}; \\ a_n &= a_{n-1}, \quad b_n = x_2^{(n-1)}, \quad \text{если } f(x_1^{(n-1)}) \leq f(x_2^{(n-1)}); \\ a_n &= x_1^{(n-1)}, \quad b_n = b_{n-1}, \quad \text{если } f(x_1^{(n-1)}) > f(x_2^{(n-1)}). \end{aligned} \quad (6.2)$$

Переход от отрезка $[a_{n-1}; b_{n-1}]$ к отрезку $[a_n; b_n]$ методом деления отрезка пополам иллюстрируется на рис. 36а, если $f(x_1^{(n-1)}) < f(x_2^{(n-1)})$, и на рис. 36б, если $f(x_1^{(n-1)}) > f(x_2^{(n-1)})$.

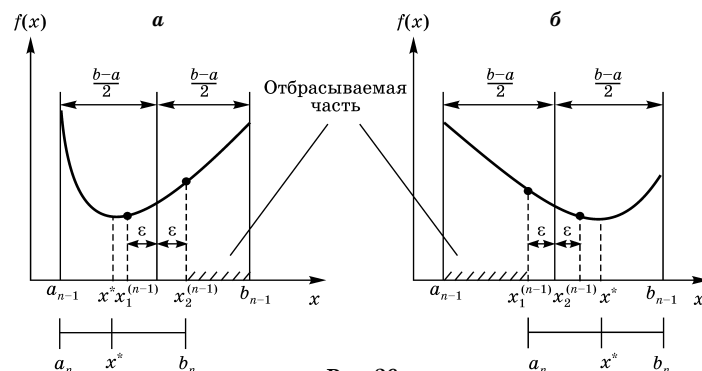


Рис. 36
Метод деления отрезка пополам

Полагая $x^* = (a_n + b_n)/2$, находим x^* с абсолютной погрешностью, не превосходящей величины

$$\varepsilon_n = \frac{b_n - a_n}{2} = \frac{b - a - \delta}{2n} + 1 + \frac{\delta}{2}.$$

Кроме этих методов известны и другие, в ряде случаев более эффективные методы, такие как: *метод Фибоначчи*, *золотого сечения*, *поиска по дискретным точкам* [12], [25].

6.3. ГРАДИЕНТНЫЕ МЕТОДЫ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ

Если ограничения в задаче НЛП отсутствуют, то применяют все методы оптимизации нелинейных функций. Причем если в задаче с ограничениями решение гарантированно находится внутри области, то его можно найти этими методами, так как эти ограничения не являются активными и их можно отбросить. Рассмотрим наиболее широко применяемые методы оптимизации нелинейных функций многих переменных — это так называемые *поисковые методы*. Здесь поиск оптимального решения начинают из начальной точки (ее выбирают по информации о примерном месте нахождения решения). Далее в соответствии с некоторым разумным правилом переходят в новую точку с лучшим значением ЦФ и так до тех пор, пока не выполнится некоторое заранее заданное правило остановки, например условие по точности.

6.3.1. КЛАССИЧЕСКИЙ ГРАДИЕНТНЫЙ МЕТОД

В этом методе поиска каждая последующая точка получается путем перехода по вектору градиента, т. е. по нормали к линии уровня. Длина шага выбирается из двух основных условий: если длина шага очень большая, то можно перейти точку экстремума, если очень маленькая, то количество шагов от начальной точки до оптимальной будет очень велико. Поиск начинается из начальной точки X^0 . Точка $X^{(k+1)}$ ищется по формуле

$$X^{(k+1)} = X^{(k)} \pm \lambda^{(k)} g(X^{(k)}),$$

(+) — если $F(X) \rightarrow \max$ и (–) — если $F(X) \rightarrow \min$,

где $g(X^{(k)})$ — градиент в точке $X^{(k)}$; $\lambda^{(k)}$ — множитель, определяющий длину шага.

В координатах:

$$X_j^{(k)} = X_j^{(k+1)} + \lambda^{(k)} \left(\frac{\partial F}{\partial X_j} \right)_{X^{(k)}}.$$

Величина $\lambda^{(k)}$ может в простейшем случае не изменяться по мере движения $\lambda^{(k)} = \lambda = \text{const}$.

Теоретически окончание поиска определяется следующим условием: $X^{(r)}$ — точка экстремума, если $g(X^{(r)}) = 0$. На рис. 37 показаны траектории движения.

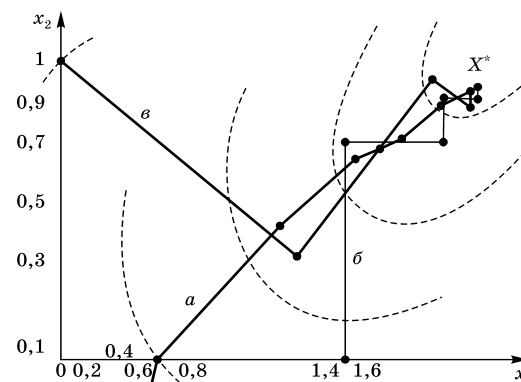


Рис. 37
Градиентные методы
оптимизации нелинейных функций:

a — классический градиентный метод; *б* — покоординатный метод; *в* — метод наискорейшего спуска.

Для практических вычислений правила остановки могут быть различны. Для простоты можно считать, что X^r является точкой экстремума, если

$$\max \left| \left(\frac{\partial F}{\partial X_j} \right)_{X^{r-1}} \right| \leq \varepsilon,$$

где ε — заданная малая величина.

Ясно, что этот метод можно применять в том случае, когда можно хотя бы численно вычислить градиент функции в точке. Заметим, что для гладкого оптимума при приближении к оптимальной точке величина градиента автоматически уменьшается, поэтому даже при постоянной величине λ длина шага уменьшается, что является желательным для любого поискового метода.

6.3.2. ПОКООРИНАТНЫЙ МЕТОД

В этом методе движение из начальной точки производится сначала вдоль первой координаты x_1 . Знак направления совпадает с направлением проекции градиента на эту координату, если $F(X) \rightarrow \max$, или противоположен ему, если $F(X) \rightarrow \min$. Конец шага определяется точкой, где $\frac{\partial F}{\partial X_1} = 0$. Затем движение производится по второй координате X_2 , затем по X_3 и т. д., и, наконец, по X_n . Затем повторяют движение опять по X_1 , X_2 и т. д. Можно использовать следующее правило остановки:

$$\max \left| \left(\frac{\partial F}{\partial X_j} \right)_{X^{r-1}} \right| \leq \varepsilon.$$

6.3.3. МЕТОД НАИСКОРЕЙШЕГО СПУСКА И ЕГО МОДИФИКАЦИИ

В этом методе из некоторой начальной точки движение осуществляется вдоль направления градиента до тех пор, пока производная по этому направлению не будет равна нулю. Далее из этой точки определяем новый градиент и т. д. Отличие здесь в том, что длина шага из точки X^k определяется из условия:

$$\min_{\lambda > 0} F[X^k \pm \lambda g(X^k)].$$

Эту вспомогательную задачу одномерной оптимизации можно решать на основе рассмотренных методов прямого поиска (дихотомии и т. д.)

Наискорейший и покоординатный методы называют *методами с длинным шагом*. Кроме этого, существуют методы, использующие вторую производную для определения длины шага, например метод Ньютона:

$$X^{k+1} = X^k + [F''(X^k)]^{-1} g(X^k),$$

где $[F''(X^k)]^{-1}$ — обратная матрица вторых производных в точке X^k .

6.4. МЕТОД ДЕФОРМИРУЕМОГО МНОГОГРАННИКА НЕЛДЕРА–МИДА

Современные методы поиска минимума (максимума) нелинейной функции чрезвычайно разнообразны. Одним из наиболее эффективных является метод Нелдера–Мида. Идея метода состоит в том, что для определения направления движения вычисляются значения целевой функции в вершинах сначала правильного, а затем деформируемого многогранника (рис. 38). Многогранник — некоторое тело в n -мерном пространстве. Правильный многогранник называется *симплексом*, в задачах для двух переменных — это правильный треугольник.

Вычисленные значения целевой функции в вершинах симплекса сравниваются, а затем выполняются операции:

1) отражения — поворот симплекса через одну из сторон;

2) растяжения — если идем в правильном направлении;

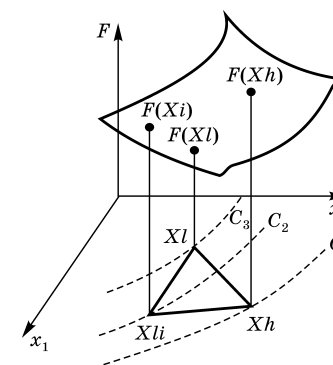


Рис. 38
Иллюстрация идеи метода Нелдера–Мида для случая двух переменных

3) редукции — возврат назад, если перескочили оптимум;

4) сжатия — уменьшение сторон, чтобы движение было с более мелким шагом.

В этом методе не используется понятие производной функции, что расширяет его возможности. Поиск оптимальной точки ведется путем поворота и деформации многогранника (симплекса) на основе анализа вершин в соответствии с приведенными операциями. Пусть k — номер итерации, тогда алгоритм Нелдера–Мида заключается в следующем:

1. Выбирается начальный симплекс (если в задаче 2 переменные $n = 2$, то симплекс — правильный треугольник). Его вершины $X_i^{(k)}$, $i = \overline{1, n+1}$. Обозначим самую худшую точку $X_h^{(k)}$, т. е. для

$$F(X) \rightarrow \min, F(X_h^{(k)}) = \max F(X_i^{(k)});$$

$X_l^{(k)}$ — лучшая точка, т. е. $F(X_l^{(k)}) = \min F(X_i^{(k)})$;

$X_0^{(k)}$ — центр тяжести всех вершин, исключая $X_h^{(k)}$, его координаты

$$X_0^{(k)} = \frac{1}{n} \left[\left(\sum_{i=1}^{n+1} X_{ij}^{(k)} \right) - X_{hj}^{(k)} \right]; \quad j = \overline{1, n}.$$

2. Осуществляют отражение (проектирование) $X_h^{(k)}$ через центр тяжести, новая точка получается так:

$$X_\alpha^{(k)} = X_0^{(k)} + \alpha(X_0^{(k)} - X_h^{(k)}),$$

где α — коэффициент отражения. Обычно $\alpha = 1$.

- Если $F(X_\alpha^{(k)}) \leq F(X_l^{(k)})$, то вектор $X_\alpha^{(k)} - X_0^{(k)}$ *растягивается* в γ раз и получаем $X_b^{(k)} = X_0^{(k)} + \gamma(X_\alpha^{(k)} - X_0^{(k)})$.
- Если для полученной точки $F(X_b^{(k)}) < F(X_l^{(k)})$, то $X_h^{(k)}$ заменяется на $X_b^{(k)}$, и переходим к пункту 2. В противном случае заменяем $X_h^{(k)}$ на $X_\alpha^{(k)}$ и переходим к пункту 2.
- Если $F(X_\alpha^{(k)}) > F(X_l^{(k)})$, для всех $i \neq h$, то вектор $X_h^{(k)} - X_0^{(k)}$ *сжимается*, получаем точку $X_c^{(k)} = X_0^{(k)} + \beta(X_h^{(k)} - X_0^{(k)})$, $0 < \beta < 1$. Точка $X_h^{(k)}$ заменяется на $X_c^{(k)}$, и переходим к выполнению пункта 2.

- Если $F(X_\alpha^{(k)}) > F(X_h^{(k)})$, то все векторы $X_i^{(k)} - X_l^{(k)}$ уменьшаются в λ раз (обычно $\lambda = 2$) — *редукция*

$$X_i^{(k)} = X_l^{(k)} + \frac{1}{\lambda}(X_i^{(k)} - X_l^{(k)}); \quad i = \overline{1, n+1}.$$

Возвращаемся к пункту 2. Правила остановки могут быть различными, например,

$$\left\{ \frac{1}{1+n} \sum_{i=1}^{n+1} [F(X_i^r) - F(X_0^r)]^2 \right\}^{\frac{1}{2}} \leq \delta;$$

для простоты можно использовать правило:

$$\begin{aligned} & i = \overline{1, n+1}, \\ \max_{j,i,p} |X_{i,j}^r - X_{p,j}^r| & \leq \varepsilon \quad j = \overline{1, n}, \\ & p = \overline{1, n+1}, \end{aligned}$$

решением будет точка $X_0^{(r)}$.

Например:

$$F(X) = -6x_1 + 2x_1^2 - 2x_1x_2 + 2x_2^2 \rightarrow \min,$$

выберем параметры:

$$\alpha = 1, \quad \beta = \frac{1}{2}, \quad \gamma = 2, \quad \lambda = 2, \quad \varepsilon = 0,1,$$

решение видно из рис. 39.

Подробнее см. в [16].

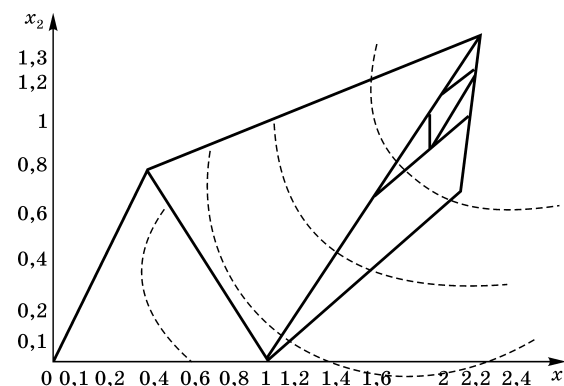


Рис. 39

Пример поиска оптимального решения по методу Нелдера–Мида

6.5. ЗАДАЧА НЛП С ОГРАНИЧЕНИЯМИ-РАВЕНСТВАМИ

В НЛП особо рассматриваются задачи с ограничениями-равенствами. Это так называемые задачи на условный экстремум:

$$F(X) \rightarrow \min, \\ \varphi_i(X) = b_i, \quad i = \overline{1, m}.$$

Решение такой задачи производится с использованием функции Лагранжа, позволяющей построить новую целевую функцию, которая бы учитывала ограничения:

$$\Phi(X, \lambda) = F(X) + \sum_{i=1}^m \lambda_i (\varphi_i(X) - b_i), \quad (6.3)$$

где $F(X)$ — целевая функция; λ_i — множители Лагранжа. Если все ограничения выполняются, то $\Phi(X) = F(X)$.

Таким образом, задача минимизации нелинейной целевой функции с ограничениями-равенствами сводится к минимизации функции Лагранжа без ограничений. Для ее решения используется классический прием — записываются необходимые условия существования экстремума:

$$\begin{cases} \frac{\partial \Phi}{\partial X_j} = \frac{\partial F}{\partial X_j} + \sum_{i=1}^m \left(\lambda_i \frac{\partial \varphi_i}{\partial X_j} \right) = 0, \quad j = \overline{1, n}; \\ \frac{\partial \Phi}{\partial \lambda_i} = \varphi_i(X) - b_i = 0, \quad i = \overline{1, m}. \end{cases} \quad (6.4)$$

После выполнения операций дифференцирования получаем в общем случае систему нелинейных уравнений относительно неизвестных X_j и λ_i — множителей Лагранжа. Решение системы (6.4) дает точки, подозрительные на экстремум, и их еще надо проверить с помощью достаточных признаков экстремума.

Рассмотрим некоторые из этих признаков.

1. Пусть X^* — точка подозрительная на экстремум, полученная с помощью выражений (6.4). Рассмотрим матрицу Гессе — матрицу H вторых производных в точке X^* :

$$\left(H = \frac{\partial^2 \Phi}{\partial X_i \partial X_j} \right)_{X^*}. \quad (6.5)$$

Если эта матрица определена положительно, то X^* — точка локального минимума, а если отрицательно, то локального максимума. Если же матрица Гессе знаконеопределена, то в этой точке экстремума нет, а если полуопределена, то этот признак не дает ответа на вопрос.

2. Пусть D_k — главный определитель матрицы Гессе k -го порядка:

а) если $D_1 > 0, D_2 > 0, D_3 > 0, \dots$ — это достаточный признак минимума;

б) если $D_1 < 0, D_2 > 0, D_3 < 0, \dots$ — это достаточный признак максимума (знаки строгих неравенств чередуются);

в) если же в определителях знаки \geq и \leq , то это необходимый, а не достаточный признак и вопрос об экстремуме не решается.

3. Наиболее «сильным» достаточным признаком является следующий [1]. Составим расширенную матрицу Гессе H^B , для чего используем матрицу производных от функций ограничений:

$$Q = \begin{pmatrix} \frac{\partial \varphi_1}{\partial X_1} & \frac{\partial \varphi_1}{\partial X_2} & \dots & \frac{\partial \varphi_1}{\partial X_n} \\ \frac{\partial \varphi_2}{\partial X_1} & \frac{\partial \varphi_2}{\partial X_2} & \dots & \frac{\partial \varphi_2}{\partial X_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varphi_m}{\partial X_1} & \frac{\partial \varphi_m}{\partial X_2} & \dots & \frac{\partial \varphi_m}{\partial X_n} \end{pmatrix}, \quad H^B = \begin{pmatrix} 0 & Q \\ Q' & H \end{pmatrix}, \quad (6.6)$$

где Q' — транспонированная матрица.

Матрица H^B имеет размерность $(m + n) \times (m + n)$.

Признак: точка X^* соответствует максимуму, если, начиная с главного определителя порядка $m + 1$, последние $n - m$ главных определителей матрицы H^B образуют знакопеременный ряд, а знак главного определителя, стоящего на $(n - m)$ -м месте от конца, совпадает со знаком $(-1)^{m+1}$, и минимуму, если начиная с главного определителя порядка $m + 1$, знак последующих $n - m$ главных определителей равен $(-1)^m$.

Например: $n = 3, m = 2, n - m = 1$, проверяем D_5 ; $n = 7, m = 4, n - m = 3$, проверяем D_9 .

Пример.

$$F = x_1 x_2 \rightarrow \max, x_1 + x_2 = 1.$$

Запишем функцию Лагранжа

$$\Phi(x_1, x_2, \lambda) = x_1 x_2 + \lambda(x_1 + x_2 - 1) \rightarrow \max.$$

Найдем решение, используя необходимые условия

$$\left. \begin{aligned} \frac{\partial \Phi}{\partial X_1} &= x_2 + \lambda = 0; \\ \frac{\partial \Phi}{\partial X_2} &= x_1 + \lambda = 0; \\ \frac{\partial \Phi}{\partial \lambda} &= x_1 + x_2 - 1 = 0. \end{aligned} \right\}$$

Получаем решение системы

$$x_1 = x_2;$$

$$x_1 = x_2 = \frac{1}{2}, \quad \lambda = -\frac{1}{2}.$$

Запишем матрицу H^B :

$$H^B = \begin{pmatrix} 0 & Q \\ Q' & H \end{pmatrix}; \quad H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix};$$

$$H^B = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}; \quad n=2; \quad m=1; \quad m+1=2; \quad n-m=1.$$

Начиная со второго определителя ($m+1=2$), знаки чередуются:

$$D_1 = 0, \quad D_2 < 0, \quad D_3 = 2 > 0, \\ \text{причем } (-1)^{m+1} = (-1)^{1+1} > 0.$$

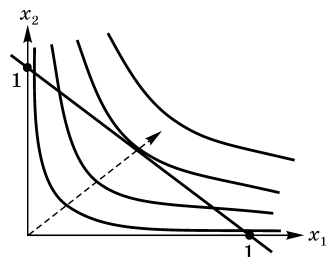


Рис. 40
Пример геометрического решения задачи на условный экстремум

Проверяем: $D_3 > 0$, значит в точке $x_1 = x_2 = 1/2$ находится максимум.

Геометрически задача на условный экстремум сводится к тому, что решение находится в той точке, где линия (поверхность), определяющая функцию (систему) ограничений, касается какой-нибудь линии уровня (рис. 40).

6.6. ВЫПУКЛОЕ НЛП

Теория нелинейного программирования разработана только для одного частного случая выпуклых целевых функций $F(X)$ и ограничений $\phi_i(X)$, и этот раздел соответственно называется выпуклым программированием.

Выпуклое множество обладает тем свойством, что отрезок, соединяющий любые две точки множества, также принадлежит этому множеству.

Таким образом, $C \subset E^n$ выпукло, если из $(X^1, X^2) \in C$ следует, что $\lambda = \theta X^1 + (1 - \theta)X^2 \in C$ для любого $0 \leq \theta \leq 1$.

Важно: если C_i — выпуклое множество, то $C = \bigcap C_i$ также выпукло, т. е. если каждое ограничение в задаче НЛП образует выпуклое множество, то и все ограничения дают выпуклое множество.

Дважды дифференцируемая функция $F(X)$ является выпуклой в том и только в том случае, если

$$\sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 F(X)}{\partial X_i \partial X_j} \Delta X_i \Delta X_j \geq 0$$

для любых $X \in M$, где M — выпуклое множество и ΔX_i и ΔX_j не обращаются в нуль одновременно.

Для проверки выпуклости $F(X)$ используется критерий Сильвестра: функция $F(X)$ — выпукла, если неотрицательны (для строго выпуклой положительны) все главные миноры матрицы Гессе:

$$\Delta_k = \begin{vmatrix} a_{11} & \dots & a_{1k} \\ \dots & \dots & \dots \\ a_{k1} & \dots & a_{kk} \end{vmatrix};$$

$$a_{ij} = \frac{\partial^2 F}{\partial X_i \partial X_j}; \quad k = 1, 2, \dots, n.$$

Если все $\Delta_k > 0$, то $F(X)$ — строго выпукла.

Функция $F(X)$ n переменных $X = (X_1, X_2, \dots, X_n)$, $X \in G$ называется выпуклой функцией в выпуклой области G , если для любых двух точек выполняется соотношение

$$F(\lambda X^{(1)} + (1 - \lambda)X^{(2)}) \leq \lambda F(X^{(1)}) + (1 - \lambda)F(X^{(2)}),$$

где $0 \leq \lambda \leq 1$.

Когда λ — переменная, то промежуточная точка пробегает значения от $X^{(1)}$ до $X^{(2)}$.

Функция строго выпукла, если знак \leq можно заменить на $<$.

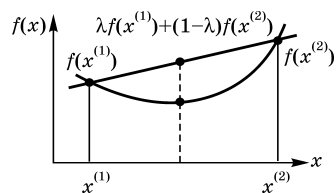


Рис. 41
Иллюстрация свойства
выпуклости функции

Таким образом, для задачи на \min выпуклая функция не может принимать больших значений, чем линейная (для задачи на \max меньших значений) (рис. 41).

Другим важным свойством функций, которое позволяет гарантировать решение

задачи НЛП, является так называемое условие Липшица.

Говорят, что функция $F(X)$ удовлетворяет на отрезке $[a, b]$ условию Липшица, если существует такое число $L > 0$ (константа Липшица), что

$$|F(X') - F(X'')| \leq L|X' - X''| \quad (6.7)$$

для всех $X', X'' \in [a, b]$.

Для проверки условия Липшица на практике используют следующий фактор: если функция $F(X)$ имеет на отрезке $[a, b]$ ограниченную производную, то она удовлетворяет условию (6.7), где

$$L \geq \max_{[a, b]} |F'(X)|.$$

6.7.

ТЕОРЕМА КУНА–ТАККЕРА ДЛЯ ВЫПУКЛОГО НЛП

Функция Лагранжа является основным инструментом для теоретического обоснования решения задачи НЛП. Если задача такова, что

$$\begin{aligned} F(X) &\rightarrow \min, \\ \varphi_i(X) &\leq 0, \quad i = \overline{1, m}, \\ X_j &\geq 0, \quad j = \overline{1, n}. \end{aligned} \quad (6.8)$$

и все функции обладают свойством выпуклости, то справедлива фундаментальная теорема Куна–Таккера, что решение (6.8) находится в так называемой седловой точке функции Лагранжа.

Функция $\Phi(X, \lambda)$ двух векторных переменных $X = (X_1 \dots X_n)$ и $\lambda = (\lambda_1 \dots \lambda_m)$ имеет в точке (X^*, λ^*) седловую точку, если выполняется соотношение $\Phi(X, \lambda^*) \geq \Phi(X^*, \lambda^*) \geq \Phi(X^*, \lambda)$ для всех X и λ из ε -окрестности (X^*, λ^*) .

Для простейшего случая одной переменной X и одного множителя Лагранжа седловая точка геометрически изображена на рис. 42 (будем считать, что $X \geq 0$ и $\lambda \geq 0$). В седловой точке максимум функции Лагранжа по λ совпадает с минимумом по X .

Необходимыми и достаточными условиями существования седловой точки функции двух векторных переменных являются соотношения

$$\begin{aligned} \frac{\partial \Phi}{\partial X_j} \Big|_{X^*, \lambda^*} &\geq 0, \quad \left(\frac{\partial \Phi}{\partial \lambda_i} \right)_{X^*, \lambda^*} \leq 0, \\ X_j^* \left(\frac{\partial \Phi}{\partial X_j} \right)_{X^*, \lambda^*} &= 0, \quad \lambda_i^* \left(\frac{\partial \Phi}{\partial \lambda_i} \right)_{X^*, \lambda^*} = 0, \\ X_j^* &\geq 0, \quad j = \overline{1, n}, \quad \lambda_i^* &\geq 0, \quad i = \overline{1, m}. \end{aligned} \quad (6.9)$$

Эти соотношения связаны с фундаментальной теоремой нелинейного программирования Куна–Таккера о том, что решение задачи НЛП находится в седловой точке функции Лагранжа $\Phi(X, \lambda)$.

Соотношения (6.9) являются не только необходимыми, но и достаточными, если кроме условий (6.9) выполняются условия регулярности Слейтера: в области допустимых решений существует такая точка X , в которой все ограничения неактивны, т. е. $j_i(X) < 0$ для всех $i = \overline{1, m}$ (в области есть «внутренние» точки).

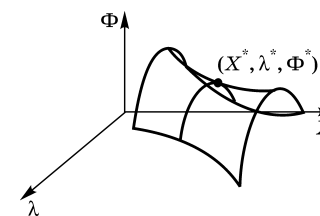


Рис. 42
Геометрическая
интерпретация понятия
седловой точки

Теорема Куна–Таккера. Если в задаче НЛП

$$F(X) \rightarrow \min; \quad \varphi_i(X) \leq 0, \quad i = \overline{1, m};$$

$$\Phi(X, \lambda) = F(X) + \sum_{i=1}^m \lambda_i \varphi_i(X)$$

все функции выпуклы, а множество допустимых решений, определяемое ограничениями, удовлетворяет условиям регулярности Слейтера, то для оптимальности точки X^* необходимо и достаточно выполнение условий (6.9). Доказательство теоремы Куна–Таккера можно найти в [6], [21].

Если все ограничения линейные, то условия Слейтера в теореме необязательны.

6.8.

КВАДРАТИЧНОЕ ПРОГРАММИРОВАНИЕ

Задачей квадратичного программирования (КП) называется такая задача НЛП, в которой целевая функция представляет собой сумму линейной и квадратичной формы (переменные не старше второй степени), а все ограничения линейные. Для решения таких задач используется симплекс-метод.

Теорема Куна–Таккера позволяет записать условие Куна–Таккера для задачи КП и найти седловую точку. Рассмотрим задачу КП в виде

$$F(X) = \sum_{j=1}^n p_j X_j + \sum_{k=1}^n \sum_{j=1}^n (C_{kj} X_k X_j) \rightarrow \min,$$

$$\varphi_i(X) = \sum_{j=1}^n a_{ij} X_j - b_i \leq 0, \quad i = \overline{1, m},$$

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

В матричном виде пусть p, X, b — векторы-столбцы:

$$F = p'X + X'CX \rightarrow \min, \quad AX \leq b, \quad X \geq 0,$$

где C — матрица квадратичной формы, которая должна быть симметрична и положительно полуопределена, это гарантирует выпуклость F .

Метод Баранкина–Дорфмана непосредственно основан на применении теоремы Куна–Таккера.

Запишем функцию Лагранжа в матричной форме для задачи КП:

$$\Phi(X, \lambda') = \underbrace{p'X + X'CX}_{F(X)} + \underbrace{\lambda'(AX - b)}_{\text{штраф}};$$

$$\frac{\partial \Phi}{\partial X} = V = p + 2CX + A'\lambda; \quad -\frac{\partial \Phi}{\partial \lambda} = Y = -AX + b.$$

Тогда условие Куна–Таккера можно записать в следующем виде:

$$\left. \begin{aligned} AX + V &= b; \\ 2CX - V + A'\lambda &= -p; \\ X \geq 0; \quad Y \geq 0; \quad V \geq 0; \quad \lambda \geq 0; \end{aligned} \right\} (*)$$

$$X'V + Y'\lambda = 0 \quad (**).$$

Неизвестными являются (X, Y, V, λ) ;

(*) — система линейных уравнений относительно неизвестных; решить ее, значит, найти допустимые решения задачи ЛП;

(**) — нелинейное условие.

Тогда можно переходить из одного допустимого решения к другому и проверять условие (**).

Введем векторы-столбцы $Z = (X, Y, V, \lambda)$ и $\tilde{Z} = (V, \lambda, X, Y)$, тогда можно записать

$$X'V + Y'\lambda = \frac{1}{2} Z' \tilde{Z}.$$

Система (*) также может быть представлена через векторы Z , тогда окончательно условия Куна–Таккера будут иметь следующий вид:

$$\begin{pmatrix} A & E & 0 & 0 \\ 2C & 0 & -E & A' \end{pmatrix} Z = \begin{pmatrix} b \\ -p \end{pmatrix}, \quad Z \geq 0, \quad (6.10)$$

$$T(Z) = Z' \tilde{Z} = 0. \quad (6.11)$$

В (6.10) E — единичная матрица.

Метод Баранкина–Дорфмана заключается в следующем: находится допустимый вектор Z , удовлетворяющий выражению (6.10), и проверяется условие (6.11). Далее выбирается новое базисное решение, причем так, чтобы

величина $T(Z)$ все время уменьшалась. Для этого используется модифицированная симплекс-таблица, в которой ведущий элемент находится из условия минимизации выпуклой функции $T(Z)$:

$$T(Z) = Z' \tilde{Z} \rightarrow \min, \quad (6.12)$$

т. е. мы решаем задачи (6.10) и (6.12) вместо (6.10) и (6.11). В симплекс-таблице записываются в качестве базисных переменных все переменные. Запишем строку базисных переменных:

$$Z_g = d_{g_0} + \sum_{h=1}^n d_{g_h} t_h, \quad g = \overline{1, 2N},$$

где $2N = 2(n + m)$, а t_h — свободные переменные.

Если строка соответствует свободным переменным, то в строке одна единица и остальные нули. Для выбора ведущего элемента используются дополнительные элементы, которые записываются в дополнительные строки:

$$\alpha_j = \sum_{i=1}^N (d_{ij} d_{i+N, 0} + d_{i+N, j} d_{i0}),$$

$$\beta_j = 2 \sum_{i=1}^N d_{ij} d_{i+N, j},$$

$$\Theta_j = \min_g \frac{d_{g_0}}{|d_{g_j}|} \quad \text{при } d_{g_j} < 0 \quad (\arg \min = r),$$

$$K_j = 2\alpha_j + \Theta_j \beta_j.$$

Можно показать, что новое значение T после замены j -й свободной переменной будет $T_j = T + \Theta_j K_j$. По определению $\Theta_j > 0$, поэтому, чтобы T уменьшалось, K_j должно быть меньше нуля. Рассмотрим величину K_j : величина β_j — вторая производная функции $T(Z)$, которая вогнута, и поэтому $\beta_j > 0$, значит, K_j может быть отрицательной; если $\alpha_j < 0$, значит, в качестве ведущего надо выбрать этот j -й столбец, строку надо выбрать ту, для которой вычисляется величина Θ .

Пример.

$$F(x) = -6x_1 + 2x_1^2 - 2x_1x_2 + 2x_2^2 \rightarrow \min;$$

$$\begin{aligned} x_1 + x_2 &\leq 2; \\ x_1 &\geq 0; \\ x_2 &\geq 0. \end{aligned}$$

Запишем все матрицы, которые нам нужны:

$$A = (1, 1); \quad b = 2; \quad C = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix};$$

$$p = \begin{pmatrix} -6 \\ 0 \end{pmatrix}; \quad n = 2; \quad m = 1; \quad N = 3.$$

Запишем условие Куна–Таккера, определяемое выражением (6.10):

$$\begin{pmatrix} \begin{pmatrix} 1 & 1 \\ 4 & -2 \\ -2 & 4 \end{pmatrix} & \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ y_1 \\ v_1 \\ v_2 \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ 0 \end{pmatrix}.$$

Перемножим матрицы, получим систему

$$\begin{aligned} x_1 + x_2 + y_1 &= 2, \\ 4x_1 - 2x_2 - v_1 + \lambda_1 &= 6, \\ -2x_1 + 4x_2 - v_2 + \lambda_1 &= 0. \end{aligned}$$

Чтобы записать симплекс-таблицу, надо выразить базисные переменные через свободные. Для получения первого базисного решения используется любой метод, например, метод Гаусса. Можно использовать метод искусственного базиса (см. главу 2), в котором применяется симплекс-процедура. Затем нужно получить опорное решение и только после этого записать модифицированную симплекс-таблицу. Для простых задач можно подобрать первое базисное решение так, чтобы сразу получить опорное решение (свободные члены > 0). Для нашего примера

$$\begin{aligned} y_1 &= 2 - x_1 - x_2, \\ \lambda_1 &= 6 - 4x_1 + 2x_2 + v_1, \\ v_2 &= 6 - 6x_1 + 6x_2 + v_1. \end{aligned}$$

Выражение для v_2 получено путем вычитания третьего уравнения из второго.

Таблица 38

	1	x_1	x_2	v_1
x_1		1		
x_2			1	
y_1	2	-1	-1	
v_1				1
v_2	6	-6	6	1
λ_1	6	-4	2	1
α_j	24	-14	4	2
β_j		8		
Θ_j		1		
K_j		-20		

Таблица 39

	1	v_2	x_2	v_1
x_1	1	$-\frac{1}{6}$	1	$\frac{1}{6}$
x_2			1	
y_1	1	$\frac{1}{6}$	-2	$-\frac{1}{6}$
v_1				1
v_2		1		
λ_1	2	$\frac{2}{3}$	-2	$\frac{1}{3}$
α_j	4	1	-6	1
β_j			8	
Θ_j			$\frac{1}{2}$	
K_j			-8	

Запишем симплекс-таблицы (табл. 38, 39) по методу Баранкина–Дорфмана.

Примечание. Алгоритм симплекс-преобразования аналогичен рассмотренному в главе 2, только: «...ведущую строку умножаем на $-\lambda$, а столбец на λ , отмечаем в строке нижние элементы, а в столбце верхние». Порядок строк в таблице нарушать нельзя.

В табл. 38 $T = \alpha_0 = 24$, а в табл. 39 равно 4. Чтобы не вычислять всю третью таблицу, подсчитаем α_0 для следующей таблицы по формуле

$$T = 4 + 1/2(-8) = 0.$$

Значит, следующая таблица оптимальная.

Вычислим только необходимые элементы (свободные члены), получим: $x_1 = 3/2$; $x_2 = 1/2$; $y_1 = v_1 = v_2 = \lambda_1 = 0$. Оптимальное значение $F = -11/2$.

Недостаток метода: иногда встречаются задачи, когда все $K_j > 0$. Значит, мы будем переходить в новую вершину, а значение T будет увеличиваться. В этом случае идем на временное ухудшение T (проходим через «мертвую зону»). В методе Франца–Вульфа это учтено [1], [7], [12], [16].

6.9. МЕТОДЫ ВОЗМОЖНЫХ НАПРАВЛЕНИЙ

Наиболее распространенным классом методов решения задач выпуклого НЛП являются градиентные методы, которые отличаются как способом получения последующей точки, так и способом «обработки» ограничений.

В методах возможных направлений переход от точки $X^{(k)}$ к точке $X^{(k+1)}$ осуществляется по направлению $S^{(k)}$, необязательно вдоль градиента. При этом движение вдоль $S^{(k)}$ должно быть таким, чтобы новая точка принадлежала области допустимых решений D . Направление, удовлетворяющее этому условию, называется *возможным*, или *допустимым*:

$$X^{(k+1)} = (X^{(k)} + \lambda S^{(k)}) \in D. \quad (6.13)$$

Таких направлений множество; среди возможных направлений выбирают то, для которого скалярное произведение градиента в точке $X^{(k)}$ и этого направления больше нуля (для задачи минимизации меньше нуля). Такое направление называется *подходящим*. Подходящих направлений в общем случае также множество. Г. Зойтендейк предложил выбирать то, которое максимально увеличивает значение целевой функции.

Рассмотрим задачу выпуклого НЛП:

$$F(X) \rightarrow \max(\min), \quad (6.14)$$

$$\varphi_i(X) \leq 0, \quad i = \overline{1, m}. \quad (6.15)$$

Обозначим область допустимых решений через D , пусть $X^{(0)}$ — начальная точка, а X^* — единственное решение задачи, $X^{(0)} \in D$, $X^{(0)} \notin X^*$.

Переход из точки $X^{(k)}$ в $X^{(k+1)}$ осуществляют по такому направлению $S^{(k)}$, чтобы при $\lambda > 0$ луч $X^{(k)} + \lambda S^{(k)}$ принадлежал D .

Обозначим $g_0(X)$ и $g_i(X)$ — градиенты функций $F(X)$ и $\varphi_i(X)$ в точке X соответственно. Обозначим $I = \{1, \dots, m\}$, а $I(X)$ — множество индексов i таких, что $\varphi_i(X) = 0$; $I(X)$ указывает номера ограничений, выполняющихся в точке X точно ($I(X)$ — номера активных ограничений).

Пусть S — некоторый вектор, $S \neq 0$, такой, что для всех $i \in I(x^{(0)})$ выполняется неравенство $(g_i(X^{(0)}), S) < 0$. Здесь и далее (X, Y) — скалярное произведение векторов X и Y .

Тогда можно определить $l > 0$, чтобы $X^0 + lS \in D$. Для этого сначала определим $\lambda' = \min_{i \in I/I(X^{(0)})} d_i$, где d_i — наименьший положительный корень уравнения $\varphi_i(X^{(0)} + d_i S) = 0$ (если нет решения, или $d_i < 0$, то $d_i = +\infty$). Затем определим

$$\lambda'' = \min_{i \in \{0\} \cup I(X^0)} C_i,$$

где C_i — корень уравнения $(g_i(X^{(0)} + C_i S), S) = 0$ (если нет решения, то $C_i = \infty$). Таким образом, λ' выбирается из условия невыхода из области D , а λ'' — из условия неперехода экстремальной точки, если она окажется внутри области. Тогда, если $\lambda = \min(\lambda', \lambda'')$, то $X^{(0)} + \lambda S \in D$. Такие направления S , удовлетворяющие условию (6.15), называют *возможными* (допустимыми) в точке $X^{(0)}$. Из возможных направлений можно выбрать такие, для которых, кроме этого, выполняется условие $(g_0(X^{(0)}), S) > 0$. Такие направления называют *подходящими*, так как при движении в этом направлении значения целевой функции улучшаются, не выходя из D (для задачи на минимум $(g_0(X^0), S) < 0$).

Далее находится направление S из подходящих направлений, которое обеспечивает максимально возможное увеличение целевой функции, т. е. на каждом шаге решается вспомогательная задача

$$(g_0(X^{(0)}), S) \rightarrow \max; \quad (6.16)$$

$$\varphi_i(X^{(k)} + S) \leq 0, \quad i \in I(X^{(0)}). \quad (6.17)$$

Для нормализации длины вектора S используют одно из следующих условий:

$$N_1: S'S \leq 1,$$

$$N_2: -1 \leq S_j \leq 1, \quad j = \overline{1, n},$$

$$N_3: \begin{cases} S_j \leq 1 & \text{для } g_j > 0, \\ S_j \geq -1 & \text{для } g_j < 0, \end{cases}$$

$$N_4: gS \leq 1,$$

$$N_5: \varphi_i(X^{(k)} + S) \leq 0, \quad i = \overline{1, m}.$$

Видно, что условие (6.17) уже содержится в N_5 . Точка X^* является решением, как только $(g_0(X^*), S) = 0$. Метод Зойтендейка можно применять для любых выпуклых функций $F(X)$ и $\varphi_i(X)$, но он может не дать решения за конечное число шагов. Для задачи квадратичного программирования можно с использованием принципа сопряженности обеспечить сходимость за конечное число шагов.

Рассмотрим этот случай. Если $X^{(k)}$ есть внутренняя точка, т. е. $\lambda^{(k-1)} = \lambda'' < \lambda'$, то новое искоемое $S^{(k)}$ должно удовлетворять, помимо условия нормировки, еще одному условию сопряженности $(S^{(k)})'CS^{(k-1)} = 0$, где C — положительно полуопределенная матрица квадратичной формы для задачи квадратичного программирования:

$$Q = p'X - X'CX \rightarrow \max, \quad (6.18)$$

$$AX \leq b. \quad (6.19)$$

Если для определения $S^{(k-1)}$ такое условие уже использовалось, то оно остается, так что все условия сопряженности имеют вид

$$(S^{(k)})'CS^{(h)} = 0, \quad \text{для } h = k-1, k-2, \dots, k-r.$$

Сопряженность последующего направления с предыдущим и обеспечивает конечность шагов при поиске решения для задачи квадратичного программирования. Это условие позволяет совместить траекторию с осями поверхности целевой функции.

Рассмотрим пример с нормирующими условиями N_5 для задачи квадратичного программирования (6.18), (6.19). Для нее

$$g_0(X^{(k)}) = p - 2CX^{(k)} = g^k.$$

Длина очередного шага определяется из соотношения

$$\lambda^k = \min(\lambda'', \lambda').$$

Величина λ'' определяется из условия попадания в экстремум, т. е.

$$(g_0(X^{(k)} + \lambda'' S^{(k)}), S^{(k)}) = 0.$$

Подставим выражение градиента, получим

$$\begin{aligned} S^{(k)'}[p - 2C(X^{(k)} + \lambda'' S^{(k)})] &= S^{(k)'}p - S^{(k)'}2CX^{(k)} - S^{(k)'}2C\lambda'' S^{(k)} = \\ &= S^{(k)'}g(X^{(k)}) - S^{(k)'}2C\lambda'' S^{(k)} = 0. \end{aligned}$$

Отсюда

$$\lambda'' = \frac{g^{k'} S^{(k)}}{2S^{(k)'} C S^{(k)}}.$$

Величина λ' для N_5 всегда равна 1.

На рис. 43 показан поиск решения для примера

$$\begin{aligned} F(X) &= 6x_1 - 2x_1^2 + 2x_1x_2 - 2x_2^2 \rightarrow \max, \\ x_1 + x_2 &\leq 2, \\ x_1 \geq 0, \quad x_2 &\geq 0. \end{aligned}$$

Для решения неквадратичных задач НЛП принцип сопряженности уже неприменим, и для построения алгоритма, сходящегося к решению, приходится применять процедуру изменения длины шага.

Рассмотрим такой алгоритм по методу возможных направлений Зойтендейка. При выборе подходящего направ-

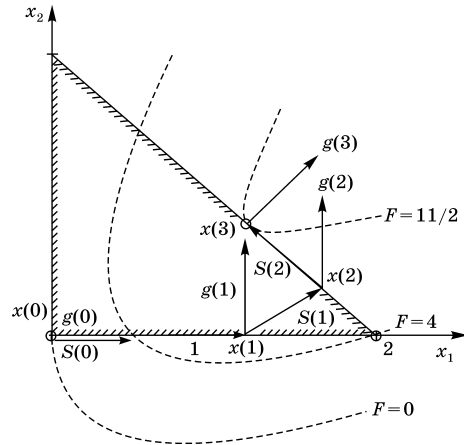


Рис. 43
Геометрическая иллюстрация решения задачи выпуклого НЛП

ления будем принимать во внимание не только те ограничения, которые в данной точке выполняются точно, но и те, которые выполняются «почти точно». В противном случае λ может измельчаться вдали от точки минимума. Для $\varepsilon > 0$ и $X \in D$ определим теперь ε — активное индексное множество $I_\varepsilon(X)$. Это те i , для которых $\varphi_i(X) + \varepsilon \geq 0$, а также $i = 0$, т. е. целевая функция $F(X)$. Для выбора λ и S в точке $X^{(k)}$ для получения следующей точки $X^{(k+1)} = X^{(k)} + \lambda S$ в соответствии с методом Зойтендейка необходимо выполнить условия невыхода из D , нормировки длины вектора λS и условия большего возрастания (убывания для задачи на минимум) целевой функции. В алгоритме, излагаемом ниже, последнее требование заменится требованием выбора направления, имеющего наибольший угол к касательным с ограничениями вблизи точки $X^{(k)}$, что дает возможность двигаться с большим шагом, не выходя за пределы D , а длина этого шага выбирается из условия наибольшего увеличения (уменьшения) $F(X)$. В качестве нормировки выберем N_2 . Тогда для задачи минимизации $F(X) \rightarrow \min$ для заданного ε в точке X необходимо определить величину $\Delta(\varepsilon, X)$ из условия

$$\Delta(\varepsilon, X) = \min_{S \in N_2} \max_{i \in I_\varepsilon(X)} (g_i(X), S),$$

где $N_2 = \{S_j | |S_j| \leq 1, j = \overline{1, n}\}$, $g_0(X)$ — градиент $F(X)$.

Легко показать, что для нахождения S необходимо решить следующую задачу ЛП (такой прием использовался в п. 4.2.3):

$$\left. \begin{aligned} \Delta &\rightarrow \min, \\ -\Delta + (g_i(X), S) &\leq 0, \quad i \in I_\varepsilon(X), \\ -1 \leq S_j \leq 1, \quad j &= \overline{1, n}. \end{aligned} \right\} \quad (6.20)$$

Решая ее, найдем Δ и S , а затем λ . Тогда новая точка $X^{(k+1)} = X^{(k)} + \lambda(X^{(k)})S(X^{(k)})$, где $\lambda(X^{(k)})$ — длина шага для точки $X^{(k)}$. Алгоритм для $F(X) \rightarrow \min$ следующий:

1. На нулевом шаге выбираем допустимое $X^{(0)}$ и скалярные величины $\varepsilon_0 > 0$, $\varepsilon' \in (0, \varepsilon_0)$, $\beta \in (0, 1)$. Обычно берут $\beta = \frac{1}{2}$.

2. На k -м шаге принимаем $\varepsilon = \varepsilon_k$; $X = X^{(k)}$.
3. Решая задачу (6.20), получаем $\Delta(\varepsilon_k, X^{(k)})$.
4. Если $\Delta(\varepsilon_k, X^{(k)}) \leq -\varepsilon_k$, принимаем $S(X^{(k)}) = S(\varepsilon_k, X^{(k)})$ и вычисляем длину шага $\lambda(X^{(k)})$, для чего сначала вычисляем $\gamma(X^{(k)}) = \max\{\gamma/\varphi_1(X^{(k)} + \alpha S(X^{(k)})) \leq 0\}$ для всех $\alpha \in [0, \gamma]$, $i = 1, m$.

Затем вычисляем $\lambda(X^{(k)}) \in [0, \gamma(X^{(k)})]$ — наименьшее (если их несколько) из этого интервала такое, что

$$F[X^{(k)} + \lambda(X^{(k)})S(X^{(k)})] = \min_{\lambda \in [0, \gamma(X^{(k)})]} F[X^{(k)} + \lambda \cdot S(X^{(k)})].$$

То есть выбираем λ такое, чтобы по лучу $S(X^{(k)})$ целевая функция уменьшилась (для задачи на min) быстрее.

Полагаем $X^{(k+1)} = X^{(k)} + \lambda(X^{(k)})S(X^{(k)})$; $\varepsilon_{k+1} = \varepsilon_k$ и переходим к п. 2.

4а. Если $\Delta(\varepsilon_k, X^{(k)}) > -\varepsilon_k$, то проверяем $\varepsilon_k \leq \varepsilon'$.

Если неравенство выполняется, то вычисляем $\Delta(0, X^{(k)})$ ($\varepsilon = 0$).

Если $\Delta(0, X^{(k)}) = 0$ — получено оптимальное решение $X^{(k)} = X^*$. В противном случае заменяем ε_k на $\beta\varepsilon_k$ и переходим к п. 2 со значением $\varepsilon = \beta\varepsilon_k$, $X = X^{(k)}$.

Пример. Не в ущерб общности, а для простоты вычислений рассмотрим задачу с линейными ограничениями и квадратичной $F(X)$:

$$\left. \begin{aligned} F(X) &= x_1^2 + x_2^2 \rightarrow \min, \\ \varphi_1(X) &= -2x_1 + 2 \leq 0, \\ \varphi_2(X) &= 4x_1 - 4x_2 \leq 0. \end{aligned} \right\}$$

Выберем $\varepsilon_0 = 1$; $\beta = \frac{1}{2}$; $X^{(0)} = (2, 2)$; $F(X^{(0)}) = 8$.

Первая итерация:

$$\begin{aligned} I_{\varepsilon_0}(X^{(0)}) &= \{0, 2\}, \\ g_0(X^{(0)}) &= (4, 4), \\ g_2(X^{(0)}) &= (4, -4). \end{aligned}$$

Получаем задачу

$$\left. \begin{aligned} \Delta &\rightarrow \min, \\ -\Delta + 4S_1 + 4S_2 &\leq 0, \\ -\Delta + 4S_1 - 4S_2 &\leq 0, \\ |S_1| &\leq 1; |S_2| \leq 1. \end{aligned} \right\}$$

Решение:

$$S_1 = -1; S_2 = 0; \Delta = -4; |\Delta| > \varepsilon_0, \Delta < -\varepsilon_0,$$

поэтому $\varepsilon_1 = \varepsilon_0 = 1$; $\gamma(X^{(0)}) = 1$; $\lambda(X^{(0)}) = 1$.

Получаем $X^{(1)} = X^{(0)} + \lambda(X^{(0)})S(X^{(0)}) = (1, 2)$; $F(X^{(1)}) = 5$ (было $F(X^{(0)}) = 8$), т. е. величина F уменьшилась.

Вторая итерация:

$$I_{\varepsilon_1}(X^{(1)}) = \{0, 1\}, g_0(X^{(1)}) = (2, 4), g_1(X^{(1)}) = (-2, 0).$$

$$\left. \begin{aligned} \Delta &\rightarrow \min, \\ -\Delta + 2S_1 + 4S_2 &\leq 0, \\ |S_1| &\leq 1; |S_2| \leq 1. \end{aligned} \right\}$$

Решение:

$$S_1 = 1; S_2 = -1; \Delta = -2; |\Delta| > \varepsilon_1,$$

поэтому $\varepsilon_2 = \varepsilon_1 = 1$; $\lambda(X^{(1)}) = 0,5$.

Получаем

$$X^{(2)} = X^{(1)} + \lambda(X^{(1)})S(X^{(1)}) = (1,5; 1,5); F(X^{(2)}) = 4,5.$$

Третья итерация:

$$\begin{aligned} I_{\varepsilon_2}(X^{(2)}) &= \{0, 1, 2\}, \\ g_0(X^{(2)}) &= (3, 3), g_1(X^{(2)}) = (-2, 0); g_2(X^{(2)}) = (4, -4). \end{aligned}$$

$$\left. \begin{aligned} \Delta &\rightarrow \min, \\ -\Delta + 3S_1 + 3S_2 &\leq 0, \\ \Delta - 2S_1 &\leq 0, \\ \Delta + 4S_1 - 4S_2 &\leq 0, \\ |S_1| &\leq 1, |S_2| \leq 1. \end{aligned} \right\}$$

Решение:

$$S_1 = 0; S_2 = 0; \Delta = 0; |\Delta| < \varepsilon_2,$$

поэтому заменяем $\varepsilon_2^{\text{нов}} = \beta\varepsilon_2 = 0,5$.

Получаем $I_{\varepsilon_2^{\text{нов}}}(X^{(2)}) = \{0, 2\}$. Снова решаем

$$\left. \begin{aligned} \Delta &\rightarrow \min, \\ -\Delta + 3S_1 + 3S_2 &\leq 0, \\ -\Delta + 4S_1 - 4S_2 &\leq 0, \\ |S_1| &\leq 1, |S_2| \leq 1. \end{aligned} \right\}$$

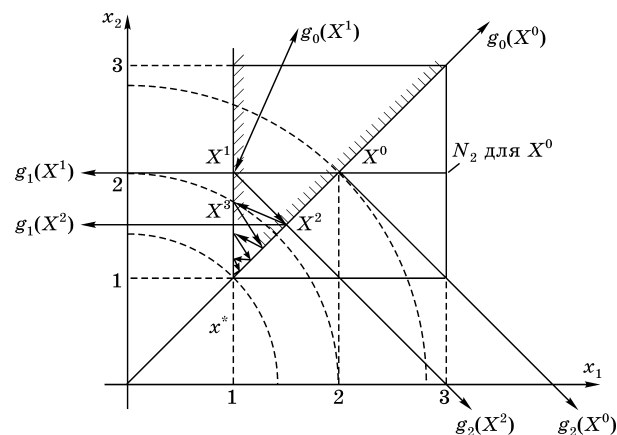


Рис. 44

Геометрическая иллюстрация примера решения задачи выпуклого НЛП методом Зойтендейка

Решение:

$$S_1 = -1; S_2 = \frac{1}{7}; \Delta = \frac{-24}{7}; |\Delta| > \varepsilon_2^{\text{нов}},$$

поэтому $\varepsilon_3 = 0,5$; $\lambda(X^{(2)}) = 0,5$.

Переходим к четвертой итерации. Продолжая вычисления согласно алгоритму, получим $X^* = (1, 1)$, $F(X^*) = 2$ (рис. 44).

Более подробные сведения об алгоритме решения и примеры можно найти в [7], [16].

6.10.

МЕТОД ПРОЕКЦИИ ГРАДИЕНТА

На каждой итерации этого метода предусмотрена процедура возврата очередного приближения градиентного спуска $X^{(k+1)} = X^{(k)} - \alpha_k f'(X^{(k)})$ на допустимое множество U , если $X^{(k+1)} \notin U$. Здесь обозначено $f(X)$ — целевая функция, $f'(X)$ — градиент целевой функции. Такой возврат

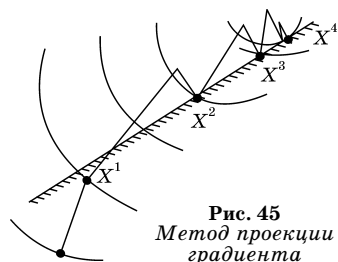


Рис. 45
Метод проекции градиента

производится посредством проектирования $X^{(k+1)}$ на U , т. е. замены $X^{(k+1)}$ на ближайшую точку множества $U \in E_n$ (рис. 45).

Определение. Пусть заданы замкнутое множество $U \subset E_n$ и точка $z \in E_n$. Точка $z_U = P_U(z)$ называется проекцией точки z на множество U , если

$$\rho(z_U, z) = \min_{X \in U} \rho(X, z),$$

где $\rho(X, y) = \|X - y\|$ — расстояние между точками X и y в пространстве E_n .

Очевидно, для точки $z \in U$ проекция $P_U(z)$ совпадает с z .

Таким образом, в методе проекции градиента последовательные приближения $X^{(k)}$ к точке минимума X^* целевой функции $f(X)$ на множестве U вычисляются по формулам

$$X^{(k+1)} = P_U[X^{(k)} - \alpha_k f'(X^{(k)})], \quad k = 0, 1, \dots, X^{(0)} \in U. \quad (6.21)$$

В зависимости от способа вычисления α_k из формулы (6.21) различают несколько вариантов метода проекции градиента, самыми распространенными из которых являются следующие:

1) α_k находится, как в методе наискорейшего спуска безусловной минимизации, т. е. $\Phi_k(\alpha_k) = \min_{\alpha > 0} \Phi_k(\alpha)$, где $\Phi_k(\alpha) = f[X^{(k)} - \alpha f'(X^{(k)})]$. В предположении, что градиент $f'(X)$ целевой функции удовлетворяет на множестве U условию Липшица, т. е.

$$\|f'(X') - f'(X'')\| \leq L \|X' - X''\|, \quad (6.22)$$

для всех $X', X'' \in U$ полагают $\alpha_k \equiv \alpha$, $k = 0, 1, \dots$, где α — произвольное число из интервала $(0; 2/L)$. Если известна минимальная константа Липшица L из (6.22), то выбирают, как правило, $\alpha = 1/L$.

Вычисления по формуле (6.21) завершаются при выполнении одного из неравенств $\|f'(X^{(k)})\| \leq \varepsilon$ или $\|(X^{(k)} - X^{(k-1)})\| \leq \varepsilon$, где $\varepsilon > 0$ определяет точность решения задачи. Окончательно полагают $X^* \approx X^{(k)}$, $f^* \approx f(X^{(k)})$.

Отметим, что определение проекции $P_U(z)$ для точки $z \notin U$ является самостоятельной задачей нелинейного программирования:

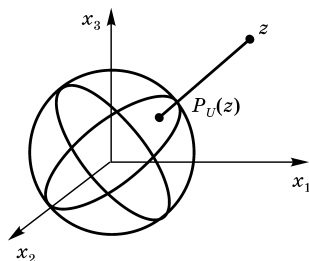


Рис. 46
Проекция точки на шар

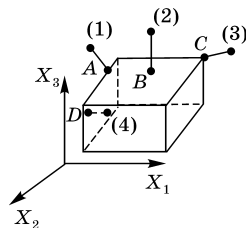


Рис. 47
Проекция точки на границу параллелепипеда

$$f(X) = \|X - z\|^2 = \sum_{j=1}^n (X_j - z_j)^2 \rightarrow \min, \quad X \in U, \quad (6.23)$$

решение которой может вызвать затруднения.

В частном случае, когда множество U определяется лишь линейными ограничениями, задача (6.23) представляет собой задачу квадратичного программирования. Ее решение может быть найдено за конечное число шагов, как описано выше.

Особый интерес при использовании метода проекции градиента представляют такие множества U , для которых задача проектирования (6.23) решается в явном виде (U — шар, параллелепипед и т. д.) (рис. 46, 47).

Иногда, как в примере с шаром, нахождение кратчайшего расстояния приводит к задаче на условный экстремум

$$\Phi(X, \lambda) = \sum_{j=1}^n (X_j - z_j)^2 + \lambda \sum_{j=1}^n (X_j^2 - R_0^2); \quad \frac{\partial \Phi}{\partial X_j} = 0; \quad \frac{\partial \Phi}{\partial \lambda} = 0.$$

В случае с шаром:

$$P_U(z) = \begin{cases} z, & \sum_{j=1}^n z_j^2 \leq R_0^2; \quad (z \in U); \\ \frac{R_0 z}{\sqrt{\sum_{j=1}^n z_j^2}}, & \sum_{j=1}^n z_j^2 > R_0^2; \quad (z \notin U). \end{cases}$$

Если граница области допустимых решений U — параллелепипед, то проекция точки определяется очень просто. Пусть область U определяется условиями

$$\begin{aligned} a_1 &\leq X_1 \leq b_1, \\ a_2 &\leq X_2 \leq b_2, \\ a_3 &\leq X_3 \leq b_3. \end{aligned}$$

Если только одна из координат $X_j \in [a_j, b_j]$, а остальные нет, то проекцией на параллелепипед будет точка на ребре (точка A).

Если две координаты X_j принадлежат области $X_j \in [a_j, b_j]$, $X_k \in [a_k, b_k]$, то проекцией на параллелепипед будет точка на грани (точка B).

Если все три координаты $X_1 \notin [a_1, b_1]$, $X_2 \notin [a_2, b_2]$, $X_3 \notin [a_3, b_3]$, то проекцией на параллелепипед будет точка в вершине (точка C). Если точка (4) внутри «запрещенного» параллелепипеда, то проекцией является точка D на ближайшей грани. Координаты проекций легко находятся по известным координатам точек и координатам вершин параллелепипеда.

6.11. МЕТОДЫ ШТРАФНЫХ И БАРЬЕРНЫХ ФУНКЦИЙ

Идея методов штрафных и барьерных функций заключается в том, что исходная задача НЛП с ограничениями заменяется последовательностью задач безусловной оптимизации (т. е. без ограничений) за счет введения новой функции, учитывающей невыполнение ограничений в виде «штрафа»: $C(X) = F(X) + P(X)$, где $F(X)$ — целевая функция, $P(X)$ — функция, равная в идеале нулю в области допустимых решений U и равная бесконечности вне этой области. Практически вместо $P(X)$ подбирается последовательность функций $P_k(X)$, зависящих от k так, чтобы $\lim_{k \rightarrow \infty} P_k(X) = P(X)$.

На рис. 48 показано решение задачи $F(x) \rightarrow \min, a < x < b$.

В методах штрафных функций последовательность точек стремится к решению в допустимой области извне и функции штрафа изменяются вне области U , поэтому

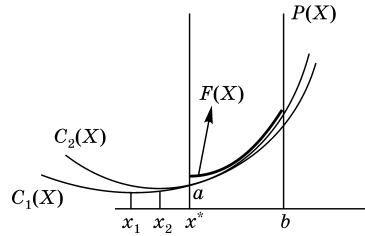


Рис. 48
Геометрическая иллюстрация
методов штрафных функций

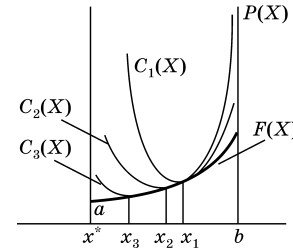


Рис. 49
Геометрическая иллюстрация
методов барьерных функций

они называются методами с внешней точкой (рис. 48). В методах барьерных функций функция $P(X)$ аппроксимируется барьерными функциями изнутри области (для них ограничения являются барьером), поэтому их называют методами с внутренней точкой (рис. 49).

Рассмотрим *метод штрафных функций* для задачи НЛП. $F(X) \rightarrow \min, X \in U$. Заменяем эту задачу последовательностью задач безусловной оптимизации

$$C_k(X) = F(X) + P_k(X) \rightarrow \min, \quad k = 1, 2, 3, \dots, \quad (6.24)$$

где $P_k(X)$ — функции, которые с ростом k во все большей степени учитывают ограничения, определяемые допустимым множеством U исходной задачи. В методе штрафных функций функции $P_k(X)$ подбираются так, чтобы при больших k функция $C_k(X)$ мало отличалась от $F(X)$ при $X \in U$ и быстро возрастала при удалении точки от допустимого множества U . Последовательностью штрафных функций множества U называется последовательность функций $\{P_k(X)\}$, обладающих свойством

$$\lim_{k \rightarrow \infty} P_k(X) = \begin{cases} 0, & \text{если } X \in U, \\ +\infty, & \text{если } X \notin U. \end{cases}$$

Рассмотрим один из вариантов метода штрафных функций приближенного решения задачи нелинейного программирования:

$$\begin{aligned} F(X) &\rightarrow \min, \\ g_i(X) &= 0, \quad i = 1, \dots, l, \\ g_i(X) &\leq 0, \quad i = l + 1, \dots, m, \end{aligned} \quad (6.25)$$

считая, что функции $F(X)$, $g_i(X)$, $i = 1, \dots, m$, заданы во всем пространстве E_n .

Положим

$$P_k(X) = kp(X), \quad k = 1, 2, 3, \dots, \quad (6.26)$$

где

$$p(X) = \sum_{i=1}^l g_i^2(X) + \sum_{i=l+1}^m [g_i^+(X)]^2,$$

$$g_i^+(X) = \begin{cases} 0, & \text{если } g_i(X) \leq 0, \\ g_i(X), & \text{если } g_i(X) > 0. \end{cases}$$

Равенство (6.26) определяет последовательность штрафных функций допустимого множества исходной задачи (проверьте!).

При определенных условиях последовательность решений задач безусловной минимизации (6.24), (6.26) сходится к решению X^* задачи (6.25), поэтому для достаточно больших k полагают $X^* = X^{(k)}$, $F^* = F(X^{(k)})$.

Критерием достижения требуемой точности решения задачи (6.25) может служить неравенство

$$\|X^{(k)} - X^{(k/2)}\| \leq \varepsilon,$$

где $\varepsilon > 0$ — число, характеризующее точность; k — четное число.

Если в задаче (6.25) $F(X)$ — выпуклая квадратичная функция, а $g_i(X)$, $i = 1, \dots, m$ — линейные функции, то точное решение вспомогательной задачи (6.24) можно найти из системы линейных уравнений $dF(X)/dX_j = 0$, $j = 1, \dots, n$, определяющих стационарную точку функции $F_k(X)$.

Пример.

Минимизировать $F(X) = -x_1x_2$ при ограничениях

$$g_1 = x_1 + x_2^2 - 1 \leq 0,$$

$$g_2 = -x_1 - x_2 \leq 0.$$

В качестве штрафной функции применим следующее:

$$p_i(g_i) = \left[\frac{(g_i + |g_i|)}{2} \right]^2, \quad p(X) = \sum_{i=1}^m p_i[g_i(X)] \quad \text{и} \quad P_k(X) = kp(X).$$

Т а б л и ц а 40

	k	$x_1(k)$	$x_2(k)$
k_1	1,0	0,89	0,64
k_2	2,0	0,77	0,62
k_3	3,0	0,73	0,61
k_4	10,0	0,67	0,58
ОПТИМУМ		$\frac{2}{3}$	$\frac{\sqrt{3}}{3}$

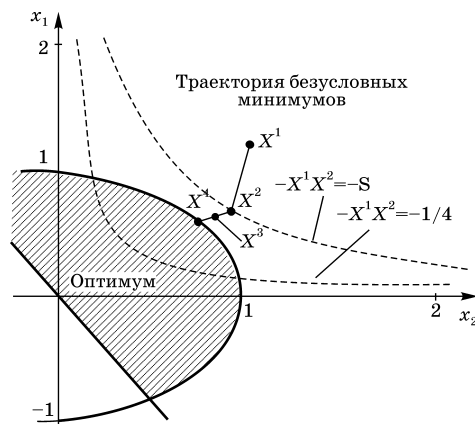


Рис. 50
Пример решения задачи НЛП методом штрафов

Тогда

$$C_k(X) = -x_1 x_2 + k \left[\frac{(x_1 + x_2^2 - 1 + |x_1 + x_2^2 - 1|)^2}{2} + \right. \\ \left. + k \left[\frac{(-x_1 - x_2 + |-x_1 - x_2|)^2}{2} \right] \right]$$

будет функцией, минимизируемой методом внешней точки.

В табл. 40 приведены значения $x_1(k)$ и $x_2(k)$ для четырех различных значений k . Из рис. 50 видно, что при возрастании k они приближаются к оптимуму из недопустимой области.

Методы внутренней и внешней точки существенно отличаются друг от друга, и не только тем, что в одном из них движение происходит в допустимой области, а в другом — в недопустимой.

Метод барьерных функций.

В методе барьерных функций исходная задача НЛП также сводится к последовательности задач безусловной минимизации (6.24), но функции $P_k(X)$ выбираются таким образом, чтобы при больших k функции $C_k(X)$ из фор-

мулы (6.24) мало отличались от $F(X)$ во внутренних точках X допустимого множества U .

В то же время при приближении точки $X \in U$ к границе множества U эти функции должны неограниченно возрастать.

Определение. Пусть множество $U \in E_n$ задано. Последовательность функций $\{P_k(X)\}$, определенных во всех внутренних точках множества U , называется последовательностью барьерных функций этого множества, если выполняются условия:

1. $\lim_{k \rightarrow \infty} P_k(X) = 0$ для любой фиксированной внутренней точки X множества U .
2. $\lim_{r \rightarrow \infty} P_k(X^{(r)}) = +\infty$ для любой последовательности $\{X^{(r)}\}$

внутренних точек множества U , сходящейся к какой-либо граничной точке этого множества.

Рассмотрим некоторые варианты метода барьерных функций решения следующей задачи нелинейного программирования:

$$F(X) \rightarrow \min; \\ g_i(X) \leq 0, \quad i = 1, \dots, m. \quad (6.27)$$

Положим

$$P_k(X) = \frac{1}{k} p(X), \quad k = 1, 2, \dots, \quad (6.28)$$

где

$$p(X) = \sum_{i=1}^m |g_i(X)|^{-k}, \quad k > 0 \quad (6.29)$$

или

$$p(X) = -\sum_{i=1}^m \ln[-g_i(X)]. \quad (6.30)$$

Выражения (6.28)–(6.30) определяют последовательности барьерных функций допустимого множества U задачи (6.27) (проверьте!).

Пусть $X^{(k)}$ — решение задачи безусловной минимизации (6.24), где функция $P_k(X)$ определена равенствами (6.28), (6.29) или (6.30). Полагая $X^* \approx X^{(k)}$, $F^* \approx F(X^{(k)})$ для достаточно большого k , находим приближенное решение задачи нелинейного программирования (6.25) методом

барьерных функций. Для контроля достигнутой точности решения можно использовать критерий, аналогичный рассмотренному.

Пример. Минимизировать $x_1 + x_2$ при ограничениях

$$g_1 = x_1^2 - x_2 \leq 0,$$

$$g_2 = -x_1 \leq 0.$$

В качестве функции $p(g_i)$ возьмем логарифмическую функцию. Обозначим $1/k = r$. Тогда $P_r(X) = r \cdot p(X)$,

$$p(X) = -\sum_{i=1}^m \ln(-g_i(X)).$$

$$C(X, r) = x_1 + x_2 - r \ln(-x_1^2 + x_2) - r \ln x_1.$$

Этот простой пример можно решать аналитически, учитывая, что $C(X, r)$ дважды дифференцируема. Необходимые условия первого порядка принимают вид

$$1 + \frac{r(x_1)}{-x_1^2 + x_2} - \frac{r}{x_1} = 0,$$

$$1 - \frac{r}{-x_1^2 + x_2} = 0.$$

Решая эту систему, получаем

$$x_1(r) = \frac{-1 \pm \sqrt{1+8r}}{4}.$$

Поскольку значение $x_1(r)$ должно быть положительно, будем рассматривать лишь один корень

$$x_1(r) = \frac{-1 + \sqrt{1+8r}}{4}.$$

Тогда

$$x_2(r) = \frac{-1 + \sqrt{1+8r}}{4} + r.$$

Значения $x_1(r)$ и $x_2(r)$ удовлетворяют достаточным условиям и поэтому дают локальный минимум задачи.

В табл. 41 приведены вычисленные значения $x_1(r)$, $x_2(r)$ для четырех различных r . В пределе при $r_k \rightarrow 0$ минимизирующие точки приближаются к решению $(0, 0)$.

В этой задаче при каждом значении r существует только один локальный минимум. Это происходит так пото-

Таблица 41

	r	$x_1(r)$	$x_2(r)$
r_1	1,000	0,500	1,250
r_2	0,500	0,309	0,595
r_3	0,250	0,183	0,283
r_4	0,100	0,085	0,107

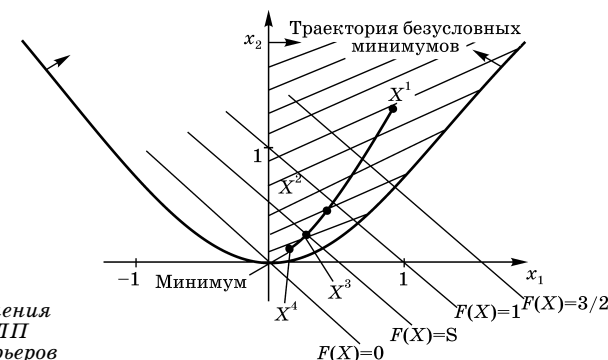


Рис. 51
Пример решения
задачи НЛП
методом барьеров

му, что исходная задача имела единственное решение. Оказывается, что в задачах со многими локальными минимумами (при выполнении вышеупомянутого условия регулярности Слейтера) существует последовательность безусловных локальных минимумов, сходящаяся к каждому из условных локальных минимумов [24], [1], [12], [22].

Геометрическая иллюстрация рассмотренного примера показана на рис. 51. Допустимая область заштрихована.

6.12. МЕТОД СКОЛЬЗЯЩЕГО ДОПУСКА

Метод скользящего допуска применяется при общей постановке задачи нелинейного программирования.

Минимизировать $F(X)$, $X \in E_n$ при ограничениях

$$\begin{aligned} h_i(X) &= 0, \quad i = 1, \dots, m, \\ g_i(X) &\geq 0, \quad i = m+1, \dots, p, \end{aligned}$$

где функции $F(X)$, $h_i(X)$, $g_i(X)$ могут быть как линейные, так и нелинейные. Алгоритм скользящего допуска позволяет улучшить значения целевой функции как за счет

информации, получаемой в допустимых точках пространства решения, так и за счет информации, которую удается получить при прохождении через точки, лежащие вне допустимой области, но близкие к допустимым. Интервалы, в пределах которых точки можно считать почти допустимыми, в ходе поиска постепенно сокращаются, так что в пределе (по мере приближения к искомому решению) учитываются только допустимые точки.

В качестве метода получения последовательности точек, улучшающих значение целевой функции, используется метод Нелдера–Мида (см. п. 6.4). При такой стратегии задачу можно заменить более простой, но имеющей то же самое решение задачей минимизации $F(X)$, $X \in E_n$ при ограничениях $\Phi^{(k)} - T(X) \geq 0$, где $\Phi^{(k)}$ — значение критерия скользящего допуска на k -м этапе; $T(X)$ — мера степени нарушения ограничений рассматриваемой задачи.

В качестве Φ выбирают положительно определенную убывающую функцию координат вершин деформируемого многогранника Нелдера–Мида.

Функция Φ служит критерием допуска для нарушения ограничений задачи и прекращения процедуры оптимизации.

Варианты выбора Φ многочисленны.

Например:

$$\Phi^{(k)} = \min \left\{ \Phi^{(k-1)}, \frac{m+1}{r+1} \sum_{i=1}^{r+1} \|X_i^{(k)} - X_{r+2}^{(k)}\| \right\},$$

$$\Phi^{(0)} = 2(m+1)t,$$

где t — размер исходного многогранника (например, его высота); m — число ограничений равенств; $X_i^{(k)}$ — вектор i -й вершины многогранника; $r = (n - m)$ — число степеней свободы целевой функции $F(X)$ в E_n ; $X_{r+2}^{(k)}$ — вектор, задающий «центр тяжести» многогранника при $r = n$.

По мере приближения к оптимальной точке $\Phi^{(k)}$ устремляется к нулю «сверху».

В пределе многогранник деформируется в точку

$$\lim_{x \rightarrow x^*} \Phi^{(k)} = 0.$$

Функционал

$$T(X) = \left[\sum_{i=1}^m h_i^2(X) + \sum_{i=m+1}^p U_i g_i^2(X) \right]^{1/2},$$

где $U_i = 0$ при $g_i(X) \geq 0$ и $U_i = 1$ при $g_i(X) < 0$,

$$T(X) \geq 0 \quad \forall X \in E^n.$$

Если $T(X^{(k)}) = 0$, то $X^{(k)}$ — допустима; если $T(X^{(k)}) > 0$, то $X^{(k)}$ — недопустима при $T(X^{(k)}) > \Phi^{(k)}$ и $X^{(k)}$ почти допустима при $0 \leq T(X^{(k)}) \leq \Phi^{(k)}$.

Это условие и определяет набор точек по допустимости.

Общая схема алгоритма выглядит так: при заданном $\Phi^{(k)}$ в точке $X^{(k+1)}$ может быть два варианта:

1) $T(X^{(k+1)}) \leq \Phi^{(k)}$ — перемещение считается разрешенным;

2) $T(X^{(k+1)}) > \Phi^{(k)}$ — необходимо найти другую точку $X^{(k+1)}$, ближе к допустимой области, чтобы $T(X^{(k+1)})$ уменьшилась. Для этого можно в алгоритме Нелдера–Мида считать точку $X^{(k+1)}$ худшей или же возвращать ее на ближайшую границу.

$T(X)$ не влияет на сходимость алгоритма, так как на заключительных этапах поиска точки $X_i^{(k)}$ являются внутренними, т. е. $T(X_i^{(k)}) \geq 0$.

Если X^* является граничной точкой, то поиск вдоль границы происходит за счет улучшения $F(X)$ при условии $\Phi^{(k)} - T(X_i^{(k)}) \geq 0$.

Степень нарушения ограничений по мере приближения к искомому решению уменьшается.

К моменту прекращения оптимизационного поиска выполняются условия

$$\Phi^{(k)} \leq \varepsilon,$$

$$F(X_i^{(k)}) \leq F(X^* \pm \varepsilon),$$

$$\varepsilon - T(X_i^{(k)}) \geq 0.$$

ГЛАВА 7

**ОСОБЕННОСТИ
СОВРЕМЕННОЙ ТЕОРИИ
ПРИНЯТИЯ
ОПТИМАЛЬНЫХ РЕШЕНИЙ**

Рассмотренные в предыдущих разделах методы поиска оптимальных решений составляют содержание математических методов оптимизации и математического программирования и в классическом понимании связаны с поиском экстремумов функций и функционалов при функциональных ограничениях на переменные. Решение здесь — математический объект. Причем вопросы оценки многих критериев и изучение того, почему такой-то критерий можно считать приемлемым, не изучаются (о критериях не спорят!).

На практике выбор решения намного сложнее.

1. В некоторых задачах сами варианты проведения операций представляют собой небольшое число отдельных альтернатив, не всегда однородных. Например: вариант 1 — строить большой завод, вариант 2 — строить небольшой завод, а потом расширять производство. Поэтому в теории принятия решений (ТПР) рассматривается как самостоятельный вопрос задания множества альтернатив, на котором ищется оптимальное решение.

2. Во многих задачах множество альтернатив не ясно с самого начала, например, альтернативы при игре в шахматы. Поэтому в ТПР говорят о генерации альтернатив в процессе решения задачи. При игре в шахматы приходится генерировать альтернативы в процессе игры и оценивать их.

3. Оценка ценности каждой альтернативы во многих случаях не сводится к простому сравнению количественного параметра (сравнению чисел). В связи с этим в ТПР

применяют специальные методы измерения полезности альтернатив.

4. В большом числе практических задач системы или операции не могут быть оценены единственным показателем эффективности. Сами системы являются очень часто многоцелевыми. Вот почему критерий не может быть скалярной величиной, а является вектором. Оказывается, что, когда по одному показателю один вариант лучше, по другому он может оказаться хуже, и поэтому возникает проблема их сравнения. Этот раздел современной ТПР называют многокритериальной, или векторной оптимизацией.

5. Во многих задачах, особенно в задачах автоматизации управления, присутствуют так называемые нечеткие переменные и нечеткие критерии. Появляется специальный раздел ТПР — принятия решений в расплывчатых ситуациях, в котором рассматриваются вопросы «человекоподобного» восприятия информации для принятия решения и создаются автоматизированные экспертные системы поддержки принятия решений.

6. Иногда встречаются задачи, в которых присутствует несколько сторон, преследующих разные интересы (боевые действия, поведение государств, конкурирующие фирмы), которые принимают решения в одной и той же системе. Такие ситуации называются конфликтными. Модель для принятия решений и принцип оптимальности в этих случаях являются совершенно иными, чем в рассмотренных задачах математического программирования. Этими вопросами занимается теория игр.

7. На практике одни решения приводят к необходимости принимать следующие решения и т. д., т. е. приходится изучать многоэтапные решения, в том числе и в условиях случайных возмущений.

8. При принятии решений в условиях неопределенности можно поставить и решить задачу устранения части неопределенности за счет проведения «эксперимента». Например, это выражается в виде разведки (военной или экономической), метеонаблюдений, радиолокации и т. п.

9. Во многих задачах принимаются так называемые групповые решения, при этом альтернативы оцениваются

группой экспертов — специалистов в данной предметной области. На основе такой экспертизы путем специальной обработки экспертных данных можно принимать решения в случаях, когда альтернативы трудно формализуемы. Такой метод был назван *методом экспертных оценок*.

Вот почему современная теория принятия решений представляет собой обобщение всей методологии получения решений в самых общих и практически важных случаях (много критериев, неопределенность принципа оптимальности, нечеткость шкалы ценности, условия риска, неповторимость ситуации, сложная логика предпочтений между альтернативами и др.).

Сказанное характерно для анализа оптимальных решений при построении современных автоматизированных систем управления и экспертных систем.

7.1. ОБЩАЯ ПОСТАНОВКА ЗАДАЧИ ПРИНЯТИЯ РЕШЕНИЯ

Задачей принятия решения назовем пару (Ω, P) , где Ω — множество вариантов, P — принцип оптимальности. Решением задачи (Ω, P) является множество $\Omega_p \subseteq \Omega$, полученное с помощью принципа оптимальности P . Отсутствие хотя бы одного из элементов лишает задачу смысла в целом. Если нет Ω , не из чего выделять Ω_p . Если нет P , то найти Ω_p невозможно. Математическим выражением принципа оптимальности P служит функция выбора C_p . Она сопоставляет любому подмножеству $X \subseteq \Omega$ его часть $C_p(X)$. Решением Ω_p исходной задачи и является множество $C_p(X)$.

Задачи принятия решений различаются в зависимости от имеющейся информации о множестве Ω и принципе оптимальности P . В общей задаче принятия решений как Ω , так и P , могут быть неизвестными.

Информацию, необходимую для выделения Ω_p , получают в процессе решения. Задачу с известным Ω называют задачей выбора, а задачу с известными Ω и P — общей задачей оптимизации.

Таким образом, задача выбора и задача оптимизации являются частными случаями общей задачи принятия решений.

В общем случае для задачи выбора необязательно необходимо полное восстановление принципа оптимальности, а можно ограничиться только информацией, достаточной для выделения Ω_p . Общая задача оптимизации не обязательно требует минимизации одной или нескольких числовых функций, может быть и так, что другим способом выделяется множество лучших элементов, т. е. выделяются значения $C_p(\Omega)$ при заданных Ω и C_p . Если C_p — скалярная функция на множестве Ω , то получаем обычную оптимизационную задачу.

Элементы множества Ω называются альтернативами, или вариантами. Принцип оптимальности P задает понятие лучших альтернатив: лучшими считают альтернативы, принадлежащие $C_p(\Omega)$. В практических задачах альтернативы обладают многими свойствами, оказывающими влияние на решение.

Пусть некоторое свойство альтернатив из Ω выражается числом, т. е. существует отображение $\varphi: \Omega \rightarrow E_1$. Тогда такое свойство называется *критерием*, а число $\varphi(x)$ — оценкой альтернативы x по критерию. Одновременный учет отдельных свойств может быть затруднительным. При этом можно выделить группы свойств, которые агрегируют в виде аспектов. *Аспект* — сложное свойство альтернатив, которое одновременно учитывает все свойства входящих в группу. В частном случае аспект может являться критерием.

Пусть все свойства k_1, \dots, k_m , учитываемые при решении задачи (Ω, P) , являются критериями. Критериальным пространством называют пространство E_m , координаты которого есть оценки соответствующих критериев.

Пример. При определении маршрута перевозок альтернативы — маршруты. Диспетчер учитывает свойства: протяженность, загрузку, энергоемкость, безопасность, техническое обслуживание и др.

Техническое обслуживание — это количество станций, сроки выполнения ремонтных работ и т. д. Следовательно, техническое обслуживание — это аспект.

Стоимость маршрута состоит из стоимости топлива, обслуживания и т. д., т. е. это тоже аспект. Однако так как стоимость можно вычислить, то это критерий (но так будет не всегда). Сформулируем схему процесса принятия решения.

Формируется множество Ω , его подготавливают для последующего решения задачи выбора (при назначении на должность сначала готовят список кандидатов, а затем назначают лицо из этого списка). Для формирования Ω используют условия возможности и допустимости альтернатив, которые определяют конкретными ограничениями задачи. При этом считают известным универсальное множество альтернатив Ω_y . Таким образом, задача формирования Ω является задачей выбора (Ω_y, P_1), где P_1 — принцип оптимальности, выражающий условия допустимости.

Множество $\Omega = C_{P_1}(\Omega_y)$, полученное в результате решения, называют исходным множеством альтернатив (ИМА) (Ω_y — все специалисты, Ω — специалисты, удовлетворяющие требованиям).

Таким образом, общая задача принятия решения сводится к решению двух последовательных задач выбора.

В процессе решения участвуют лицо, принимающее решение (ЛПР), эксперты, консультанты.

ЛПР — человек, имеющий цель, которая служит мотивом постановки задачи. ЛПР имеет полномочия и несет ответственность. Основная функция ЛПР — выделять Ω_P . ЛПР дает информацию о принципе оптимальности P .

Эксперт не несет ответственности, дает оценки альтернатив, необходимые для формирования ИМА и решения задачи выбора.

Консультант — специалист по теории выбора и ПР. Он разрабатывает модель, процедуру принятия решений, организует работу ЛПР и экспертов.

В простейшем случае задачу (Ω, P) решает непосредственно ЛПР.

Прикладные результаты ТПР имеют вид алгоритмов решения задач. Часть алгоритмов может быть реализована вручную, но основная часть ориентирована на диалоговый режим работы ЭВМ или на применение в экспертных системах.

7.2. КЛАССИФИКАЦИЯ ЗАДАЧ ПРИНЯТИЯ РЕШЕНИЙ

Для классификации задач принятия решений представим более подробно задачу ПР в виде семерки:

$$\text{ПР} = (Z, \Omega, K, S, F, G, P);$$

Z — описание предметной области, в которой принимается решение;

Ω — множество альтернатив;

K — множество критериев оценки альтернатив;

S — множество шкал измерения по критериям;

F — отображение множества альтернатив в множество критериальных оценок $F(X) \rightarrow K$;

G — система предпочтений;

P — решающее правило, отображающее систему предпочтений.

Любой из элементов этой семерки может служить классификационным признаком задач ПР. Однако в основном для классификации используются тип отображения F , мощность множества K , тип системы предпочтений G и характеристики области допустимых решений $X \subseteq \Omega$.

Отображение X в множество критериальных оценок может быть детерминированным (рис. 52), вероятностным (рис. 53) либо неопределенным (см. рис. 54). Поэтому задачи принятия решений можно разделить на ПР в условиях определенности, риска и неопределенности.

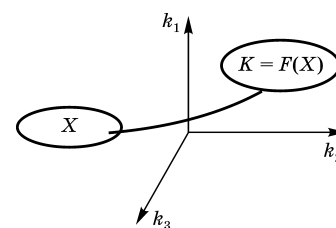


Рис. 52
Схема соответствия решения
и критерия в условиях
определенности

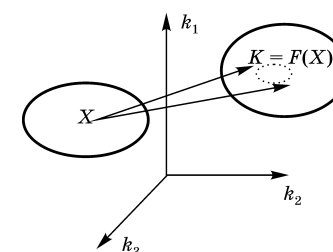


Рис. 53
Схема соответствия решения
и критерия в условиях
риска

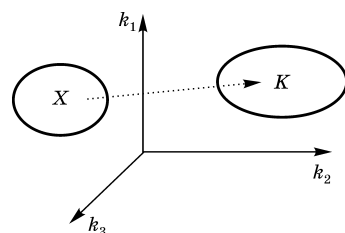


Рис. 54
Схема соответствия решения и критерия в условиях неопределенности

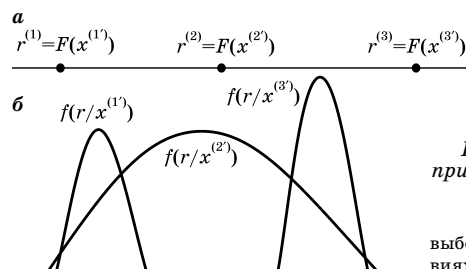


Рис. 55
Иллюстрация проблемы принятия решения в условиях риска при скалярном критерии:
выбор при одном критерии в условиях определенности (а) и риска (б).

ПР в условиях *определенности* предполагает детерминированное (взаимно однозначное) соответствие между X и K ;

При ПР в условиях *риска* отображение F зависит не только от X , но и от случайных факторов, поэтому F соответствует вероятностному отображению X в K ;

ПР в условиях *неопределенности* называют ситуацией, когда в системе присутствуют неопределенные факторы, существенно влияющие на результаты решения (например, действия противника, неизвестные обстоятельства и т. п.). Здесь не предполагается соответствия X и K .

Если K — скаляр, то это однокритериальная задача ПР. Если K — вектор, то это задача многокритериальной (векторной) оптимизации.

Отметим, что задача ПР со скалярным критерием является однозначно сформулированной задачей оптимизации только в случае детерминированного отображения F . Как только решающий элемент имеет дело с ПР при вероятностном характере отображения F либо априори неизвестном (неопределенном) F , то задача ПР даже со скалярным критерием приводит к различным трактовкам

принципа оптимальности (рис. 55). Действительно, в случае *а* сравнение вариантов $x^{(i)}$ производится по величине r , а в случае *б* для сравнения этих вариантов можно руководствоваться различными параметрами вероятностных распределений.

7.3. МНОГОКРИТЕРИАЛЬНАЯ ОПТИМИЗАЦИЯ

В главах 1–6 мы рассмотрели задачи, в которых требуется выбрать решение, доставляющее максимум (или минимум) единственного показателя эффективности (критерия) K . На практике часто возникает ситуация, когда эффективность операции приходится оценивать не по одному, а сразу по нескольким показателям k_1, k_2, \dots, k_L .

Примеры.

1. Оценка деятельности завода: прибыль, средняя зарплата, объем производственных фондов.
2. Оценка учебы студента: оценки по предметам.
3. Военная операция: потери, вероятность выполнения задачи.

Одни из этих показателей необходимо сделать больше, другие меньше. Как правило, эффективность больших по объему сложных операций, а также сложных многоцелевых систем не может быть исчерпывающим образом охарактеризована одним показателем: на помощь ему приходится привлекать и другие, дополнительные показатели.

Главной особенностью этой ситуации является то, что требования ко всем показателям в реальных системах несовместимы или противоречивы. Как правило, требование $\max k_1$ не обращает ни в максимум, ни в минимум другие показатели k_2, k_3, \dots . Поэтому широко распространенная формулировка «достижение максимального эффекта при минимальных затратах» некорректна. Корректными являются следующие формулировки:

1. Достижение максимального эффекта при заданных затратах.
2. Достижение заданного эффекта при минимальных затратах.

Для сравнения значений векторов $K = (k_1, k_2, \dots, k_L)$ иногда удобно предварительно привести все показатели k_1, k_2, \dots, k_L к стандартному виду, чтобы все критерии минимизировались и чтобы они имели безразмерный масштаб измерения. Векторный критерий K считается стандартным, если он удовлетворяет условию $k_i \geq 0$, и чем меньше k_i , тем лучше операция (система). Таким образом, идеальной является система, у которой $k_i = 0$. Нестандартный показатель можно привести к стандартному.

Если $k_i \rightarrow \max$, то $k_i^{\text{ст}} = k_i^{\max} - k_i$.

Если $k_i^{\max} = \infty$, то $k_i^{\text{ст}} = \frac{1}{k_i}$.

Стандартный критерий можно задать в виде отношения

$$k_i^{\text{ст}} = \left(\frac{1 - k_i}{k_i^{\text{и}}} \right),$$

где $k_i^{\text{и}}$ — идеальное значение критерия (либо определенное значение, либо $\max k_i$).

При этом в L -мерном пространстве задается вектор K , каждая компонента которого изменяется от 0 до 1. Полностью идеальной системе соответствует $K = 0$.

Прежде всего анализ векторов, соответствующих альтернативам из области допустимых альтернатив, позволяет заранее отбросить явно нерациональные варианты решений, уступающие лучшим вариантам по всем показателям. Этот этап векторной оптимизации называется *безусловной оптимизацией*.

Пусть анализируется боевая операция, оцениваемая по двум показателям:

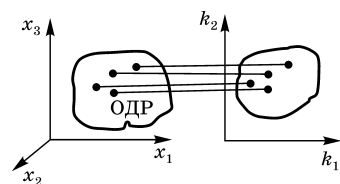


Рис. 56
Отображение вариантов
решения в пространство
критериев

k_1 — вероятность выполнения боевой задачи;

k_2 — стоимость израсходованных средств.

Очевидно, первый из показателей желательно обратить в максимум, а второй — в минимум. Пусть предлагается на выбор конечное число различных вариантов решения, обо-

значим их X_1, X_2, \dots, X_n . Для каждого известны значения k_1, k_2 . Изобразим каждый вариант решения в виде точки на плоскости с координатами k_1, k_2 . Когда X пробегают ОДР, точки k_1, k_2 заполняют критериальное пространство решений (рис. 56).

Итак, если некоторая операция оценивается несколькими критериями, каждый из которых число, то такая задача называется *многокритериальной*, или задачей векторной оптимизации. Для стандартных k имеет вид

$$K = (k_1, k_2, \dots, k_L) = \bar{f}(X, A) \rightarrow \min, \quad (7.1)$$

$$\varphi_i(X, A) \leq 0, \quad i = \overline{1, m}. \quad (7.2)$$

Существует принципиальная трудность в объективной (безусловной) оценке альтернатив при двух или более критериях. Она связана с проблемой сравнения двух векторов.

Безусловным критерием предпочтения называют критерий, основанный на сравнении компонент вектора. Два вектора критериев k^A и k^B безусловно сравнимы, если для любой l -й компоненты вектора выполняются неравенства

$$k_l^A \geq k_l^B, \quad l = \overline{1, L}; \quad \exists l, k_l^A > k_l^B.$$

Если все $k_l \rightarrow \min$, то альтернатива B безусловно предпочтительнее (лучше) A . Это означает, что A никак не может быть оптимальной и поэтому должна быть отброшена. Если же знаки неравенств для различных компонент вектора различны, то такие альтернативы *безусловно не-сравнимы*.

Пример. Необходимо выбрать лучших по успеваемости студентов А, Б, В, Г, Д. Критериями служат оценки по предметам М, N, О, Р (табл. 42).

Таблица 42

	М	N	О	Р
А	4	4	5	3
Б	5	5	4	3
В	4	5	3	3
Г	5	4	5	4
Д	4	5	4	3

Проведем безусловное сравнение векторов оценок. Студент А не сравним со студентом Б; В безусловно хуже Б — отбросим В; Д безусловно хуже Б — отбросим Д; А безусловно хуже Г — отбросим А. Остались две альтернативы Б и Г, которые безусловно несравнимы.

Определение. Множество безусловно несравнимых альтернатив, оставшихся после отбрасывания всех безусловно худших альтернатив, называется *множеством Парето*. Парето-оптимальное множество еще называют областью компромиссов. Ясно, что множество Парето можно получить в результате анализа критериального пространства.

Таким образом, векторная оптимизация включает два этапа:

1. Безусловная оптимизация. Здесь анализируется критериальное пространство, отсеиваются безусловно худшие варианты и получается множество Парето.

2. Условная оптимизация. Так как множество Парето, как правило, состоит из более чем одной точки, то для получения единственного решения применяются дополнительные принципы оптимальности (условия согласования критериев).

7.4.

ОПРЕДЕЛЕНИЕ МНОЖЕСТВА ПАРЕТО

Если целевая функция — векторная функция векторного аргумента и каждому малому изменению переменных X соответствует малое изменение каждого критерия, то критериальное пространство — это область с внутренними

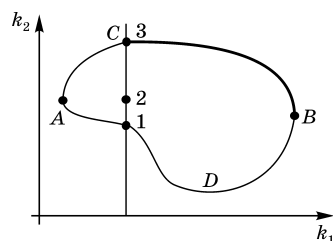


Рис. 57
Парето-оптимальное
множество

точками и границей (не обязательно замкнутая). Пусть есть только два критерия (k_1 , k_2) и $k_1 \rightarrow \max$; $k_2 \rightarrow \max$.

Рассмотрим область возможных значений критериев (рис. 57). Ясно, что точки, лежащие на одной вертикали или на одной горизонтали, всегда, безусловно, сравни-

мы (точки 1, 2, 3). Получается, что все внутренние точки можно отсеять. Дуга АВ состоит из лучших точек по критерию k_2 при постоянном k_1 .

Но часть точек дуги АВ, а именно точки дуги АС можно отбросить, так как они хуже точек СВ. Поэтому точками множества Парето будут точки, находящиеся на дуге СВ. При анализе характерными являются точки касания вертикалей и горизонталей к области критериев (точки А, В, С, D).

Существует несколько алгоритмов нахождения множества Парето. Они зависят от того, в каком виде задано множество вариантов критериев. В практических задачах сначала приходится получать само множество критериев, затем множество Парето в критериальном пространстве и только затем можно определить Парето-оптимальное множество в области допустимых решений X .

Существуют два основных метода нахождения множества Парето:

1. Метод обхода конусом применяется для непрерывной области критериев. Назовем конусом пространственный угол, образуемый лучами, идущими из общей вершины и ограниченными в каждой плоскости углом 90° . Направление ограничивающих лучей соответствует направлению оптимизации критериев.

Пусть для двух критериев: $k_1 \rightarrow \max (\rightarrow)$ и $k_2 \rightarrow \min (\downarrow)$, тогда конус имеет вид (рис. 58).

Для получения множества Парето достаточно установить вершину конуса на все точки границы области критериев, если при этом ни один луч не пересекает область, то вершина лежит в точке Парето, а если пересекает, то нет.

Пример: $k_1 \rightarrow \min$, $k_2 \rightarrow \min$ (см. рис. 59).

Множество Парето: $[AB] \cup [CD]$.

2. Метод прямоугольников применяют, когда критериальное пространство представляет собой отдельные точки или табличные значения. Сформулируем алгоритм для случая двух критериев, когда $k_1 \rightarrow \min$ и $k_2 \rightarrow \min$,

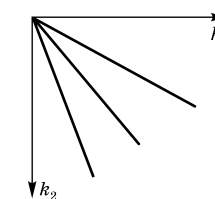


Рис. 58
Геометрическая
иллюстрация конуса
для двух критериев

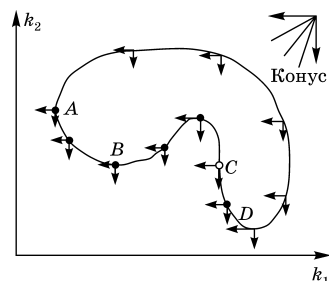


Рис. 59
Иллюстрация
метода обхода конусом

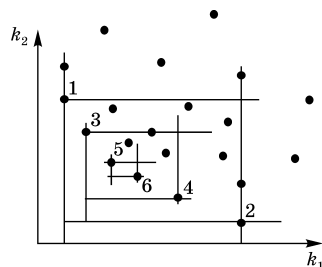


Рис. 60
Иллюстрация метода
прямоугольников

а критериальное пространство — точки на плоскости (рис. 60).

1) Фиксируем самые левые точки. Если их несколько, выбираем среди них самую нижнюю. Эта точка является точкой Парето, фиксируем ее.

2) Выберем самую нижнюю точку. Если их несколько, то выбираем самую левую. Это точка Парето, фиксируем ее.

3) Через полученные точки проводим вертикаль и горизонталь. Отбрасываем сами точки Парето, точки, лежащие на границе полученного прямоугольника, и точки вне прямоугольника.

4) К внутренним точкам полученного прямоугольника применяем алгоритм из пункта 1.

Алгоритм заканчивается тогда, когда внутри прямоугольника не останется ни одной точки.

Множество Парето: 1, 2, 3, 4, 5, 6.

Изложенный алгоритм легко обобщается на случай большего числа критериев, при других направлениях оптимизации, а также на случай табличного задания критериев.

7.5. МЕТОДЫ УСЛОВНОЙ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ

После нахождения множества Парето, если количество точек в нем ≥ 2 , встает вопрос о выборе единственного решения. Так как точки на множестве Парето обладают таким свойством, что каждая из них лучше по одному

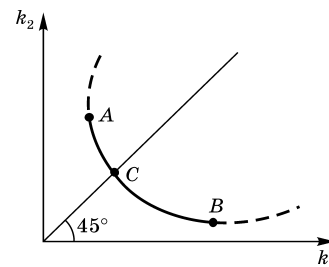


Рис. 61
Иллюстрация принципа
равенства критериев

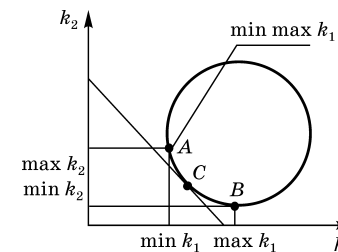


Рис. 62
Иллюстрация принципа
равного влияния и принципа
минимакса

критерию, но хуже по другому, то объективно предпочесть эти точки невозможно. Условная оптимизация предполагает введение условия согласованности в компонентах критерия, которое всегда является хотя и разумным, но субъективным. Подобные условия еще называют схемой компромиссов.

Рассмотрим несколько простейших из них.

Принцип равенства критериев. Рассматривают множество Парето для стандартных критериев. Тогда оптимальной считается та точка множества Парето, где $k_1 = k_2 = \dots = k_L$ — точка C (рис. 61).

Принцип равного влияния. Здесь оптимальной считается точка множества Парето, являющаяся точкой касания прямой $k_1 + k_2 \rightarrow \min$ и области Парето — точка A (рис. 62).

Принцип минимакса. Оптимальной считается точка множества Парето, где обеспечивается $\min_i \max_i k_i$ (рис. 62).

Кроме указанных простейших приемов применяются более сложные.

1. **Метод скаляризации.** Здесь формируется некоторая скалярная функция многих переменных:

$$K_0 = f(k_1, k_2, \dots, k_L),$$

где K_0 — скалярная величина, которая обобщает все критерии.

Наиболее распространенной разновидностью функции f является линейная функция

$$K_0 = \sum_{i=1}^L \alpha_i k_i \rightarrow \min(\max),$$

где α_i — вес каждой компоненты критерия, чем эта величина больше, тем большее влияние оказывает этот критерий. Для нестандартных критериев вес может иметь размерность, причем если $k_i \rightarrow \max$, то $\alpha_i \geq 0$, а когда $k_i \rightarrow \min$, то $\alpha_i \leq 0$.

Подобная комплексная оценка зачастую затруднительна. На практике часто используют и другие способы искусственного объединения нескольких показателей в один обобщенный показатель (критерий).

В качестве обобщенного критерия, например, берут дробь:

$$K_0 = (k_1 \cdot k_2 \cdot \dots \cdot k_m) / (k_{m+1} \cdot \dots \cdot k_L),$$

где k_1, \dots, k_m — увеличиваются, k_{m+1}, \dots, k_L — уменьшаются.

Вольное использование подобных критериев чревато опасностями и может привести к неправильным рекомендациям.

Общим недостатком «составных критериев» является то, что недостаток эффективности одного показателя всегда можно скомпенсировать за счет другого (например, малую вероятность выполнения боевой задачи — за счет малого расхода боеприпасов и т. п.).

На этот счет известна шутка Л. Н. Толстого, что если представить оценку человека в виде дроби:

$$\frac{\text{Достоинство человека}}{\text{Мнение о себе}},$$

то можно получить самую высокую оценку в случае

$$\frac{\text{Мало достоинств}}{\text{Нет мнения о себе}} = \infty.$$

2. Метод главного критерия. Здесь сначала выбирают главный критерий и объявляют его единственным. Все остальные критерии становятся управляемыми переменными, на которые накладываются ограничения, чтобы они были не хуже заданной величины k_i^0 . Например, если все

$k_i \rightarrow \min$ и главный k_1 , то получаем однокритериальную задачу математического программирования

$$\begin{aligned} k_1 &= f(X) \rightarrow \min, \\ \varphi_i(X) &< 0, \quad i = \overline{1, m}, \\ k_2(X) &< k_2^0, \quad k_3(X) < k_3^0, \dots, \quad k_L(X) < k_L^0. \end{aligned}$$

Тогда эти ограничения войдут в комплекс заданных условий. Например, в промышленности можно потребовать, чтобы прибыль $\rightarrow \max$, план по ассортименту — выполнен, а себестоимость — не выше заданной.

В военной операции понесенный ущерб противника максимальный, а потери не выходили за известные пределы.

Здесь все показатели эффективности, кроме одного, переводятся в разряд ограничений. Варианты решения, не укладывающиеся в заданные границы, сразу же отбрасываются как недопустимые.

Такой метод чаще всего используется при оптимизации сложных технических систем (самолетов, автомобилей, бытовой техники и т. п.).

3. Метод последовательных уступок. Проранжируем все критерии и расположим их в порядке убывающей важности: k_1, k_2, \dots, k_L .

Пусть все критерии нужно максимизировать. Сначала отбросим все критерии, кроме k_1 , и найдем допустимое решение, обращающее в максимум k_1 . Как правило, при этом другие критерии не получают своих наилучших значений. Поэтому исходя из практических соображений и точности исходных данных, назначается уступка Δk_1 , которую мы согласны допустить, для того чтобы обратиться в максимум k_2 . Наложим на критерий k_1 ограничение, чтобы он был не меньше, чем $k_1^* - \Delta k_1$ (k_1^* — максимально возможное значение k_1), и при этом ограничении ищем решение, обращающее в максимум k_2 . Далее снова назначается уступка в показателе k_2 , ценой которой находится максимум k_3 и т. п.

$$\begin{aligned} k_2 &\rightarrow \max, \\ k_{1\max} - \Delta k_1 &\leq k_1 \leq k_{1\max}. \end{aligned}$$

На втором этапе решаем задачу

$$\begin{aligned} k_3 &\rightarrow \max, \\ k_{1\max} - \Delta k_1 &\leq k_1 \leq k_{1\max}, \\ k_{2\max} - \Delta k_2 &\leq k_2 \leq k_{2\max} \text{ и т. д.} \end{aligned}$$

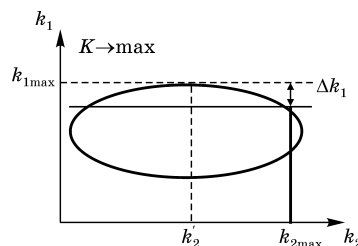


Рис. 63
Иллюстрация
метода уступок

Такой способ хорош тем, что здесь сразу видно, ценой какой «уступки» в одном показателе приобретаетс выигрыш в другом. Заметим, что иногда даже при малом Δk «свобода» выбора позволяет достичь существенной оптимизации k_2 . Это зависит от вида границы критериального пространства (рис. 63).

Очевидно, успех такого метода зависит от того, насколько «тупой» максимум. Процедура может быть повторена для других Δk .

4. Метод последовательной оптимизации. В некоторых случаях критерии системы не слишком связаны друг с другом, т. е. улучшение одного критерия, по крайней мере в ограниченных пределах, не «мешает» другому. Это характерно для сложных систем, состоящих из отдельных достаточно автономных подсистем (например: процессор компьютера, системы электропитания, охлаждения, отображения), когда отдельные критерии относятся к этим подсистемам. Расположим все критерии в порядке убывания важности. Сначала оптимизируем систему по важнейшему показателю, отбросив все остальные, затем по второму, и т. д.

Следует отметить, что в приведенных методах используется процедура упорядочения критериев по важности. Эта самостоятельная проблема является достаточно сложной и поэтому решается на основе методов экспертных оценок, изложенных в 10-й главе.

ГЛАВА 8

ИГРОВЫЕ МОДЕЛИ ПРИНЯТИЯ РЕШЕНИЙ

Рассмотренные нами математические модели выражают стремление получить максимальный результат для какой-то одной системы (фирмы, отрасли, страны). Если эта система в свою очередь состоит из элементов (подсистем), то считается, что их целевые функции совпадают, т. е. эти подсистемы работают «все как один». Во многих задачах это не так. Приходится анализировать ситуации, когда присутствует несколько сторон, преследующих различные интересы (цели). Это и боевые действия, и внешняя экономическая деятельность государств, и многое другое. При переходе к рыночной экономике и возникновении конкуренции как основного рычага развития экономики постановка задачи с отождествлением целей является неполной, а иногда и неправильной.

Интересы отдельных фирм сталкиваются на одном рыночном пространстве и иногда настолько различны, что конкуренция приобретает форму борьбы и поэтому невозможно оценить результат принимаемого решения единообразно. Такого рода ситуации называются *конфликтными* и могут быть описаны моделями, которые называются *играми*. Модели конфликтных ситуаций, где присутствует несколько сторон, преследующих различные интересы, называются *игровыми моделями*. Игровая модель представляет собой особый вид модели для принятия оптимальных решений. Здесь меняется само понятие оптимального решения. Теория, описывающая конфликтные ситуации с количественной стороны, называется *теорией игр*.

8.1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ ИГР

Ситуация, когда эффективность решения, принимаемого одной стороной, зависит от действий другой стороны, называется *конфликтной*. Любая конфликтная ситуация подразумевает наличие нескольких сторон, вступающих в конфликт и действующих в своих интересах.

Игра представляет собой математическую модель некоторой реальной конфликтной ситуации и отличается от реальной ситуации тем, что ведется по определенным правилам, устанавливающим последовательность ходов, возможные действия игроков, объем информации, доступный каждой из сторон, правила окончания и исход игры.

Стороны, участвующие в конфликте, называются *игроками*. По количеству участников игры различаются на *парные*, когда в игре принимают участие две стороны, и *множественные*, когда количество сторон больше двух. Парные игры могут быть *антагонистическими* (игры с нулевой суммой), когда увеличение выигрыша одной из сторон на некоторую величину приводит к уменьшению выигрыша другой стороны на ту же величину, и наоборот. Примерами антагонистических игр являются шахматы, «крестики–нолики», азартная игра «орлянка». Множественные игры в свою очередь могут быть *коалиционными*, когда часть игроков соединяют свои интересы и действуют как один игрок.

Стратегией игрока называется совокупность правил, определяющих выбор варианта действий при каждом личном ходе в зависимости от сложившейся ситуации. По количеству стратегий игры делятся на *конечные*, когда количество стратегий игроков конечно, и *бесконечные*, когда число стратегий хотя бы одного игрока бесконечно.

Выбор одной из предусмотренных правилами игры стратегий и ее осуществление называется *ходом*. Ходы бывают *личными*, когда игрок делает осознанный выбор стратегии и *случайными*, когда выбор стратегии осуществляется независимо от игрока некоторым случайным образом.

Существует два способа описания игр:

- *позиционный способ* используется для описания игр в развернутой форме, когда игра представляется графом последовательных шагов, называемым деревом игры;
- *нормальный способ* заключается в явном представлении совокупности стратегий игроков и платежной функции, определяющей выигрыш каждой из сторон в зависимости от выбранной стратегии.

Если антагонистическая игра описана платежной матрицей, то это *матричная игра*. В игре с ненулевой суммой выигрыша каждая из сторон задается своей платежной матрицей. Такие игры называются *биматричными*.

Задачей теории игр является выработкой рекомендаций по рациональному образу действий участников конфликта.

8.1.1. ПОЗИЦИОННЫЕ ИГРЫ

Позиционная игра — это бескоалиционная игра, описывающая конфликтную ситуацию, динамика которой оказывает влияние на поведение участников. Процесс игры заключается в чередовании ходов, которые могут быть личными или случайными. Состояния игры называются позициями, а возможные выборы игрока в каждой позиции — *альтернативами*.

В каждой позиционной игре выделяется *начальная позиция*, которая рассматривается как начало игры, а также некоторое множество *конечных позиций*, при достижении которых игра завершается. При этом каждой конечной вершине поставлен в соответствие вектор выигрышей участников (каждая компонента которого соответствует выигрышу определенного игрока), совокупность значений которого определяет *функцию выигрыша*.

Позиционную игру удобно представлять графом, называемым деревом игры. При этом каждая вершина графа соответствует позиции игры, а каждое ребро характеризует одну из возможных альтернатив в данной позиции. Пользуясь графическим описанием игры, можно сказать, что процесс описания игры состоит в переходе от началь-

ной позиции к конечной через последовательно следующие одна за другой промежуточные позиции. Каждая конечная вершина может быть соединена с начальной единственной цепью, которая называется *партией*.

Различают позиционные игры с *полной информацией* и позиционные игры с *неполной информацией*. В играх с полной информацией игроку в свой ход доступна информация о той позиции (той вершине дерева игры), в которой он находится. В играх с неполной информацией игрок знает лишь, в каком множестве позиций он находится, но не знает в какой именно позиции этого множества. Такое множество позиций называется *информационным множеством*. Отметим, что из позиций информационного множества выходит одно и то же число альтернатив, так как иначе по числу альтернатив игрок может уточнить позицию игры. Следует также отметить, что в играх с полной информацией информационное множество состоит из единственной фактической позиции игрока. Например, шахматы и шашки — это игры с полной информацией, а бридж и покер — нет.

Пример. *Игра в орлянку.* Игроки M и N делают по одному ходу, поочередно выбирая «орел» (О) или «решку» (Р). Если выборы совпадают, то игрок N выигрывает у игрока M одну единицу, в противном случае выигрывает игрок M . На дереве игры (рис. 64) пары чисел показывают

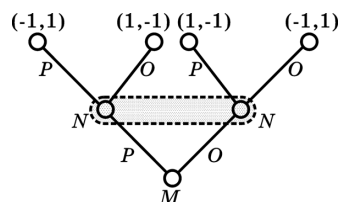


Рис. 64
Дерево игры «Орлянка»

выигрыши игроков. Затененная область охватывает позиции из одного информационного множества для случая, если игрок N делает выбор, не зная выбора игрока M . Таким образом, игра в орлянку, представленная на рисунке, есть игра с неполной информацией. Представим теперь, что игрок N делает выбор, зная выбор игрока M на предыдущем шаге. Такая игра имеет мало смысла, поскольку игрок N , зная в какой именно позиции он находится, не даст противнику ни малейшего шанса. Однако она демонстрирует пример игры с полной

информацией. В этом случае информационное множество игрока N распадается на пару одноточечных информационных множеств.

8.1.2. СТРАТЕГИЯ, НОРМАЛЬНАЯ ФОРМА ИГРЫ

В интуитивном понимании стратегия есть некоторый план разыгрывания игры, согласно которому еще до начала игры игроки принимают решение о своих действиях в зависимости от действий противника. Приведем более точное определение.

Определение. *Чистая стратегия* игрока есть некоторая функция, которая каждому информационному множеству ставит в соответствие индекс одной из возможных альтернатив этого информационного множества.

Таким образом, в каждой позиции игры выбранная стратегия однозначно определяет ход игрока, а выбор игроками своих чистых стратегий однозначно определяет исход игры, т. е. партию.

Рассмотрим игру n лиц и обозначим множество всех чистых стратегий i -го игрока через S^i , а элементы множества — чистые стратегии — через $s_i \in S^i$ ($i = 1, n$).

Поскольку результаты случайных ходов (если таковые имеются) известны только в вероятностном смысле, то рассматривается математическое ожидание функции выигрыша

$$h(s_1, s_2, \dots, s_n) = (h_1(s_1, s_2, \dots, s_n), \\ h_2(s_1, s_2, \dots, s_n), \dots, h_n(s_1, s_2, \dots, s_n)),$$

где i -я компонента вектора выигрыша есть выигрыш i -го игрока.

Следовательно, функция выигрыша может быть представлена в виде n -мерной таблицы, элементами которой являются n -векторы (в случае двух игроков $n = 2$ получаем матрицу элементами которой являются пары вещественных чисел). Такая n -мерная таблица называется нормальной формой игры, а процесс приведения игры к нормальной форме называется нормализацией позиционной игры.

Пример. В игре в орлянку каждый игрок имеет две стратегии Р и О. Нормальная форма игры имеет вид таблицы 43.

Таблица 43

	Р	О
Р	(-1, 1)	(1, -1)
О	(1, -1)	(-1, 1)

Каждая строка матрицы представляет стратегию игрока M , а каждый столбец — стратегию игрока N .

Пример. Игра полковника Блотто. Игра полковника Блотто — об-

щее название большого класса тактических военных игр. Приведем один из наиболее простых вариантов этой игры. Две воюющие армии ведут борьбу за два пункта. Первая армия под командованием полковника Блотто состоит из четырех полков; вторая под командованием Кижы состоит из трех полков. Армия, которая посылает больше полков на тот или иной пункт, занимает его и уничтожает все направленные на этот пункт силы противоположной стороны, получая единицу, как за занятый пункт, так и за каждый уничтоженный полк противника. Блотто и Кижы должны решить, как распределить силы, чтобы выиграть как можно больше очков. В модели представлены два игрока Блотто и Кижы. Стратегии игроков есть распределение полков между двумя пунктами. Стратегии Блотто — это распределение четырех полков между первым и вторым пунктами. Имеется пять таких распределений. Они и составляют множество чистых стратегий первого игрока (т. е. $m = 5$). Вот это множество:

$\{(4, 0), (3, 1), (2, 2), (1, 3), (0, 4)\}$.

Точно также у Кижы его четыре чистые стратегии ($n = 4$) могут быть представлены парами

$\{(3, 0), (2, 1), (1, 2), (0, 3)\}$.

В каждой паре первое число указывает число полков, направленных на первый пункт, второе число — на второй пункт. По усло-

Таблица 44

	N1	N2	N3	N4
M1	4	2	1	0
M2	1	3	0	-1
M3	-2	2	2	-2
M4	-1	0	3	1
M5	0	1	2	4

вию игра является антагонистической, поэтому выигрыш одного игрока (например, Кижы) равен выигрышу другого игрока с обратным знаком.

В таблице 44 приведена матрица выигрышей Блотто в зависимости от действий, предпринимаемых игроками. Эти же числа указывают проигрыши Кижы.

8.2. ИГРЫ ДВУХ ЛИЦ С НУЛЕВОЙ СУММОЙ

8.2.1. МАТРИЧНАЯ ИГРА С СЕДЛОВОЙ ТОЧКОЙ

Рассмотрим конечную антагонистическую игру в нормальной форме. Такую игру удобно описывать матрицей выигрышей, называемой также *платежной матрицей*¹, где чистым стратегиям игроков соответствуют строки и столбцы матрицы, а элементы матрицы определяют платежи (выигрыш и проигрыш) игроков.

Пусть игрок M имеет m чистых стратегий M_1, M_2, \dots, M_m , а игрок N — n чистых стратегий N_1, N_2, \dots, N_n . Платежная матрица $A = (a_{ij})$, $i = 1, m$, $j = 1, n$. Будем полагать для определенности, что если игрок M применяет свою i -ю чистую стратегию, игрок N — j -ю чистую стратегию, то выигрыш первого составит a_{ij} , а выигрыш второго — $-a_{ij}$ (отрицательный выигрыш также называют проигрышем). Таким образом, столкновение интересов игроков заключается в стремлении игрока M увеличить свой выигрыш, а игрока N — уменьшить свой проигрыш. Необходимо наметить пути решения, состоящие в выборе игроками рациональных стратегий.

Полагаем, что игрок M при выборе своей i -й чистой стратегии рассчитывает на наихудшее для себя развитие событий из-за активных действий второго игрока, стремящегося уменьшить выигрыш M . Вследствие этого из возможных достижимых выигрышей $a_{i1}, a_{i2}, \dots, a_{in}$ игроку M

¹ Для игр с полной информацией платежную матрицу можно построить всегда.

приходится рассчитывать на выигрыш (гарантированный выигрыш)

$$\alpha_i = \min_j a_{ij}.$$

Наилучшей с точки зрения M будет та стратегия, которая принесет максимальный гарантированный выигрыш. Она называется *максиминной стратегией* и обеспечивает выигрыш

$$\alpha = \max_i \alpha_i = \max_i \min_j a_{ij}, \quad (8.1)$$

называемый *нижней ценой игры*. Предложенный подход к выбору действий игрока M отражает идею гарантированного результата: независимо от действий второго игрока игрок M получает выигрыш не меньше α [31].

Аналогичные рассуждения для игрока N приводят к минимаксной стратегии, обеспечивающей ему максимальный проигрыш не больше

$$\beta = \min_j \beta_j = \min_j \max_i a_{ij}, \quad (8.2)$$

где $\beta_j = \max_i a_{ij}$ — максимальный (гарантированный) проигрыш при выборе игроком N своей j -й чистой стратегии. Величина β называется *верхней ценой игры* и отражает стремление игрока N получить гарантированный результат независимо от действий противной стороны.

В общем случае $\alpha \leq \beta$.

Равенство $\alpha = \beta$ означает, что в платежной матрице присутствует элемент a_{pq} , который является одновременно минимальным в p -й строке и максимальным в q -м столбце. Такой элемент называется *седловым элементом* матрицы², а матричная игра в этом случае становится *игрой с седловой точкой*. Стратегии M_p и N_q , соответствующие седловой точке, называются *оптимальными чистыми стратегиями*, а величина $\gamma = \alpha = \beta$ — *чистой ценой игры*. Решением матричной игры с седловой точкой будет тройка чисел: (p, q, γ) .

² Матрица может иметь несколько седловых элементов, значения которых совпадают. Характер рекомендуемых решений игры при этом не изменится.

Оптимальные стратегии обладают важным свойством: они определяют в игре *положение равновесия*, заключающееся в том, что ни одна из сторон не заинтересована в отходе от своей оптимальной стратегии, поскольку любой переход к новой стратегии означает отказ от принципа гарантированного результата и ведет к проигрышу.

Пример. Матричная игра

Т а б л и ц а 45

	N1	N2	N3
M1	5	1	3
M2	3	2	4
M3	-3	0	1

имеет седловую точку. Элемент $a_{22} = 2$ является седловым элементом матрицы. Таким образом, решение игры будет тройка $(2, 2, 2)$.

8.2.2.

МАТРИЧНЫЕ ИГРЫ БЕЗ СЕДЛОВОЙ ТОЧКИ

На практике более распространен другой случай, когда $\alpha < \beta$, т. е. платежная матрица не имеет седловой точки. Игрок M обеспечивает себе выигрыш не меньше α , а игрок N — проигрыш не больше β .

Возникает вопрос, как распределить между игроками разность $\beta - \alpha$? Оказывается, что эффективного распределения разности игроки могут достичь (при многократном повторении игры) путем чередования своих чистых стратегий.

Если в игре без седловой точки один из игроков раскрывает противнику свою стратегию, то лучшее на что он может рассчитывать есть нижняя (для второго игрока — верхняя) цена игры. Поэтому игрокам следует как-то скрыть свое поведение от противника, чтобы ослабить влияние информационного фактора и получить преимущество. С этой целью в действия игроков вносится некоторый элемент случайности так, чтобы каждый отдельный ход оставался непредсказуемым, но вся совокупность ходов обладала вполне определенными вероятностным

распределением. В этом и состоит идея использования смешанных стратегий [2].

Смешанная стратегия игрока есть вероятностное распределение на множестве его чистых стратегий. В случае конечной игры смешанная стратегия игрока есть вектор $x = (x_1, x_2, \dots, x_m)'$, где $x_i \geq 0$ — вероятность использования i -й чистой стратегии, $\sum_{i=1}^m x_i = 1$. Обозначим множество всех смешанных стратегий игроков M и N через X и Y соответственно.

Предположим, что игрок M выбирает смешанную стратегию x , а игрок N выбирает y , тогда *ожидаемый выигрыш* (математическое ожидание выигрыша) M будет равен

$$h(x, y) = \sum_i \sum_j a_{ij} x_i y_j = x' A y. \quad (8.3)$$

Как и прежде при выборе стратегий игроки должны ориентироваться на худшее развитие событий. Тогда для игрока M лучшей оказывается максиминная стратегия x^* , позволяющая достичь выигрыша

$$\gamma_M = \max_{x \in X} \min_{y \in Y} h(x, y).$$

Аналогично для игрока N лучшей будет минимаксная стратегия y^* с выигрышем

$$\gamma_N = \min_{y \in Y} \max_{x \in X} h(x, y).$$

Справедлива следующая важная теорема, доказательство которую опирается на известную теорему Какутани [33], которую приведем здесь без доказательства.

Теорема Какутани. Пусть Z — компактное выпуклое множество в \mathbf{R}^n и ϕ — многозначное отображение, переводящее точки из Z в компактные выпуклые подмножества из Z и удовлетворяющее условию: если $x_k \in Z$, $\lim_{k \rightarrow \infty} x_k = x$, $y_k \in \phi(x_k)$ и $\lim_{k \rightarrow \infty} y_k = y$, то $y \in \phi(x)$. Тогда существует такое $x^* \in Z$, что $x^* \in \phi(x^*)$.

Теорема о минимаксе. В конечной игре двух лиц с нулевой суммой и полной информацией имеет место равенство

$$g_M = g_N.$$

Доказательство. Множества X и Y — выпуклые компактные множества (многогранники) в соответствующих евклидовых пространствах, а функция $h(x, y)$ билинейна и поэтому непрерывна на $X \times Y$, вогнута по x и выпукла по y . Положим

$$\begin{aligned} Y_x &= \left\{ y^* \in Y \mid h(x, y^*) = \min_{y \in Y} h(x, y) \right\}, \\ X_y &= \left\{ x^* \in X \mid h(x^*, y) = \max_{x \in X} h(x, y) \right\}, \\ \phi(x, y) &= X_y \times Y_x. \end{aligned}$$

Легко проверить, что точечно-множественное отображение ϕ удовлетворяет условиям теоремы Какутани и, значит, имеет неподвижную точку. Она является седловой точкой функции $h(x, y)$, откуда вытекает условие теоремы.

Согласно теореме существуют такие стратегии x^* и y^* , которые позволяют получить среднеожидаемый выигрыш $\gamma = \gamma_N = \gamma_M$, называемый *ценой игры*.

Стратегии x^* , y^* называются *оптимальными стратегиями* игроков M и N соответственно, если

$$h(x^*, y^*) = g.$$

Это условие равносильно следующему, поскольку (x^*, y^*) является седловой точкой функции $h(x, y)$:

$$h(x, y^*) \leq h(x^*, y^*) \leq h(x^*, y).$$

Тройка (x^*, y^*, γ) называется *решением матричной игры*.

Теорема (основная теорема матричных игр). Всякая матричная игра имеет по крайней мере одно решение в смешанных стратегиях.

Доказательство. Достаточно показать, что функция $h(x, y)$ имеет седловую точку на $X \times Y$. Доказательство этого факта напрямую следует из доказательства теоремы о минимаксе.

Оптимальные смешанные стратегии обладают важным свойством: ни одна из сторон не заинтересована в отходе от своей оптимальной смешанной стратегии, поскольку каждый игрок при многократном повторении игры, при-

держиваясь оптимальных смешанных стратегий, получит более выгодный для себя результат [35].

Стратегии игроков, входящие в их оптимальные смешанные стратегии (с ненулевой вероятностью), называются *активными*.

Множество активных стратегий называют *носителем смешанной стратегии*.

Теорема (об активных стратегиях). Применение оптимальной смешанной стратегии обеспечивает игроку максимальный средний выигрыш (минимальный средний проигрыш), равный цене игры γ , независимо от того, какие действия предпринимает другой игрок в пределах его активных стратегий.

Доказательство. Пусть J_A и J_B — множество индексов, соответствующих активным стратегиям игроков A и B соответственно. Пусть γ — цена игры, достижение которой обеспечивается применением оптимальных стратегий x^* и y^* игроков A и B соответственно, а $\gamma_i = h(x^*, y_i)$, где γ_i — применение игроком B своей i -й чистой стратегии. Очевидно, что $\gamma = \sum_{i \in J_B} \gamma_i y_i^*$, причем $\sum_{i \in J_B} y_i^* = 1$. В то же время для

любого i должно выполняться $\gamma_i \geq \gamma$ (переход к неоптимальной стратегии игрока B может только увеличить его проигрыш). Сохранить в этих условиях последнее равенство можно, лишь положив $\gamma_i = \gamma$, для всех $i \in J_B$, что и доказывает утверждение теоремы в случае, когда стратегии x^* противопоставляются отдельно взятым y_i .

Аналогичный вывод нетрудно получить и при сравнении x^* с произвольной смешанной стратегией y , определяемой вероятностями $\gamma_i \neq \gamma^*$, $i \in J_B$. Здесь ожидаемый выигрыш A (проигрыш B) составит $\bar{\gamma} = \sum_{i \in J_B} \gamma_i y_i$. Он должен всегда оцениваться как $\bar{\gamma} \geq \gamma$ в силу неоптимальности y , однако нельзя ожидать $\bar{\gamma} > \gamma$, так как это означало бы неоптимальность x^* , что противоречит исходным предположениям. Остается, как и ранее, считать $\bar{\gamma} = \gamma$.

Аналогично, повторяя рассуждения для неизменной стратегии y^* , получим тем самым утверждение теоремы [10].

8.2.3. АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ИГР 2Х2

В частном случае матричных игр 2×2 решение может быть получено аналитически. Пусть дана матричная игра с платежной матрицей

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

Если платежная матрица имеет седловую точку, то решение очевидно. Предположим теперь, что игра не имеет седловой точки. Отсюда следует, что оптимальные стратегии $x = (x_1, x_2)$ и $y = (y_1, y_2)$ должны иметь положительные компоненты. Далее, если цена игры есть γ , то из (8.3) следует

$$a_{11}x_1y_1 + a_{12}x_1y_2 + a_{21}x_2y_1 + a_{22}x_2y_2 = g,$$

или

$$x_1(a_{11}y_1 + a_{12}y_2) + x_2(a_{21}y_1 + a_{22}y_2) = g.$$

Оба выражения в скобках в левой части не превосходят цены игры γ , так как это выигрыши, когда N действует оптимально, а M выбирает чистую стратегию (отклоняется от оптимальной). Поскольку $x_1, x_2 > 0$ и $x_1 + x_2 = 1$ для достижения равенства оба выражения в скобках должны быть равны γ , т. е.

$$\begin{aligned} a_{11}y_1 + a_{12}y_2 &= g, \\ a_{21}y_1 + a_{22}y_2 &= g. \end{aligned}$$

Аналогично можно показать, что

$$\begin{aligned} a_{11}x_1 + a_{21}x_2 &= g, \\ a_{12}x_1 + a_{22}x_2 &= g. \end{aligned}$$

Эти уравнения вместе с уравнениями

$$\begin{aligned} x_1 + x_2 &= 1, \\ y_1 + y_2 &= 1. \end{aligned}$$

позволяют найти x, y, γ .

Сформулируем полученный результат в виде теоремы.

Теорема. Если A — матричная игра 2×2 без седловой точки и $(a_{11} - a_{12} - a_{21} + a_{22}) \neq 0$, то ее единственные оптимальные стратегии и цена игры определяются формулами

$$x^* = \frac{(a_{22} - a_{21}, a_{11} - a_{12})'}{a_{11} - a_{12} - a_{21} + a_{22}}, \quad (8.4)$$

$$y^* = \frac{(a_{22} - a_{12}, a_{11} - a_{21})'}{a_{11} - a_{12} - a_{21} + a_{22}}, \quad (8.5)$$

$$\gamma = \frac{a_{11}a_{22} - a_{12}a_{21}}{a_{11} - a_{12} - a_{21} + a_{22}}. \quad (8.6)$$

Пример. Для платежной матрицы

Т а б л и ц а 46

	N1	N2
M1	0	3
M2	2	1

аналитическое решение будет иметь вид:

$$x^* = \frac{1}{-4}(-1, -3)' = \left(\frac{1}{4}, \frac{3}{4}\right)',$$

$$y^* = \frac{1}{-4}(-2, -2)' = \left(\frac{1}{2}, \frac{1}{2}\right)',$$

$$\gamma = \frac{-6}{-4} = \frac{3}{2}.$$

8.2.4. УПРОЩЕНИЕ ИГР

Решение матричной игры без седловой точки тем сложнее, чем больше размер платежной матрицы. К счастью, отыскание решения можно упростить, используя следующие правила [32].

1. Вычеркивание дублирующих стратегий.

Если в платежной матрице A игры все элементы одной строки (столбца) равны соответствующим элементам другой строки (столбца), то соответствующие строкам (столб-

цам) стратегии называются дублирующими. Дублирующие стратегии можно объединить.

2. Вычеркивание заведомо невыгодных стратегий.

Если в платежной матрице A все элементы некоторой строки (столбца), определяющей чистую стратегию игрока не больше (не меньше) соответствующих элементов другого столбца (строки), то такую стратегию называют заведомо невыгодной. Заведомо невыгодные стратегии можно удалить, поскольку их использование нецелесообразно.

3. Замена некоторых групп чистых стратегий смешанными.

Некоторые группы чистых стратегий могут быть объединены в одну смешанную стратегию с соответствующим изменением элементов платежной матрицы.

Пример. Решить игру со следующей платежной матрицей (табл. 47):

Т а б л и ц а 47

	N1	N2	N3
M1	6	3	2
M2	2	3	6
M3	2	5	2

В матрице нет дублирующих и заведомо невыгодных стратегий, но элементы на пересечении первой и второй строк и первого и третьего столбцов расположены симметрично, а элементы в третьей строке в первом и третьем столбцах совпадают. Следовательно, если стратегии N_1 и N_3 войдут в решение, то с одинаковыми вероятностями, а значит можно заранее объединить их в одну смешанную стратегию $N_{1,3}$, элементы которой равны полусуммам соответствующих элементов N_1 и N_3 . Получим матрицу:

Т а б л и ц а 48

	N1, N3	N2
M1	4	3
M2	4	3
M3	2	5

Стратегии M_1 и M_2 — дублирующие. Объединяем их в одну $M_{1,2}$ и получаем матрицу 2×2 (табл. 49):

Т а б л и ц а 49

	$N1, N3$	$N2$
$M1, M2$	4	3
$M3$	2	5

Игра имеет следующее решение:

$$x^* = \left(\frac{3}{8}, \frac{3}{8}, \frac{1}{4}\right)', \quad y^* = \left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right)', \quad \gamma = \frac{7}{2}.$$

8.2.5.

ГЕОМЕТРИЧЕСКОЕ РЕШЕНИЕ ИГР $2 \times N$, $M \times 2$

Рассмотрим игру 2×2 . Отложим по оси абсцисс единичный отрезок. Левый конец будет соответствовать чистой стратегии M_1 , а правый — стратегии M_2 (рис. 65). Любая точка внутри отрезка обозначает смешанную стратегию $x = (x_1, x_2)$. На оси ординат будем откладывать выигрыши. На левой вертикали будут выигрыши, полученные при применении чистых стратегий (M_1, N_1) и (M_1, N_2) , а на правой — чистых стратегий (M_2, N_1) и (M_2, N_2) . Если игрок M применяет смешанные стратегии, а N — чистые, то выигрыши будут ординатами соответствующих прямых. Решим игру для игрока M — для этого найдем сначала самые маленькие выигрыши при различных смешанных стратегиях — это будет *линия минимумов*. Затем на этой линии найдем наивысшую точку. Получим *максимин*.

Аналогично задача решается для игрока N — находим *линию максимумов* и *минимакс* (рис. 66).

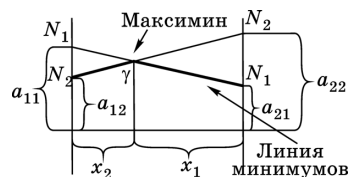


Рис. 65
Линия минимумов, максимин

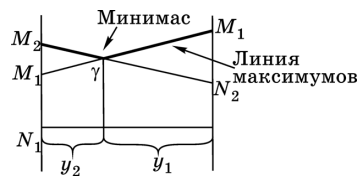


Рис. 66
Линия максимумов, минимакс

По принципу максимина игрок M сначала фиксирует свои минимальные выигрыши, а затем выбирает максимальный выигрыш. Так получаются величины x_1 и x_2 y_1 и y_2 на единичном отрезке и величина цены игры γ .

Примечания.

- решение не обязательно находится на пересечении прямых;
- если у одного игрока две стратегии, а у другого три и более стратегий, то игра также может быть решена геометрически.

Пример. Решить матричную игру с платежной матрицей

	$N1$	$N2$	$N2$	$N3$
$M1$	3	0	-1	2
$M2$	1	-1	-2	0
$M3$	-1	-2	4	-3

Все элементы строки M_2 меньше элементов строки M_1 , поэтому стратегию M_2 можно отбросить как заведомо невыгодную. Все элементы столбца N_1 больше соответствующих элементов столбца N_2 , поэтому стратегию N_1 отбрасываем как заведомо невыгодную. После вычеркивания столбца N_1 и строки M_2 игра упрощается (табл. 50). Строим график решения со стороны игрока M .

Т а б л и ц а 50

	$N2$	$N2$	$N3$
$M1$	0	-1	2
$M3$	-2	4	-3

Находим линию минимумов (на рисунке 67 выделено жирной линией) и определяем на ней максимум, точку P . Построив уравнения линий, образующих точку пересечения P , легко найти значения x_1, x_2, γ . В нашем случае эти значения будут

$$x_1 = \frac{6}{7}; \quad x_2 = \frac{1}{7}; \quad \gamma = -\frac{2}{7}.$$



Рис. 67
Линия минимумов
первого игрока

Решим игру со стороны игрока N . Из графика на рисунке 68 видно, что стратегия N_4 не формирует оптимальное решение — точку P , т. е. не является *активной*, следовательно, ее отбрасываем (она в согласованном решении не участвует).

Находим линию максимумов и определяем на ней минимум — точку Q . Аналогично предыдущему случаю найдем точное значение координат точки минимума

$$y_2 = \frac{5}{7}, \quad y_3 = \frac{2}{7} \quad \text{и} \quad \gamma = -\frac{2}{7}.$$

Следовательно, игрок M должен чередовать стратегии M_1 и M_3 с вероятностями $\frac{6}{7}$ и $\frac{1}{7}$ соответственно. Игрок N должен чередовать стратегии N_2 и N_3 с вероятностями $\frac{5}{7}$ и $\frac{2}{7}$. При этом цена игры $\gamma = -\frac{2}{7}$.

8.2.6. РЕШЕНИЕ МАТРИЧНОЙ ИГРЫ СВЕДЕНИЕМ К ПАРЕ ДВОЙСТВЕННЫХ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Пусть дана конечная матричная игра с платежной матрицей A размера $m \times n$ без седловой точки. Без ограничения общности будем считать, что элементы платежной матрицы положительны, тогда цена игры γ также положительна. Рассмотрим последовательно стратегии первого и второго игроков.

Стратегия игрока M .

При любой стратегии N_j игрока N оптимальная смешанная стратегия $x = (x_1, x_2, \dots, x_m)$ игрока M обеспечит его средний выигрыш не меньше γ , т. е. выполняется:

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

$$\sum_{i=1}^m x_i = 1, \quad x_i \geq 0, \quad i = \overline{1, m}.$$

Введем обозначение

$$u_i = \frac{x_i}{\gamma}, \quad i = \overline{1, m}.$$

С учетом этого получим:

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

$$\sum_{i=1}^m u_i = \frac{1}{\gamma}, \quad u_i \geq 0, \quad i = \overline{1, m}.$$

Поскольку игрок M стремится максимально увеличить свой гарантированный выигрыш, задача сводится к отысканию такого вектора $u = (u_1, u_2, \dots, u_m)'$, чтобы

$$b'u \rightarrow \min,$$

где $b = (1, 1, \dots, 1)'$ — m -вектор, при условии:

$$A'u \geq c,$$

где $c = (1, 1, \dots, 1)'$ — n -вектор,

$$u \geq 0.$$

Таким образом для получения оптимального решения для игрока M необходимо решить задачу линейного программирования.

Стратегия игрока N .

Аналогично для игрока N , оптимальная смешанная стратегия $y = (y_1, y_2, \dots, y_n)$ при любом выборе игроком M своих стратегий M_i обеспечит проигрыш игроку N не больше γ :

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

$$\sum_{j=1}^n y_j = 1, \quad y_j \geq 0, \quad j = \overline{1, n}.$$

Обозначив

$$v_j = \frac{y_j}{\gamma}, \quad j = \overline{1, n},$$

получим:

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

$$\sum_{j=1}^n v_j = \frac{1}{\gamma}, \quad v_j \geq 0, \quad j = \overline{1, n}.$$

Поскольку игрок N стремится сделать свой гарантированный проигрыш минимальным, задача сводится к отысканию такого $v = (v_1, v_2, \dots, v_n)'$, для которого выполняется:

$$c'v \rightarrow \max,$$

при условии:

$$Av \leq b,$$

$$v \geq 0.$$

Рассмотренные для игроков задачи представляют собой пару двойственных задач линейного программирования. Сформулируем полученный результат в виде теоремы [36].

Теорема. Решение матричной игры с положительной платежной матрицей

$$A = (a_{ij})_{\substack{i=\overline{1,m} \\ j=\overline{1,n}}},$$

где $a_{ij} \geq 0$ равносильно решению пары двойственных задач линейного программирования:

$$\begin{cases} b'u \rightarrow \min, \\ A'u \geq c, \\ u \geq 0, \end{cases} \quad (8.7)$$

$$\begin{cases} c'v \rightarrow \max, \\ Av \leq b, \\ v \geq 0, \end{cases} \quad (8.8)$$

где $b = (1, 1, \dots, 1)'$ — m -вектор, $c = (1, 1, \dots, 1)'$ — n -вектор.

Решение матричной игры (x^*, y^*, γ) связано с решением (u^*, v^*) пары двойственных задач следующими соотношениями.

$$\gamma = \frac{1}{b'u^*} = \frac{1}{c'v^*},$$

$$x^* = \gamma u^*,$$

$$y^* = \gamma v^*.$$

Пример. Рассмотрим решение A матричной игры на примере матрицы для «игры полковника Блотто»:

Т а б л и ц а 51

	N1	N2	N3	N4
M1	4	2	1	0
M2	1	3	0	-1
M3	-2	2	2	-2
M4	-1	0	3	1
M5	0	1	2	4

Решение не изменится, если ко всем элементам матрицы прибавить одну и ту же величину, чтобы все элементы были неотрицательными (при этом значение цены игры увеличится на ту же величину). Получим следующую матрицу

Т а б л и ц а 52

	N1	N2	N3	N4
M1	6	4	3	2
M2	3	5	2	1
M3	0	4	4	0
M4	1	2	5	3
M5	2	3	4	6

Применив рассуждения, приведенные выше, мы получим пару двойственных задач линейного программирования. Для игрока M (Блотто):

$$\begin{cases} u_1 + u_2 + u_3 + u_4 + u_5 \rightarrow \min \\ 6u_1 + 3u_2 + u_4 + 2u_5 \geq 1 \\ 4u_1 + 5u_2 + 4u_3 + 2u_4 + 3u_5 \geq 1 \\ 3u_1 + 2u_2 + 4u_3 + 5u_4 + 4u_5 \geq 1 \\ 2u_1 + u_2 + 3u_4 + 6u_5 \geq 1 \\ u_1, u_2, \dots, u_5 \geq 0. \end{cases}$$

Для игрока N (Киж):

$$\begin{cases} v_1 + v_2 + v_3 + v_4 \rightarrow \max \\ 6v_1 + 4v_2 + 3v_3 + 2v_4 \leq 1 \\ 3v_1 + 5v_2 + 2v_3 + v_4 \leq 1 \\ 4v_2 + 4v_3 \leq 1 \\ v_1 + 2v_2 + 5v_3 + 3v_4 \leq 1 \\ 2v_1 + 3v_2 + 4v_3 + 6v_4 \leq 1 \\ v_1, v_2, \dots, v_4 \geq 0. \end{cases}$$

Полученная пара двойственных задач линейного программирования, решается, например, симплекс-методом (см. главу 2) с последующим применением условий дополняющей нежесткости [36].

8.3. ИГРЫ ДВУХ ЛИЦ С ПРОИЗВОЛЬНОЙ СУММОЙ

Рассмотренные ранее модели игр с нулевой суммой достаточно точно описывают салонные игры, но в более общем случае интересы двух игроков не являются прямо противоположными и выигрыш одного не обязательно равен проигрышу другого игрока. Такие игры называют *играми с произвольной суммой*. В играх с произвольной суммой игроки зачастую могут выгадать путем кооперирования.

Игру двух лиц с произвольной суммой можно описать парой $m \times n$ матриц $A = (a_{ij})$ и $B = (b_{ij})$ или одной матрицей (A, B) с элементами (a_{ij}, b_{ij}) , где a_{ij}, b_{ij} — выигрыши игроков M и N соответственно при условии, что они использу-

ют свои i -ю и j -ю чистые стратегии. Игра в такой форме называется *биматричной игрой*.

Различают *некооперативные биматричные игры*, в которых запрещен любой тип соглашений между игроками и *кооперативные игры*, в которых такая кооперация разрешена.

8.3.1. НЕКООПЕРАТИВНЫЕ БИМАТРИЧНЫЕ ИГРЫ

Смешанные стратегии в этом случае имеют тот же смысл, что и в матричных играх. Введем понятие равновесия.

Пара смешанных стратегий (x^*, y^*) в биматричной игре (A, B) является *ситуацией равновесия*, если для любых смешанных стратегий $x \in X$ и $y \in Y$

$$\begin{aligned} h_A(x, y^*) &\leq h_A(x^*, y^*); \\ h_B(x^*, y) &\leq h_B(x^*, y^*), \end{aligned}$$

где $h_A(x, y) = x' Ay$, $h_B(x, y) = x' By$ — ожидаемый выигрыш первого и второго игроков при использовании ими своих смешанных стратегий x и y соответственно.

Теорема (Дж. Нэш). Каждая биматричная игра имеет по крайней мере одну равновесную ситуацию в смешанных стратегиях.

Доказательство [40]. Множества смешанных стратегий X и Y игроков M и N — выпуклые многогранники, поэтому множество ситуаций определяется декартовым произведением $X \times Y$ и является компактным выпуклым множеством.

Пусть φ — многозначное отображение,

$$\varphi: X \times Y \rightarrow X \times Y,$$

определяемое соотношением

$$\begin{aligned} \varphi: (x_0, y_0) &\rightarrow \{(x', y') \mid h_A(x', y_0) \max_X h_A(x, y_0); \\ &h_B(x_0, y') = \max_Y h_B(x_0, y)\}. \end{aligned}$$

Образ отображения φ состоит из пар наилучших ответов игроков на стратегии y_0 и x_0 соответственно. Функции h_A и h_B как математические ожидания выигрышей в си-

туации (x, y) билинейны по x и y , и следовательно, образ $\varphi(x_0, y_0)$ ситуации (x_0, y_0) при отображении φ представляет собой выпуклое компактное подмножество в $X \times Y$. Более того, если последовательность пар

$$\{(x_0^n, y_0^n)\}, (x_0^n, y_0^n) \in X \times Y$$

и

$$\{(x'_n, y'_n)\}, (x'_n, y'_n) \in \varphi(x_0^n, y_0^n)$$

имеют предельные точки, т. е.

$$\begin{aligned} \lim_{n \rightarrow \infty} (x_0^n, y_0^n) &= (x_0, y_0); \\ \lim_{n \rightarrow \infty} (x'_n, y'_n) &= (x', y'); \end{aligned}$$

то в силу билинейности функций h_A и h_B и компактности множеств X и Y имеем, что $(x', y') \in \varphi(x_0, y_0)$. Тогда по теореме Какутани (см. п. 8. 2. 2) существует ситуация $(x^*, y^*) \in X \times Y$, для которой $(x^*, y^*) \in \varphi(x^*, y^*)$, т. е.

$$\begin{aligned} h_A(x, y^*) &\leq h_A(x^*, y^*), \\ h_B(x^*, y) &\leq h_B(x^*, y^*), \end{aligned}$$

для всех $x \in X$ и $y \in Y$.

Теорема доказана.

Согласно приведенной теореме точка равновесия существует, возникает вопрос: как ее найти? Рассмотрим ситуацию равновесия на примере 2×2 биматричной игры. Платежные матрицы в этом случае имеют вид:

$$A = \begin{pmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{21} \\ b_{21} & b_{22} \end{pmatrix}.$$

Обозначим p и q — вероятности выбора игроками своих чистых стратегии x_1 и y_1 соответственно, тогда ожидаемые выигрыши для первого и второго игрока запишутся в виде

$$\begin{aligned} h_A(p, q) &= a_{11}pq + a_{12}p(1-q) + a_{21}(1-p)q + a_{22}(1-p)(1-q), \\ h_B(p, q) &= b_{11}pq + b_{12}p(1-q) + b_{21}(1-p)q + b_{22}(1-p)(1-q), \end{aligned}$$

а ситуация равновесия будет определяться парой чисел (p^*, q^*) .

Теорема. Выполнение неравенств

$$h_A(p, q^*) \leq h_A(p^*, q^*), h_B(p^*, q) \leq h_B(p^*, q^*)$$

равносильно выполнению следующих неравенств

$$\begin{aligned} h_A(0, q^*) &\leq h_A(p^*, q^*), h_B(p^*, 0) \leq h_B(p^*, q^*); \\ h_A(1, q^*) &\leq h_A(p^*, q^*), h_B(p^*, 1) \leq h_B(p^*, q^*). \end{aligned}$$

Доказательство теоремы достаточно очевидно в силу билинейности функций $h_A(p, q)$ и $h_B(p, q)$. Действительно

$$\begin{aligned} h_A(p, q^*) &= ph_A(1, q^*) + (1-p)h_A(0, q^*), \\ h_B(p^*, q) &= qh_B(p^*, 1) + (1-q)h_B(p^*, 0). \end{aligned}$$

Опираясь на приведенную теорему легко показать [37], что для того, чтобы точка (p, q) определяла ситуацию равновесия, необходимо и достаточно одновременное выполнение неравенств

$$\begin{aligned} (p-1)(cq-\alpha) &\geq 0, p(cq-\alpha) \geq 0, \\ (q-1)(dp-\beta) &\geq 0, q(dp-\beta) \geq 0, \end{aligned}$$

где

$$\begin{aligned} c &= a_{11} - a_{12} - a_{21} + a_{22}, \alpha = a_{22} - a_{12}, \\ d &= b_{11} - b_{12} - b_{21} + b_{22}, \beta = b_{22} - b_{21}. \end{aligned}$$

при $p \in [0, 1], q \in [0, 1]$.

Пример. Дилемма узников. Два узника находятся в камере предварительного заключения по подозрению в совершении преступления. Следствие, при отсутствии явных улик, опирается в основном на показания заключенных. Если оба подозреваемых будут молчать, то наказанием будет лишь срок предварительного заключения (потери составят -1). Если оба заключенных дают показания, то, учитывая признание как смягчающее обстоятельство, получают небольшой срок (потери составят -6). Если один дает показания, а второй молчит, то вся вина ложится на второго (его потери составят -9), а первый отпускается на свободу (его потери равны нулю). Матрицы выигрышей имеют вид

$$A = \begin{pmatrix} -1 & -9 \\ 0 & -6 \end{pmatrix}, B = \begin{pmatrix} -1 & 0 \\ -9 & -6 \end{pmatrix}.$$

Вычислим $c = 2$, $\alpha = 3$, $d = 2$, $\beta = 3$. Тогда

$$\begin{aligned} (p-1)(2q-3) &\geq 0, \\ p(2q-3) &\geq 0 \end{aligned} \quad (1)$$

$$\begin{aligned} (q-1)(2p-3) &\geq 0, \\ q(2p-3) &\geq 0 \end{aligned} \quad (2)$$

Откуда:

$$1.1 : p = 1, q \geq \frac{3}{2};$$

$$1.2 : p = 0, q \leq \frac{3}{2};$$

$$1.3 : 0 < p < 1, q = \frac{3}{2}.$$

$$2.1 : q = 1, p \geq \frac{3}{2};$$

$$2.2 : q = 0, p \leq \frac{3}{2};$$

$$2.3 : 0 < q < 1, p = \frac{3}{2}.$$

Полученные зигзаги для первой и второй системы неравенств изображены на рисунке 69.

Единственная равновесная ситуация достигается в точке пересечения кривых $(0, 0)$. Это ситуация, в которой каждый из игроков выбирает вторую чистую стратегию — давать показания и его выигрыш составляет -6 . Отклонение игроков от их оптимальных стратегий не дает им никаких преимуществ, однако отклонение обоих позволяет получить выигрыш больший, чем в ситуации равновесия. Например, в точке $(1, 1)$, когда оба молчат, потери каждого составят лишь -1 . Однако по условию игры сговор игроков невозможен.

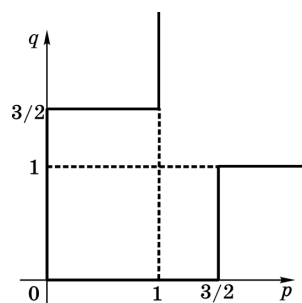


Рис. 69
Точка равновесия
для примера
«Дилемма узников»

Решение биматричных игр с числом стратегий ≥ 3 представляет большие трудности. Рассмотрим пример решения игр $m \times n$ в случае, если известны носители

смешанных стратегий. Носителем называют стратегии, вероятность которых > 0 . Как и для парных антагонистических игр для равновесия Нэша в некооперативных играх n лиц справедлива теорема о носителях (аналогичная доказанной в п. 8.2.2 теореме об активных стратегиях). Она утверждает, что в равновесии игрокам безразлично, какую чистую стратегию, входящую в носитель смешанной оптимальной стратегии, использовать. В то же время использование любой стратегии, не входящее в носитель, приводит к тому, что ожидаемый выигрыш i -го игрока не увеличится. Простое доказательство этого можно найти в [39]. Этот результат позволяет найти вероятности использования стратегий игроками в равновесии, если известны активные стратегии — *носитель*.

Пример. Определить равновесие Нэша во вполне смешанных стратегиях в биматричной игре (табл. 53):

Таблица 53

	N1	N2	N3
M1	-1 -1	3 -5	-3 1
M2	-6 -6	-2 10	2 0
M3	2 4	-4 0	0 -4

Решение. Вполне смешанной называют смешанную стратегию, носитель которой совпадает с множеством всех чистых стратегий игрока. Следовательно, во вполне смешанном равновесии E для любой чистой стратегии l_i игрока i должны быть выполнены условия-равенства выигрышей

$$h_i(l_i, E_{-i}) = h_i(E).$$

Обозначим через h_1 и h_2 ожидаемые выигрыши первого и второго игроков в равновесии. Тогда при использовании вторым игроком своей равновесной смешанной стратегии q , выигрыш на любой чистой стратегии игрока составит:

- для стратегии M_1 : $-q_1 + 3q_2 - 3q_3 = h_1$,
- для стратегии M_2 : $-6q_1 - 2q_2 + 2q_3 = h_1$,
- для стратегии M_3 : $2q_1 - 4q_2 = h_1$.

Кроме этого учтем, что

$$q_1 + q_2 + q_3 = 1.$$

Решая систему четырех уравнений с четырьмя неизвестными, получаем равновесную стратегию второго игрока и ожидаемый выигрыш первого $q = (0,2, 0,3, 0,5)$, $h_1 = -0,8$.

Аналогично, рассматривая второго игрока, можно получить следующую систему:

$$\begin{aligned} -p_1 - 6p_2 + 4p_3 &= h_2, \\ -5p_1 + 10p_2 &= h_2, \\ p_1 - 4p_3 &= h_2, \\ p_1 + p_2 + p_3 &= 1. \end{aligned}$$

Решение данной системы $p = (0,52, 0,20, 0,28)$, $h_2 = -0,6$ представляет собой равновесную стратегию первого игрока и ожидаемый выигрыш второго.

Когда носитель неизвестен, либо требуется найти все равновесия надо осуществить перебор различных носителей. Другой путь — проанализировать наилучшие ответы игроков. Для игр $2 \times n$ и $m \times 2$ для такого анализа можно использовать геометрическую интерпретацию.

Примеры решения таких игр можно найти, например, в [39].

8.3.2. ЗАДАЧА О СДЕЛКАХ

Рассмотрим теперь второй случай, когда допускается кооперирование между игроками. В результате кооперирования возникает некоторое множество исходов $S \subset \mathbf{R}^2$, называемое *допустимым множеством*. Игроки, действуя совместно, могут получить выигрыши $(u, v) \in S$. Очевидно, что вступая в коалицию, стороны рассчитывают получить больше, чем смогли бы получить при односторонних действиях, т. е. сделка имеет смысл, если в результате игроки получают выигрыши (u, v) больше, чем их гарантиро-

ванные выигрыши (u^*, v^*) , определяемые максиминными стратегиями:

$$\begin{aligned} u^* &= \max_x \min_y h_A(x, y), \\ v^* &= \max_x \min_y h_B(x, y). \end{aligned}$$

Пусть дана тройка (S, u^*, v^*) . Необходимо найти правило, которое приписывает этой тройке некоторую точку $(\bar{u}, \bar{v}) \in S$, удовлетворяющую обоим игрокам:

$$(\bar{u}, \bar{v}) = \phi(S, u^*, v^*).$$

Чтобы определить функцию ϕ , необходимо ограничить класс искомых функций. С этой целью Джоном Нэшем сформулированы аксиомы, которым должна удовлетворять любая такая функция.

Аксиома 1 (индивидуальная разумность). $(\bar{u}, \bar{v}) \geq (u^*, v^*)$.

Аксиома 2 (допустимость). $(\bar{u}, \bar{v}) \in S$.

Аксиома 3 (оптимальность). Если $(u, v) \in S$ и $(u, v) \geq (\bar{u}, \bar{v})$, то $(u, v) = (\bar{u}, \bar{v})$.

Аксиома 4 (независимость от посторонних альтернатив). Если $(\bar{u}, \bar{v}) = \phi(S, u^*, v^*)$ и $(\bar{u}, \bar{v}) \in T \subset S$, то $(\bar{u}, \bar{v}) = \phi(T, u^*, v^*)$.

Аксиома 5 (независимость от линейного преобразования). Пусть T получается из S с помощью линейного преобразования $u' = \alpha_1 u + \beta_1$, $v' = \alpha_2 v + \beta_2$. Тогда если $\phi(S, u^*, v^*) = (\bar{u}, \bar{v})$, то $\phi(T, \alpha_1 u^* + \beta_1, \alpha_2 v^* + \beta_2) = (\alpha_1 \bar{u} + \beta_1, \alpha_2 \bar{v} + \beta_2)$.

Аксиома 6 (симметрия). Предположим, что S такое, что $(u, v) \in S \Leftrightarrow (v, u) \in S$. Пусть также $u^* = v^*$ и $\phi(S, u^*, v^*) = (\bar{u}, \bar{v})$, тогда $\bar{u} = \bar{v}$. Решение задачи о сделках (\bar{u}, \bar{v}) для этой системы аксиом позволяет получить следующая теорема.

Теорема. Существует единственная функция ϕ , определенная для всех задач о сделках (S, u^*, v^*) и удовлетворяющая аксиомам **A1–A6**.

Доказательство теоремы мы не рассматриваем, поскольку это заняло бы слишком много места, а направляем читателя к [34].

Отметим только, что доказательство опирается на следующие две полезные леммы, которые позволяют полу-

чить более конструктивный подход к решению задачи о сделках.

Лемма 1. Если существуют такие точки $(u, v) \in S$, что $u > u^*$, $v > v^*$, то существует единственная точка (\bar{u}, \bar{v}) , максимизирующая функцию

$$g(u, v) = (u - u^*)(v - v^*) \quad (8.9)$$

на подмножестве множества S , для которого $u \geq u^*$ (либо $v \geq v^*$).

Доказательство. По предположению, это подмножество S компактно. Так как функция g непрерывна, то она должна достигать на нем своего максимума. Также по предположению, этот максимум μ положителен. Предположим теперь, что существуют две точки (u', v') и (u'', v'') , которые максимизируют $g(u, v)$. Так как $\mu > 0$, то не может быть $u' = u''$ потому, что из этого равенства следовало бы равенство $v' = v''$. Предположим, что $u' < u''$, это значит $v' > v''$.

Так как S выпукло, то $(\hat{u}, \hat{v}) \in S$, где

$$\hat{u} = \frac{u' + u''}{2}, \quad \hat{v} = \frac{v' + v''}{2}.$$

Но

$$g(\hat{u}, \hat{v}) = \frac{(u' - u^*)(v'' - v^*)}{2} + \frac{(u'' - u^*)(v' - v^*)}{2} + \frac{(u' - u'')(v'' - v')}{4}.$$

Далее, каждое из первых двух слагаемых в последнем выражении равно $\mu/2$, но третье слагаемое положительно, а это противоречит тому, что μ — максимум функции g . Таким образом, точка (\bar{u}, \bar{v}) , которая максимизирует g единственна.

Лемма 2. Пусть S , (u^*, v^*) и (\bar{u}, \bar{v}) имеют тот же смысл, что и в предыдущей лемме и пусть

$$h(u, v) = (\bar{v} - v^*)u + (\bar{u} - u^*)v.$$

Тогда если $(u, v) \in S$, то имеет место неравенство $h(u, v) \leq h(\bar{u}, \bar{v})$.

Доказательство. Предположим, что существует такая точка $(u, v) \in S$, что $h(u, v) > h(\bar{u}, \bar{v})$. Пусть $0 < \varepsilon < 1$. В силу выпуклости S имеем $(u', v') \in S$, где $u' = \bar{u} + \varepsilon(u - \bar{u})$ и $v' = \bar{v} + \varepsilon(v - \bar{v})$. В силу линейности $h(u - \bar{u}, v - \bar{v}) > 0$. Но

$$g(u', v') = g(\bar{u}, \bar{v}) + \varepsilon h(u - \bar{u}, v - \bar{v}) + \varepsilon^2 (u - \bar{u})(v - \bar{v}).$$

При $\varepsilon \rightarrow \infty$ последним слагаемым можно пренебречь, поэтому $g(u', v') > g(\bar{u}, \bar{v})$. Но это противоречит максимальнойности $g(\bar{u}, \bar{v})$.

Таким образом, согласно **лемме 1**, если граница множества S гладкая в точке (\bar{u}, \bar{v}) , то прямая, на которой функция $h(u, v)$ постоянна, является касательной к S в этой точке. Наклон границы множества S в произвольной точке представляет собой отношение, в котором дополнительный выигрыш может передаваться от одного игрока к другому. Т. е. согласно схеме Нэша дополнительный выигрыш должен делиться между игроками в таком же отношении, в каком он может передаваться.

Пример 1. Двоим людям дают 100 рублей, если они смогут их разделить. Пусть первый очень богат, у второго всего 100 рублей. Предполагается, что полезность суммы денег пропорциональна ее логарифму. Как поделить деньги?

Решение. Полагаем, конечно, что $u^* = v^* = 0$. Для первого игрока полезность суммы денег x , где $x \leq 100$ будет $u = \log(A + x) - \log A$. Эта величина при больших A пропорциональна x . Тогда можно положить, что $u = x$.

Итак, если первый получит x , то у второго будет $200 - x$, поэтому полезность второго, выраженная через полезность первого, будет

$$v = \log(200 - u) - \log(100).$$

Таким образом, множество S будет выпуклой оболочкой точки $(0, 0)$ и дуги кривой, задаваемой уравнением

$$v = \log \frac{200 - u}{100} \quad (\text{рис. 70}).$$

Ищем точку в S , которая максимизирует функцию $g = uv$, т. е. такое u , которое максимизирует $g = u \log \frac{200 - u}{100}$.

После дифференцирования и приравнивания нулю производной получаем уравнение

$$\frac{u}{200 - u} = \log \frac{200 - u}{100}.$$

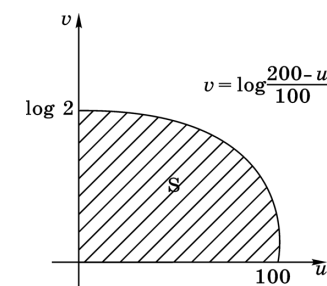


Рис. 70
Множество S

Решая его, получим: $u \approx 54,4$. Т. е. первый игрок должен получить 54,40 рублей, а второй игрок — 45,60 рублей. В некотором смысле это странный результат (богатый получает больше бедного). Решение учитывает, что фактически полезность денег у второго игрока убывает быстро, а у первого — медленно. В результате второй игрок стремится получить хоть что-то и при сделке может уступить первому игроку.

Пример 2. Игра «семейный спор».

Муж и жена каждый вечер решают проблему развлечения: пойти на Футбол или пойти на Балет. Муж любит Футбол и терпеть не может Балет. Жена любит Балет и не любит Футбол, но посещение этих зрелищ в одиночку не дает полного удовлетворения. Поэтому, если они вместе, то получают удовольствие на 4 единицы на любимом зрелище и 1 единицу на нелюбимом, а если поодиночке, — то 2 единицы на любимом и 0 на нелюбимом. Платежные матрицы имеют вид таблица 54 (левое число — выигрыш жены, правое — выигрыш мужа):

Таблица 54

		Жена	
		Балет	Футбол
Муж	Балет	(4; 1)	(0; 0)
	Футбол	(2; 2)	(1; 4)

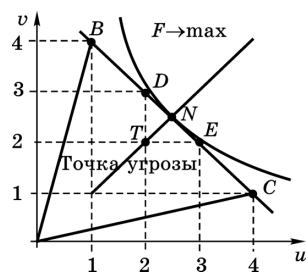


Рис. 71
Геометрическая иллюстрация решения игры «Семейный спор»

В случае двух игроков выигрыш в игре — это пара выигрышей, представляющих общие выигрыши. Если игроки будут перебирать всевозможные стратегии (чистые и смешанные), то все пространство выигрышей в игре образует некоторое множество точек, ограниченное некоторой линией. Множество возможных выигрышей — треугольник с верши-

нами, соответствующими чистым стратегиям (рис. 71). На рисунке можно указать точку с координатами выигрышей игроков, которые они могут гарантировать себе, не прибегая к договоренностям $T = (T_1, T_2)$. Такая точка называется точкой угрозы. Определяя максиминные стратегии Мужа и Жены, получаем $T = (T_1, T_2) = (2, 2)$.

На множестве выигрышей можно найти множество Парето-оптимальных решений. Они являются самыми правыми на горизонтали и самыми верхними на вертикали.

Множество Парето — точки на стороне BC.

Точки Парето-оптимального множества, находящиеся одновременно выше и правее точки угрозы T , образуют переговорное множество.

Очевидно, что игрокам выгодно договариваться, если их выигрыш будет лучше, чем тот, который получается в точке T . Переговорное множество — точки на DE. На переговорном множестве выделяется точка N — решение Нэша (равновесная точка), в которой достигается максимум произведения превышения выигрышей над величинами T_1, T_2

$$F = (u - T_1)(v - T_2) \rightarrow \max.$$

Решение Нэша — точка N , получается как точка касания линий постоянного уровня величины F и границы DE при $F \rightarrow \max$. Очевидно, что максимум F достигается при $u = v = 2,5$. Это решение означает, что супруги договариваются половину вечеров проводить вместе на Футболе, а половину — на Балете.

Следовательно, решение такой игры дает супругам выигрыш (2,5; 2,5).

8.4. ИГРЫ МНОГИХ ЛИЦ

Игры n лиц (при $n \geq 3$), как и в случае игр двух лиц с произвольной суммой, имеют две альтернативы: когда правила игры допускают образование коалиций и когда любая кооперация игроков запрещена.

Рассмотрим некооперативные игры n лиц.

8.4.1. НЕКООПЕРАТИВНЫЕ ИГРЫ МНОГИХ ЛИЦ

Некооперативная теория игр n лиц существенно не отличается от случая биматричных игр. Аналогично биматричным играм здесь рассматриваются смешанные стратегии игроков, и основным вопросом является существование ситуации равновесия.

Пусть дана игра n лиц в нормальной форме. i -й игрок имеет в своем распоряжении стратегии x_i из множества X_i , $i = \overline{1, n}$. Каждый из игроков выбирает стратегию независимо от остальных участников игры, в результате чего возникает набор стратегий $x = (x_1, x_2, \dots, x_n)$, называемый *ситуацией*. У каждого игрока имеется функция выигрыша $h_i(x_1, x_2, \dots, x_n)$, $i = \overline{1, n}$, определенная на множестве ситуаций.

Важнейшим принципом оптимального поведения в данной ситуации считается понятие равновесия по Нэшу.

Ситуация $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ *равновесна* или является *ситуацией равновесия* (по Нэшу), если для любого $i = \overline{1, n}$ и любого $x_i \in X_i$ справедливо неравенство

$$h_i(x_1^*, \dots, x_{i-1}^*, x_i x_{i+1}^*, \dots, x_n^*) \leq h_i(x_1^*, x_2^*, \dots, x_n^*). \quad (8.10)$$

Другими словами, ситуация равновесна, если ни один игрок не имеет никаких разумных оснований для изменения своей стратегии при условии, что все остальные игроки собираются придерживаться своих стратегий. Это обстоятельство делает игру весьма устойчивой.

Возникает вопрос о существовании ситуации равновесия. Теория утверждает, что любая конечная игра с полной информацией имеет ситуацию равновесия [34].

Теорема. Любая конечная бескоалиционная игра n лиц в нормальной форме имеет ситуацию равновесия в смешанных стратегиях.

Доказательство. Множество смешанных стратегий X_i , $i = \overline{1, n}$ каждого игрока является выпуклым многогранником в соответствующем евклидовом пространстве. Поэтому множество всех ситуаций в смешанных стратегиях определяется как декартово произведение $X = X_1 \times X_2 \times \dots \times X_n$. X — компактное выпуклое множество.

Пусть ϕ — многозначное отображение: $\phi: X \rightarrow X$, определяемое соотношением:

$$\phi(x'_1, \dots, x'_n) = \begin{cases} x'' = (x''_1, \dots, x''_n): h_i(x'_1, \dots, x''_i, \dots, x'_n) = \\ = \max_{x_i \in X_i} h_i(x'_1, \dots, x_i, \dots, x'_n), \quad i = \overline{1, n} \end{cases}.$$

Другими словами, образ отображения ϕ состоит из наилучших ответов игроков на стратегии остальных игроков.

Функции $h_i(x)$ как математические ожидания выигрышей в ситуации $x = (x_1, x_2, \dots, x_n)$ полилинейны по x_i . Поэтому образ $\phi(x)$ представляет собой выпуклое компактное подмножество в X . Более того, если последовательности $\{x^k\}$ и $\{x'_k\}$, $x_k \in X$, имеют предельные точки, т. е.

$$\lim_{k \rightarrow \infty} x^k = x, \quad \lim_{k \rightarrow \infty} x'^k = x',$$

то в силу полилинейности функции $h_i(x)$ и компактности множеств X_i имеем: $x' \in \phi(x)$. Тогда по теореме Какутани существует ситуация $x^* = (x_1^*, x_2^*, \dots, x_n^*) \in X$, для которой $x^* \in \phi(x^*)$, т. е.

$$h_i(x_1^*, \dots, x_i, \dots, x_n^*) \leq h_i(x_1^*, \dots, x_i^*, \dots, x_n^*),$$

для всех $x_i \in X_i$, $i = \overline{1, n}$.

Несмотря на фундаментальные результаты последней теоремы, проблема нахождения равновесия в играх n лиц по-прежнему актуальна. Отыскания ситуации равновесия при рассмотрении игр n лиц существенно усложняется по сравнению с бинарными играми.

Пример. На пастбище ограниченной площади могут пастись коровы трех независимых хозяев: A , B и C . Известно количество молока, которое дает каждая корова, если на пастбище пасутся n коров. Каждый хозяин намерен пасти на этом пастбище 1 или 2 коровы. Из опыта известно, что если на пастбище окажутся 3 коровы, каждая даст по 15 л, если 4, то по 14 л, если 5, то по 12 л, а если 6, то всего по 7 л. Определить оптимальные стратегии поведения хозяев.

Решение. Ситуация представляет собой игру трех лиц A , B , C . У каждого две стратегии: привести 1 корову или 2. Каждый хочет получить наибольшее количество

молока, но боится, что если другие хозяева приведут много коров, то его результат может быть небольшим. Здесь выигрыш a_{ijk} — это количество молока, которое получит хозяин A , если A применит стратегию i , B применит стратегию j , а C применит стратегию k . Аналогично для B будут выигрыши b_{ijk} , а для C величины c_{ijk} . Построим таблицу этих выигрышей. В каждой клетке таблицы записаны выигрыши (слева–направо) для A , для B и для C .

Таблица 55

BC A	$i = 1$			$i = 2$		
	$j = 1$	$j = 2$	$j = 3$	$j = 1$	$j = 2$	$j = 3$
$i = 1$	15 15 15	14 14 28	14 28 14	12 24 24		
$i = 2$	28 14 14	24 12 24	24 24 12	14 14 14		

Поясним вычисление выигрышей на примере выделенной клетки. A привел 2 коровы, B и C по одной. Всего 4 коровы, следовательно каждая даст 14, поэтому A получит 28, а B и C по 14. Применим принцип гарантированного результата и вычислим гарантированные выигрыши игроков

$$u_A^* = \max_i \min_{j,k} a_{ijk} = \max\{12; 14\} = 14; \quad i = 2;$$

$$u_B^* = \max_j \min_{i,k} b_{ijk} = \max\{12; 14\} = 14; \quad j = 2;$$

$$u_C^* = \max_k \min_{i,j} c_{ijk} = \max\{12; 14\} = 14; \quad k = 2.$$

Таким образом, решение (2, 2, 2), дающее игрокам по 14 л является равновесным в чистых стратегиях. Легко проверить вышеприведенное свойство равновесия по Нэшу. Если два хозяина приведут по две коровы, а третий одну, то он получит только 12 л. Из таблицы видно, что если от оптимальной стратегии отклонятся сразу два игрока, то условие равновесия не выполняется (можно получить и больше 14 л). Пример показывает, что несогласованные действия игроков здесь очень невыгодны. Действительно, договорившись, хозяева могут привести по одной корове и уже получают больший выигрыш (15 л). Наилучший результат в условиях кооперации между хо-

зяевами будет, если один приведет 1 корову, а двое других по 2 — они получают вместе 60 л, т. е. могут получить по 20 л каждый (!). Таким образом, кооперация — важнейший способ улучшения решений игр.

8.4.2.

КООПЕРАТИВНАЯ ТЕОРИЯ ИГР МНОГИХ ЛИЦ

Объединение игроков в коалицию подразумевает осуществление ими некоторых совместных действий с целью достижения общей цели. В совместных действиях можно выделить несколько уровней взаимодействия [10]:

- обмен информацией о ходе игры;
- совместный выбор стратегий на основе общей договоренности и взаимной информированности;
- объединение активных средств (ресурсов) с соответствующей координацией предпринимаемых действий.

Важным условием для образования устойчивой коалиции является тот факт, что игроки, входящие в коалицию должны находиться в некотором *равновесном* или *устойчивом* состоянии. Помимо объединения интересов важным фактором в кооперативных играх являются *побочные платежи*, которыми игроки могут обмениваться в процессе игры (например, плата за участие в коалиции, штраф за выход из коалиции или нежелание сотрудничать). Побочные платежи изменяют значение выигрышей, предусмотренных исходными правилами игры, и, следовательно, становятся частью используемых игроками стратегий.

Пример. Пусть имеется три игрока A , B , C . Правила допускают образование коалиций из двух игроков. Если такая коалиция возникает, то игрок, не вошедший в коалицию, платит каждому из игроков, образующих коалицию по 10 рублей. Не имея дополнительных сведений о личностях игроков можно выделить три равновозможных исхода игры, выраженных платежами

$$(10, 10, -20), (10, -20, 10), (-20, 10, 10).$$

Теперь положим, что если игрок A входит в коалицию, то получает 15 рублей, а другой игрок, вошедший в коа-

лицию — 5 рублей. Казалось бы, положение игрока улучшилось, теперь он получает больше остальных при тех же условиях. Однако на самом деле коалиция с игроком A становится практически невозможной, поскольку игроки B и C оказываются в выигрыше, образовав совместную коалицию. В этом случае игроку A придется искать пути для образования коалиций, не предусмотренные исходными правилами, например, выплатив игроку B часть выигрыша.

Этот пример иллюстрирует идею побочных платежей.

Обозначим множество всех игроков игры n лиц через $N = \{1, 2, \dots, n\}$. Любое непустое подмножество множества N , включая само множество N , а также любое одноточечное подмножество $\{i\}$ называется *коалицией*.

В игре n лиц общее число возможных коалиций

$$\sum_{k=1}^n C_n^k = 2^n - 1.$$

Характеристической функцией игры n лиц называется вещественнозначная функция γ , определенная на подмножествах множества N , которая ставит в соответствие $\forall S \subset N$ максимумное значение (для S) игры двух лиц, которую разыграли бы игроки S и N/S , т. е. значение характеристической функции есть минимальный гарантированный выигрыш для коалиции S .

Свойства характеристической функции:

$$\begin{aligned} \gamma(\emptyset) &= 0; \\ \gamma(S \cup T) &\geq \gamma(S) + \gamma(T), \end{aligned}$$

если $S \cap T = \emptyset$ (свойство *супераддитивности*).

В свойствах отражена идея полезности кооперирования. Функция γ , удовлетворяющая указанным свойствам, определяет игру n лиц в форме *характеристической функции* (игру γ) [34].

Необходимо найти устойчивые коалиции, обеспечивающие своим участникам наибольший дополнительный выигрыш, по сравнению с индивидуальными действиями.

После того, как та или иная коалиция сформирована и игра сыграна, возникает вопрос о разделении общего выигрыша $\gamma(N)$.

Дележом для игр n лиц с характеристической функцией γ называется вектор $x = (x_1, \dots, x_n)$, удовлетворяющий условиям:

$$\sum_{i \in N} x_i = \gamma(N), \quad (8.11)$$

$$x_i \geq \gamma(\{i\}), \forall i \in N. \quad (8.12)$$

Первое условие выражает желательное свойство решений — *эффективность*, которое следует из предположения о том, что на оптимальном решении не должны происходить потери ресурсов. Это означает, что выигрыш, получаемый коалицией, должен полностью распределяться между ее членами. Второе условие получило название *индивидуальной рациональности* и выражает то, что игрок будет вступать в коалицию только тогда, когда предлагаемый ему выигрыш x_i превышает то, что он получит, действуя отдельно.

Таким образом, решение игры сводится к поиску дележа или множества дележей, наилучших в некотором смысле. Пусть $E(\gamma)$ — множество всех дележей. Из супераддитивности следует, что

$$\sum_{i \in N} \gamma(\{i\}) \leq \gamma(N).$$

В случае, если в последнем выражении достигается равенство, то множество $E(\gamma)$ является одноточечным и проблема поиска решения тривиальна. Эта игра называется *несущественной*. Нас интересуют только существенные игры, т. е. когда

$$\sum_{i \in N} \gamma(\{i\}) < \gamma(N).$$

8.4.3.

ДЕЛЕЖИ. ПРИНЦИП ФОРМИРОВАНИЯ РЕШЕНИЙ

Важная роль в формировании решений принадлежит идее доминирования дележей. Основываясь на них, можно последовательно сравнивать дележи. Пусть $x^1 = (x_1^1, \dots, x_n^1)$, $x^2 = (x_1^2, \dots, x_n^2)$ — дележи. Ясно, что одни игроки предпочтут x^1 , так как $x_i^1 > x_i^2$, другие x^2 , так как $x_i^2 > x_i^1$.

Пусть S — некоторая коалиция. Говорят, что x^1 доминирует x^2 по коалиции S

$$x^1 \succ_S x^2,$$

если

$$\begin{aligned} x_i^1 &> x_i^2 \quad \forall i \in S; \\ \sum_{i \in S} x_i^1 &\leq \gamma(S). \end{aligned}$$

Дележ x^1 доминирует x^2 , если существует хотя бы одна коалиция S , такая что $x^1 \succ_S x^2$.

Естественно, если мы найдем некоторое множество дележей, которое будут доминировать все остальные, тогда в некотором смысле это множество можно считать решением игры. Такое решение не должно допускать так называемой блокирующей коалиции. Говорят, что коалиция S блокирует распределение выигрыша x , если существует достижимое для коалиции S распределение x' , такое, что

$$\forall i \in S; \quad x'_i \geq x_i.$$

Распределение выигрыша, которое блокируется какой-либо коалицией S , не может рассматриваться как удовлетворительное решение задачи, так как игроки из S , договорившись между собой, могут гарантировать себе на распределении x' выигрыш не меньший, чем на распределении x .

Чтобы избежать этого, на распределение выигрыша необходимо наложить дополнительное ограничение — условие отсутствия блокирующей коалиции: не существует коалиции S , блокирующей распределение x . Формально это условие можно записать так

$$\sum_{i \in S} x_i \geq \gamma(S), \quad \forall S \subseteq N. \quad (8.13)$$

Отметим, что упомянутое выше свойство индивидуальной рациональности является частным случаем этого условия, когда коалиция S состоит из одного игрока. Использование условия отсутствия блокирующей коалиции приводит нас к более сильному принципу оптимальности в кооперативной игре — понятию ядра.

8.4.4. С-ЯДРО

Ядром кооперативной игры или *С-решением* (от англ. core) называется множество C недоминируемых дележей.

Таким образом, основу компромисса между конфликтующими сторонами в каких-то случаях составляет поиск ядер.

Теорема. Ядро игры γ есть множество таких дележей x , составляющие которых удовлетворяют следующим условиям:

$$\sum_{i \in S} x_i \geq \gamma(S), \quad \forall S \subseteq N; \quad (a)$$

$$\sum_{i \in N} x_i = \gamma(N). \quad (б)$$

Доказательство [34]. При $S = \{i\}$ условие (а) превращается в

$$x_i \geq \gamma(\{i\}).$$

Вместе с условием (б) это означает, что все такие векторы являются дележами.

Предположим теперь, что x удовлетворяет (а) и (б) и что $y_i > x_i$ при всех $i \in S$. В сочетании с (а) это означает, что

$$\sum_{i \in S} y_i \geq \gamma(S),$$

(и поэтому y не может доминировать x по коалиции S). Следовательно, $i \in C(\gamma)$.

Допустим теперь, что вектор y не удовлетворяет (а) или (б).

Если y не удовлетворяет (б), то он даже не является дележом и потому не может входить в $C(\gamma)$. Пусть теперь существует такое непустое $S \subset N$, что

$$\sum_{i \in S} y_i = \gamma(S) - \varepsilon,$$

где $\varepsilon > 0$. Пусть

$$\alpha = \gamma(N) - \gamma(S) - \sum_{i \in N \setminus S} \gamma(\{i\}).$$

Как легко видеть, из супераддитивности γ следует, что $\alpha \geq 0$. Наконец, пусть s — число элементов в S . Определим z , положив, что

$$z_i = y_i + \varepsilon/s, \text{ если } i \in S, \\ z_i = \gamma(\{i\}) + \alpha/(n-s), \text{ если } i \notin S.$$

Легко видеть, что вектор z является дележом и что $z \succ_s y$. Следовательно, $y \notin C(\gamma)$.

Теорема о ядре показывает, что $C(\gamma)$ представляет собой замкнутое выпуклое множество (как множество допустимых решений задачи линейного программирования (см. главу 2)).

Любой дележ из ядра устойчив, так как ни одна из коалиций не имеет ни желания, ни возможности изменить этот исход игры. В общем случае ядро может содержать более одной точки. Это означает, что устойчивых исходов может быть несколько. Выбор единственного из них остается проблемой. Другим недостатком решения в виде ядра является неопределенность ситуации, когда $C = \emptyset$.

Пример 1. Оркестр из трех музыкантов A, B и C играет в подземном переходе. Вместе они могут получить 100 рублей в час. Играя по двое, они могут получить соответственно: A и B — 30 руб. в час, A и C — 30 руб. в час, B и C — 50 руб. в час. Играя поодиночке, C мог бы зарабатывать 10 руб. в час, остальные не заработали бы ничего. Как распределить заработок?

Формализуем эту задачу в виде кооперативной игры. Характеристической функцией в данном случае будет заработок, который может получить каждая из возможных коалиций игроков:

$$\gamma(\{A, B, C\}) = 100; \gamma(\{A, B\}) = \gamma(\{A, C\}) = 30; \gamma(\{B, C\}) = 50; \\ \gamma(\{A\}) = \gamma(\{B\}) = 0; \gamma(\{C\}) = 10.$$

Для того чтобы распределение выигрыша x принадлежало C -ядру этой игры, оно должно удовлетворять следующим условиям:

- эффективность:
(а) $x_A + x_B + x_C = 100$;

- отсутствие блокирующей коалиции:

$$(б) x_A \geq 0; x_B \geq 0; x_C \geq 10;$$

$$(в) x_A + x_B \geq 30; x_A + x_C \geq 30; x_B + x_C \geq 50.$$

Условия (б) представляют собой условия индивидуальной рациональности игроков. Вместе с условием (а) они дают множество дележей в данной задаче (на рисунке 72 выделено штриховкой), тогда как добавление условий (в), означающих, что никакая двух-

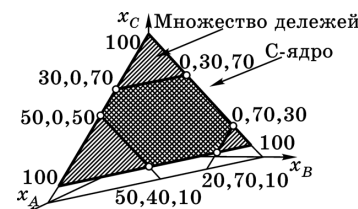


Рис. 72
Множество дележей и C -ядро

элементная коалиция не может заблокировать данное распределение, приводит к C -ядру. Решением данной системы неравенств является многоугольник, выделенный на рисунке 72 двойной штриховкой. Это и есть C -ядро.

Пример 2. Трое игроков решают, каким образом разделить выигрыш в 300 рублей. Каждый игрок может предложить распределение выигрышей, сумма которых не превосходит 300. Если игроки предлагают одинаковое распределение, то они получают предложенные выигрыши. В противном случае игроки получают 0.

Характеристическая функция этой игры имеет вид

$$\gamma(\{1, 2, 3\}) = 300; \gamma(\{1, 2\}) = \gamma(\{1, 3\}) = \gamma(\{2, 3\}) = 0; \\ \gamma(\{1\}) = \gamma(\{2\}) = \gamma(\{3\}) = 0.$$

Записав систему условий для ядра легко получить, что ядро будет совпадать с множеством дележей и иметь вид

$$C = \{(x_1, x_2, x_3) | x_1 + x_2 + x_3 = 300; x_1, x_2, x_3 \geq 0\}.$$

Пример 3. Модифицируем предыдущий пример. Пусть теперь распределение выигрыша принимается большинством голосов, т. е. если любая пара игроков предлагает одинаковое распределение, то все игроки получают его.

Характеристическая функция игры в этом случае изменится и будет выглядеть так:

$$\gamma(\{1, 2, 3\}) = \gamma(\{1, 2\}) = \gamma(\{1, 3\}) = \gamma(\{2, 3\}) = 300; \\ \gamma(\{1\}) = \gamma(\{2\}) = \gamma(\{3\}) = 0.$$

Такой вид функции отражает тот факт, что теперь любые два игрока, объединившись, могут получить весь выигрыш.

Нетрудно видеть, что условия эффективности и индивидуальной рациональности, а следовательно, и множество дележей в этом случае не изменится, а условия отсутствия двухэлементных блокирующих коалиций примут вид

$$x_i + x_j \geq 300; i, j = 1, 2, 3; i \neq j.$$

Легко увидеть, что вместе с условиями *эффективности*

$$x_1 + x_2 + x_3 = 300;$$

и отсутствия блокирующей коалиции

$$x_i \geq \gamma(\{i\}) = 0; i = 1, 2, 3;$$

получаем, что система неравенств несовместна, т. е. *С*-ядро в данной игре пусто.

Последние два примера иллюстрируют два основных недостатка *С*-ядра: оно может содержать большое количество альтернатив, как в примере 2, где оно совпадает с множеством всех дележей, либо не существовать вовсе, как в примере 3.

8.4.5. РЕШЕНИЕ НЕЙМАНА–МОРГЕНШТЕРНА

Возможная пустота *С*-ядра привела к необходимости поиска других принципов оптимальности. Исторически первой из них возникла концепция решения игры, сформулированная Дж. фон Нейманом и О. Моргенштерном — решение по Нейману–Моргенштерну (*НМ-решение*). Используется та же идея доминирования дележей, но на уровне множеств. Предполагается существование некоторого множества X , элементы которого $x \in X$ представляют собой равноценные дележи (доминирование между ними исключено), но вместе с тем, каждый дележ $y \notin X$ доминируется хотя бы одним $x \in X$.

Определение. Распределение выигрыша x доминирует распределение выигрыша y (обозначается $x \succ y$), если най-

дется коалиция S , такая, что x и y — достижимые распределения для S и выполнено

$$\begin{aligned} \forall i \in S \\ x_i > y_i. \end{aligned}$$

Определение. Множество распределений выигрыша V называется решением игры γ по Нейману–Моргенштерну (*НМ-решением*), если оно удовлетворяет следующим свойствам:

1. *Внутренняя устойчивость.* Для любых $x, y \in V$ не выполнено $x \succ y$.

2. *Внешняя устойчивость.* Для любого $y \notin V$ найдется $x \in V$, такое, что $x \succ y$.

НМ-решение представляет собой более слабый принцип оптимальности, чем *С*-ядро, и существует в гораздо большем числе игр.

В качестве примера отыскания *НМ-решения* вновь рассмотрим изложенную выше задачу распределения 300 рублей между тремя игроками, решение в которой принимается по принципу большинства.

Нетрудно увидеть, что множество распределений выигрыша

$$V = \{(150, 150, 0), (150, 0, 150), (0, 150, 150)\}$$

является *НМ-решением* этой игры.

Действительно, для любой пары распределений из V и для любой коалиции, содержащей не менее двух игроков, найдется игрок, который в обоих случаях получает выигрыш в 150 рублей, а следовательно строгое неравенство $\forall i \in S x_i > y_i$ не может быть выполнено. Для одноэлементных коалиций в этом случае будет нарушаться условие достижимости распределения, так как $\gamma(i) = 0$ для любого i . Следовательно, множество V внутренне устойчиво.

НМ-решения обобщают *С-решения*, т. е., если для некоторой игры $C \neq \emptyset$, и *НМ-решение* существует, то C содержится в нем.

Рассмотренные выше концепции решения кооперативных игр имеют одни и те же недостатки: они представляют собой *множественные* принципы оптимальности, ко-

торые каждой игре ставят в соответствие некоторое множество решений или же эти решения могут не существовать вовсе. Поэтому естественным является желание сформулировать такой принцип оптимальности, который в любой кооперативной игре позволял бы выбрать ровно одно оптимальное решение.

Такие принципы оптимальности приводят к понятию *значение игры*.

8.4.6. ЗНАЧЕНИЕ ИГРЫ. ВЕКТОР ШЕПЛИ

Наибольшее распространение получило значение игры — *вектор Шепли*. Л. Шепли в 1953 году сформулировал набор аксиом, которыми должен обладать данный принцип оптимальности, и предложил правило ϕ , ставящее в соответствие любой кооперативной игре γ распределение выигрыша $\phi(\gamma)$, удовлетворяющее данным аксиомам.

Пусть игра $\pi \gamma$ представляет собой игру γ , в которой изменена нумерация игроков за счет перестановки π . При этом роль любого игрока i в игре γ в точности такая же, как и роль игрока $\pi(i)$ в игре $\pi \gamma$.

Аксиома 1 (Симметричность).

Для любой игры γ , для любой перестановки π и для любого игрока $i \in N$

$$\phi_{\pi(i)}(\pi \gamma) = \phi_i(\gamma).$$

Интуитивный смысл первой аксиомы Шепли состоит в том, что выигрыш игрока зависит только от его роли в игре, а не от номера или расположения во множестве N .

Назовем коалицию $R \subseteq N$ носителем кооперативной игры γ , если $\gamma(S \cap R) = \gamma(S)$, $\forall S \subseteq N$, т. е. носитель игры — это множество игроков, создающих полезность. Игроки, не входящие в носитель, называются нулевыми игроками («болванами»).

Аксиома 2 (Носитель).

Для любой игры γ и для любого носителя R выполнено

$$\sum_{i \in R} \phi_i(\gamma) = \gamma(R).$$

Вторая аксиома Шепли утверждает, что совокупное благосостояние, равное $\gamma(N)$, должно распределяться только между игроками, входящими в носитель игры.

Третьей аксиомой Шепли является требование линейности значения игры.

Аксиома 3 (Линейность).

Для любых двух игр γ и ω , любого числа $0 \leq p \leq 1$ и для любого игрока $i \in N$

$$\phi_i(p\gamma + (1-p)\omega) = p\phi_i(\gamma) + (1-p)\phi_i(\omega).$$

Как показал Шепли, существует единственное отображение ϕ , удовлетворяющее данным аксиомам.

$$\phi_i(\gamma) = \sum_{K: i \in K} \frac{(k-1)!(n-k)!}{n!} \text{inp}(\{i\}, K), \quad (8.14)$$

где k, n — размер множеств K и N соответственно;

$$\text{inp}(\{i\}, K) = \gamma(K \cup \{i\}) - \gamma(K)$$

— вклад игрока i в коалицию $K \cup \{i\}$.

Доказательство этого можно найти, например, в [34], [38].

Формула определяет единственный вектор Шепли, дающий значения для каждого игрока i . Эта формула может быть интерпретирована так. Пусть игроки упорядочиваются случайным образом и последовательно образуются коалиции $\{i_1\}$, $\{i_1, i_2\}$ и т. д. Имеется $n!$ различных упорядочений игроков. Следовательно, если различные упорядочения равновероятны, коэффициенты в формуле представляют собой вероятность того, что коалиция $K \cup \{i\}$ сформируется присоединением игрока i к множеству K . Величина $\text{inp}(\{i\}, K)$ представляет собой дополнительную полезность, приносимую игроком i в образующуюся коалицию. Тогда компонента вектора Шепли $\phi_i(\gamma)$ представляет собой математическое ожидание выигрыша игрока i .

Вектор Шепли широко используется для исследования распределения силы игроков в кооперативной игре.

Во многих играх величина $\text{inp}(\{i\}, K) = \gamma(K \cup \{i\}) - \gamma(K)$ будет либо 1, когда K — выигрывающая коалиция, а коалиция $K \setminus \{i\}$ — не является выигрывающей, либо 0.

Тогда формула вектора Шепли упрощается

$$\varphi_i(\gamma) = \sum_{K: i \in K} \frac{(k-1)!(n-k)!}{n!}, \quad (8.15)$$

где суммирование распространяется на все такие выигрывающие коалиции K , что коалиция $K \setminus \{i\}$ не является выигрывающей.

Пример 1. Четыре акционера одной компании обладают соответственно 10, 20, 30, и 40 акциями. Любое решение принимается простым большинством акций. Найти вектор Шепли, показывающий силу акционеров в компании.

Решение. Это простая игра четырех лиц. Выигрывающими коалициями являются: $\{2, 4\}$, $\{3, 4\}$, $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$ и $\{1, 2, 3, 4\}$. Единственной выигрывающей коалицией K , такой, что $K \setminus \{1\}$ не выигрывающая, является коалиция $\{1, 2, 3\}$. Следовательно, так как $k = 3$

$$\varphi_1 = \frac{2!1!}{4!} = 1/12.$$

Аналогично находим, что коалиции $\{2, 4\}$, $\{1, 2, 3\}$, $\{2, 3, 4\}$ являются выигрывающими, но каждая перестает быть выигрывающей, если из нее устраняется игрок 2. Поэтому $\varphi_2 = 1/12 + 1/12 + 1/12 = 1/4$. Точно таким же образом находим, что $\varphi_3 = 1/4$, а $\varphi_4 = 5/12$. Следовательно вектор Шепли — это вектор $(1/12, 1/4, 1/4, 5/12)$. Заметим, что компоненты вектора Шепли для игроков 2 и 3 равны, хотя игрок 3 имеет гораздо больше акций. Это не удивительно, так как у игрока 3 не больше возможностей для образования выигрывающей коалиции, чем у игрока 2. Точно так же «сила» игрока 4 больше, чем доля его акций, а «сила» игрока 1 меньше его доли акций.

Пример 2. Пусть в игре примера 1 игроки имеют соответственно 10, 30, 30, 40 акций. Тогда вектор Шепли будет $(0, 1/3, 1/3, 1/3)$, т. е. акции первого игрока бесполезны, а избыточные акции игрока 4 не дают ему преимуществ по сравнению с игроками 2 и 3.

В рассмотренных примерах 2 и 3 п. 8.4.4, в которых три игрока должны прийти к соглашению, чтобы разделить 300 рублей, вектор Шепли составляет $(100, 100, 100)$.

Проблема поиска каких-то иных подходов к исследованию кооперативных игр остается открытой. В частности рассматривается дележ с помощью арбитража (сторонний арбитр принимает решение о дележе, опираясь на некоторые правила) [31].

8.5. МНОГОШАГОВЫЕ ИГРЫ

8.5.1. ИГРЫ НА РАЗОРЕНИЕ

Игры на разорение характеризуются тем, что каждый из двух игроков, начиная игру, имеет некоторые ограниченные ресурсы. На каждом шаге ресурсы одного из игроков уменьшаются. Выигрывает тот, кто разорит противника, или тот, кто первым наберет фиксированное количество очков.

Подобные игры целесообразно решать, двигаясь «от конца к началу», при этом каждый шаг можно рассматривать как отдельную матричную игру. После того, как стратегии для этого шага выбраны, выигрыш определяется либо как обычный выигрыш (если многошаговая игра закончена), либо как обязательство разыгрывать следующий шаг игры. Поскольку выигрыш понимается в вероятностном смысле, можно заменить обязательство разыгрывать игру ценой этой игры.

Пример. Игра инспектирования. Игрок A — нарушитель, хочет совершить кражу. Имеется N моментов времени, когда это злодеяние может быть осуществлено. Игрок B — инспектор, хочет предотвратить преступление, однако может провести только одну инспекцию в один из N моментов.

Если нарушитель действует, а инспектор в этот момент не инспектирует, то считаем, что преступление совершилось, выигрыш равен 1. Если нарушитель действует, а инспектор в тот же момент инспектирует, то считаем, что нарушитель пойман, выигрыш равен -1 . Если нарушитель не действует ни в один из моментов времени, то его выигрыш равен 0.

У каждого игрока на каждом шаге две альтернативы: действовать — не действовать.

Матрица для игры для первого шага имеет вид

$$A_N = \begin{pmatrix} -1 & 1 \\ 1 & \Gamma_{N-1} \end{pmatrix},$$

где Γ_{N-1} — обязательство разыгрывать следующий шаг игры, на котором количество оставшихся моментов времени равно $N-1$. Цена этой матричной игры не может быть найдена непосредственно, однако мы знаем цену игры A_0 (т. е. в конечный момент времени): $\gamma_1 = 0$. Заменив обязательство Γ_1 ценой этой игры мы сможем решить матричную игру A_2 (в предпоследний момент времени) и найти γ_2 , которое подставим в A_3 чтобы найти γ_3 и т. д. Двигаясь от конца к началу, мы сможем найти цену игры γ_N в начальный момент времени, которая и будет решением задачи.

На практике вместо последовательного перебора матричных игр решают задачу в общем виде и составляют рекуррентное соотношение с конечным условием.

Определим значение цены игры в начальный момент времени

$$\gamma_N = \text{val} \begin{pmatrix} -1 & 1 \\ 1 & \Gamma_{N-1} \end{pmatrix},$$

где val — цена матричной игры.

Поскольку $\gamma_{N-1} < 1$, матрица A_N не имеет седловой точки, тогда, применив аналитическую формулу решения матричных игр 2×2 , получим:

$$\gamma_N = \frac{\gamma_{N-1} + 1}{-\gamma_{N-1} + 3}$$

с начальным условием $\gamma_1 = 0$. Решим это соотношение, получим

$$\gamma_N = \frac{N-1}{N+1}.$$

Тогда

$$A_N = \begin{pmatrix} -1 & 1 \\ 1 & \frac{N-2}{N} \end{pmatrix}$$

и, решая эту матричную игру при $N \geq 2$, получим оптимальные смешанные стратегии:

$$x^N = \left(\frac{1}{N+1}, \frac{N}{N+1} \right)', \quad y^N = \left(\frac{1}{N+1}, \frac{N}{N+1} \right)'. \quad (8.16)$$

8.5.2. СТОХАСТИЧЕСКИЕ ИГРЫ

Стохастические игры аналогичны играм на разорение, однако в отличие от последних здесь имеется конечное число различных позиций, которые могут повторяться в процессе игры. Теоретически игра может продолжаться бесконечно и иметь бесконечную сумму, однако правила игры включают рандомизацию, гарантирующую, что вероятность бесконечной игры равна нулю и математическое ожидание выигрыша конечно.

Стохастическая игра — набор p игровых элементов или позиций Γ_k . Каждый игровой элемент представляется $m_k \times n_k$ -матрицей, элементы которой имеют вид

$$\alpha_{ij}^k = a_{ij}^k + \sum_{l=1}^p q_{ij}^{kl} \Gamma_l,$$

где

$$q_{ij}^{kl} \geq 0,$$

$$\sum_{l=1}^p q_{ij}^{kl} < 1.$$

Элементы матрицы α_{ij}^k означают, что если в k -м игровом элементе игроки выбирают свои i -ю и j -ю чистые стратегии, то выигрыш первого игрока равен a_{ij}^k и с вероятностью q_{ij}^{kl} игра продолжится в l -м игровом элементе Γ_l . С вероятностью $q_{ij}^{k0} = 1 - \sum_{l=1}^p q_{ij}^{kl} > 0$ игра прекратится.

Полагаем, что при попадании в один и тот же игровой элемент стратегия игрока не меняется (стационарная стратегия). Для каждой отдельной игровой ситуации стратегия игрока есть m_k -вектор, тогда стратегия игрока в стохастической игре есть совокупность из p m_k -векторов, представляющих смешанные стратегии отдельных игровых ситуаций.

Обозначим x^k — стратегию игрока A и y^k — стратегию игрока B , $k = 1, p$.

Аналогично, цена игры есть вектор, состоящий из цен отдельных игровых элементов, при условии, что эти игровые элементы разыгрываются на первом шаге игры. Обозначим цену игры $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_p)$.

Аналогично играм на разорение заменим игровой элемент Γ_k компонентой γ_k вектора цены, т. е.

$$\gamma_k = \text{val} B_k,$$

где $B_k = (b_{ij}^k)$ — $m_k \times n_k$ -матрица игры k -го игрового элемента:

$$b_{ij}^k = a_{ij}^k + \sum_{l=1}^p q_{ij}^{kl} \gamma_l.$$

Поскольку вероятность продолжения игры меньше единицы, то с увеличением числа шагов вероятность продолжения стремится к нулю, вследствие чего можно считать, что после некоторого шага игра останавливается. Таким образом, ее можно аппроксимировать игрой на разорение. При этом можно показать [34], что следующая индуктивная последовательность сходится к истинному вектору цены игры

$$\gamma^0 = (0, 0, \dots, 0),$$

$$b_{ij}^{kr} = a_{ij}^k + \sum_{l=1}^p q_{ij}^{kl} \gamma_l^r,$$

$$r = 1, 2, \dots,$$

$$\gamma_k^{r+1} = \text{val} B_k^r,$$

и при этом

$$tr \leq s \cdot t_{r-1},$$

где

$$t_r = \max_k (|\gamma_k^{r+1} - \gamma_k^r|), \quad s = \max_{k,l,j} \left(\sum_{l=1}^p q_{ij}^{kl} \right). \quad (8.17)$$

Последнее соотношение позволяет оценить количество шагов, необходимых для обеспечения точности приближения вектора решения.

Пример. Игроки A и B имеют вместе пять единиц. На каждом шаге игры игрок A выбирает либо «орел», либо «решку»; игрок B , не зная выбора игрока A делает аналогичный выбор. Если выпадает два «орла», то B платит A три единицы; если две «решки», то A платит B одну единицу; если результаты не совпадают — A платит B две единицы. После каждого шага бросается монета, чтобы определить, продолжать игру или закончить. Игра заканчивается, если один из игроков разоряется. Кроме того, выполняется условие, что ни один игрок не может платить больше, чем он имеет.

Здесь четыре игровых элемента Γ_k , $k = \overline{1,4}$, где k — величина, которую имеет игрок A в начале шага.

$$\Gamma_1 = \begin{pmatrix} 3 + \frac{\Gamma_4}{2} & -1 \\ -1 & 1 + \frac{\Gamma_2}{2} \end{pmatrix}, \quad \Gamma_2 = \begin{pmatrix} 3 & -2 \\ -2 & 1 + \frac{\Gamma_3}{2} \end{pmatrix},$$

$$\Gamma_3 = \begin{pmatrix} 2 & -2 + \frac{\Gamma_1}{2} \\ -2 + \frac{\Gamma_1}{2} & 1 + \frac{\Gamma_4}{2} \end{pmatrix}, \quad \Gamma_4 = \begin{pmatrix} 1 & -2 + \frac{\Gamma_2}{2} \\ -2 + \frac{\Gamma_2}{2} & 1 \end{pmatrix}.$$

Используя индуктивные формулы, найдем значения векторов цены игры

$$\gamma^0 = (0, 0, 0, 0),$$

$$\gamma^1 = (0, 33, -0, 13, -0, 29, -0, 5),$$

$$\dots$$

$$\gamma^4 = (0, 26, -0, 19, -0, 32, -0, 55).$$

Последнее приближение имеет требуемую точность, используя его, найдем оптимальные стратегии игроков. Имеем

$$B_1 = \begin{pmatrix} 2,72 & -1 \\ -1 & 0,91 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 3 & -2 \\ -2 & 0,84 \end{pmatrix},$$

$$B_3 = \begin{pmatrix} 2 & -1,87 \\ -1,87 & 0,72 \end{pmatrix}, \quad B_4 = \begin{pmatrix} 1 & -2,1 \\ -2,1 & 1 \end{pmatrix}.$$

Далее, решая отдельные матричные игры для каждой игровой ситуации, найдем следующие оптимальные стратегии

$$\begin{aligned}x^1 &= (0,34, 0,66), y^1 = (0,34, 0,66); \\x^2 &= (0,38, 0,62), y^2 = (0,38, 0,62); \\x^3 &= (0,4, 0,6), y^3 = (0,4, 0,6); \\x^4 &= (0,5, 0,5), y^4 = (0,5, 0,5).\end{aligned}$$

8.5.3. ДИФФЕРЕНЦИАЛЬНЫЕ ИГРЫ

Обобщением стохастических и рекурсивных игр является игра, в которой промежуток времени между ходами убывает и в пределе получается игра, в которой каждый игрок должен делать ход в каждый момент времени. Ввиду непрерывности совершения ходов игровые элементы меняются незначительно (непрерывно) в течение малых промежутков времени.

Наиболее характерными в классе дифференциальных игр являются игры преследования. Много примеров игр преследования можно привести из военного дела: торпеда и корабль, корабль и подводная лодка, ракета и самолет и т. д.

Будем обозначать преследователя P (от *англ.* pursuer — преследователь), а преследуемого E (от *англ.* evader — убегающий). Соответствующие движущиеся объекты могут управляться человеком или автоматически, но, так или иначе, P и E — разумные противники с противоположными интересами. Рассмотрим пример наиболее простой игры преследования (рис. 73). Пусть C — область расположения объекта, который P защищает от атакующего врага E .

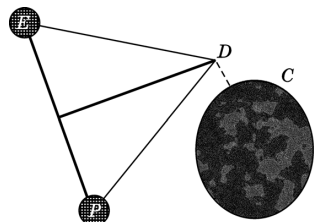


Рис. 73
Геометрическая иллюстрация оптимального решения дифференциальной игры преследования

P и E совершают движение с одинаковой скоростью и начинают двигаться из положения, изображенного на рисунке. Примем для простоты, что захват означает совпадение точек P и E . Платой является расстояние от точки захвата до области C (если захват возможен). P должен максимизировать это расстояние, а E — минимизировать его. Если E может достичь C и захвата не произойдет, то этот исход считается для E наилучшим. Решением этой игры будет ответ на вопрос: как должны двигаться оба игрока.

Приведем некоторые геометрические соображения относительно разумных действий игроков. Проведем перпендикуляр к отрезку PE через его середину. Каждую точку верхней полуплоскости игрок E достигнет раньше, чем P ; для точек нижней полуплоскости — наоборот. Пусть D — ближайшая к C точка перпендикуляра. Оптимальные стратегии игроков состоят в движении к точке D . Здесь происходит захват, и длина пунктирного отрезка есть цена игры.

Пусть P действует оптимально, а E нет; например он решил двигаться по прямой L (рис. 74). В этом случае P всегда направляется в ближайшую к C точку на перпендикуляре (относительно текущего положения P и E). Несколько положений этой точки обозначены D, D', D'', \dots

Длина пунктирных отрезков увеличивается (растет штраф, который платит E за свою неудачную стратегию).

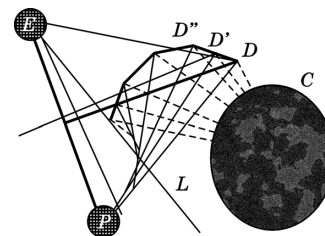


Рис. 74
Геометрическая иллюстрация решения дифференциальной игры преследования: игрок E действует неоптимально

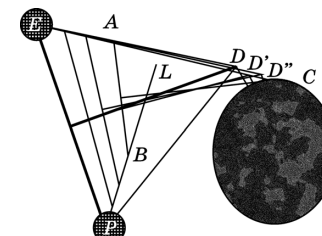


Рис. 75
Геометрическая иллюстрация решения дифференциальной игры преследования: игрок P действует неоптимально

Лучшее что может сделать E в текущий момент — это вернуться к оптимальной стратегии. Поскольку точки D , D' , D'' , ... последовательно являются целями движения P , он описывает криволинейную траекторию до тех пор, пока D не достигнет L . С этого момента точка D остается неподвижной, оба игрока движутся прямолинейно и игра становится оптимальной для обеих сторон.

Пусть теперь E действует оптимально, а P нет (см. рис. 75). P решил двигаться по прямой L . Теперь E должен всегда направляться в движущуюся точку D . Пунктирные отрезки укорачиваются, и когда E находится в точке A , точка D фактически достигает C . Из точки A он движется прямо по направлению к D и действительно достигает C , а P уже ничего не может сделать, чтобы остановить его (P находится в B , когда E находится в A).

Приведенные рассуждения показывают, что оптимальная стратегия игроков обладает свойством устойчивости (равновесия).

Рассмотрение более сложных игр подобного рода требует более строгого теоретического обоснования. Следуя [34], [40], рассмотрим математическую модель дифференциальной игры и условия для получения решения игры в общем виде.

Пусть игровой элемент представляется набором n вещественных чисел $x = (x_1, x_2, \dots, x_n)$, называемых *переменными состояниями*. В момент времени t игрок A выбирает набор p вещественных чисел $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_p)$, называемых *управляющими переменными*. Аналогично игрок B выбирает управляющие переменные $\psi = (\psi_1, \psi_2, \dots, \psi_p)$.

Управляющие переменные влияют на изменение переменных состояния в соответствии с системой дифференциальных уравнений, называемых *кинематическими уравнениями*

$$\dot{x}_i = f_i(x, \varphi, \psi), \quad (8.18)$$

где

$$\dot{x}_i = \frac{dx_i}{dt}.$$

Дифференциальная игра заканчивается, когда переменные состояния достигают границы некоторого замк-

нутого множества $C \subset \mathbb{R}^n$, называемого *терминальной поверхностью*. Например, игрок A приблизился к B достаточно, чтобы поймать его, или же закончился определенный период времени.

Выигрыш может быть *терминальным* и *интегральным*, а также их комбинацией. Если игра начинается в момент $t = 0$ и заканчивается в момент $t = T$ в точке $y = (y_1, y_2, \dots, y_n)$, то терминальный выигрыш есть просто функция $G(y)$, определенная на терминальной поверхности игры, а интегральный выигрыш имеет вид

$$\int_0^T K(x) dt.$$

Как и в случае дискретных многошаговых игр метод решения дифференциальных игр состоит в замене игровых элементов их значениями с последующим решением рекуррентных уравнений для этих значений (они будут дифференциальными).

Обозначим $\gamma(x)$ значение дифференциальной игры, которая начинается в точке $x = (x_1, \dots, x_n)$. Пусть в момент $t = 0$ первый игрок выбирает управляющую переменную $\bar{\varphi}$, а второй — управляющую переменную $\bar{\psi}$. В этом случае после малого интервала времени Δt мы увидим, что переменные состояния будут приближенно равны

$$x + \Delta x,$$

где

$$\Delta x_i = f_i(x; \bar{\varphi}, \bar{\psi}) \Delta t.$$

Если игра имеет интегральный выигрыш, то общий выигрыш будет приближенно равен

$$K(x_1, \dots, x_n) \Delta t.$$

Игра начинается снова из точки $x + \Delta x$ и с уже достигнутым выигрышем. Если, начиная с момента Δt , используются оптимальные стратегии, то общий выигрыш

$$K(x_1, \dots, x_n) \Delta t + \gamma(x + \Delta x).$$

Известно, что

$$\gamma(x + \Delta x) \cong \gamma(x) + \sum_{i=1}^n \frac{\partial \gamma}{\partial x_i} \Delta x_i.$$

Подставляя выражение Δx_i , получаем

$$\gamma(x + \Delta x) \cong \gamma(x) + \sum_{i=1}^n \frac{\partial \gamma}{\partial x_i} f_i(x; \bar{\varphi}, \bar{\psi}) \Delta t.$$

Предполагая, что $\bar{\varphi}$ и $\bar{\psi}$ — оптимальные выборы управляющих переменных в момент $t = 0$, мы имеем

$$\gamma(x) \cong K(x) \Delta x + \gamma(x) + \sum_{i=1}^n \frac{\partial \gamma}{\partial x_i} f_i(x; \bar{\varphi}, \bar{\psi}) \Delta t.$$

Полагая $\Delta t \rightarrow 0$, получаем

$$K(x) + \sum_{i=1}^n \frac{\partial \gamma}{\partial x_i} f_i(x; \bar{\varphi}, \bar{\psi}) = 0,$$

или, что эквивалентно,

$$\max_{\varphi} \min_{\psi} \left(K(x) + \sum_{i=1}^n \frac{\partial \gamma}{\partial x_i} f_i(x; \varphi, \psi) \right) = 0, \quad (8.19)$$

где $\bar{\varphi}$, $\bar{\psi}$ — оптимальные значения управляющих переменных.

Это уравнение известно как *основное уравнение* дифференциальных игр.

Теперь, имея основное уравнение, можно (как и в играх на разорение) двигаться назад вдоль траекторий дифференциальных уравнений с терминальной поверхности (как ранее мы двигались вдоль траектории разностных уравнений). В результате получим уравнения

$$\dot{\gamma}_j = - \left[K_j(x; \bar{\varphi}, \bar{\psi}) + \sum_i \gamma_i(x) f_{ij}(x; \bar{\varphi}, \bar{\psi}) \right], \quad (8.20)$$

которые вместе с системой

$$\dot{x}_j = f_j(x; \bar{\varphi}, \bar{\psi}) \quad (8.21)$$

называются *уравнениями траекторий* дифференциальной игры. Здесь

$$K_j = \frac{\partial K}{\partial x_j}, \quad \dot{\gamma}_j = \frac{d\gamma_j}{dt}, \quad f_{ij} = \frac{\partial f_i}{\partial x_j}.$$

Эти $2n$ уравнений вместе со значением функции G в качестве конечных условий представляют собой формальное решение игры.

Примеры решения некоторых дифференциальных игр с использованием *основного уравнения* можно найти в [34], [40].

ГЛАВА 9

ЭЛЕМЕНТЫ ТЕОРИИ СТАТИСТИЧЕСКИХ ОПТИМАЛЬНЫХ РЕШЕНИЙ

Теория статистических решений чрезвычайно развитая область в экономике, военном деле, в области обработки информации на фоне шумов и т. д. Рассмотрим элементы этой теории как продолжение теории игр.

Существуют задачи, в которых B — «бессознательный игрок», он мешает нам принимать правильные решения, но он не противодействует активно, а действует в соответствии с природными случайными явлениями, поэтому такую ситуацию называют игрой с природой. Например, это помехи в канале связи для передачи информации, шумы при записи или воспроизведении звука и т. д. Ясно, что бессознательное воздействие в целом вредит нам меньше, чем сознательное. С другой стороны, эта «бессознательность» приводит к непредсказуемому поведению операции. Можно считать, что, как и в теории игр, мы боремся с противником, который неосознанно мешает нам. Например, это погодные условия, или случайный спрос при продаже товара. Вот почему в теории статистических решений главной проблемой является обоснование принципов оценки различных ситуаций со стороны A . Заметим, что в теории игр был один принцип — принцип минимакса.

Мы будем рассматривать дискретные альтернативы (стратегии) природы. Тогда, если у A имеется m стратегий, а у «природы» n альтернатив, то может быть получена матрица выигрышей при применении каждой пары (A_i, P_j) (см. табл. 56).

Т а б л и ц а 56

	P_1	P_2	...	P_n
A_1	a_{11}	a_{12}	...	a_{1n}
A_2	a_{21}	a_{22}	...	a_{2n}
...
A_m	a_{m1}	a_{m2}	...	a_{mn}

можно отбросить и оставить только одну, безусловно, лучшую. Столбцы платежной матрицы нельзя отбрасывать, так как условия «природы» могут быть и в нашу пользу. При анализе платежной матрицы можно сделать неверный вывод о качестве нашего решения.

Пусть сравниваются два выигрыша, находящиеся в разных столбцах a_{ij} и a_{kl} , причем $j \neq l$. Если $a_{ij} > a_{kl}$, то вроде бы решение в i -строке лучше, чем решение в k -строке, но так просто можно сравнивать, если выигрыш соответствует одинаковым условиям.

П р и м е р. Пусть в Томской области, приняв определенные управляющие решения, получили урожайность пшеницы 20 центнеров с гектара, а в Краснодарском крае — 25. Эти значения сравнивать нельзя, так как для Томской области это может быть рекордный (наилучший) результат, а для Краснодарского края — плохой, так как рекорд 50. Решение нужно сравнивать с потенциальными возможностями.

Вот почему необходимо преобразовывать платежную матрицу таким образом, чтобы каждый наш выигрыш соотносился с тем максимумом, который можно достигнуть в данных условиях P_j . Для каждого P_j можно найти максимальную величину β_j и вычислить величину $r_{ij} = \beta_j - a_{ij}$, называемую риском. Здесь

$$\beta_j = \max_i a_{ij}.$$

Чем риск меньше, тем лучше.

Условия P_j иногда называются гипотезами. Если платежная матрица построена, то задача состоит в анализе матрицы с целью получения стратегии A_i , которая наиболее выгодна по отношению к другим. В простейшем случае, если какие-то строки матрицы заведомо невыгодны, то их

9.1. ПРИНЯТИЕ РЕШЕНИЙ ПРИ ИЗВЕСТНЫХ АПРИОРНЫХ ВЕРОЯТНОСТЯХ

Будем обозначать вероятности гипотез: $Q_1 = p(P_1)$, $Q_2 = p(P_2)$, ..., $Q_n = p(P_n)$. Таким образом, мы считаем, что вероятности известны до того, как мы принимаем решение. **Р е ш е н и е** — выбор оптимальной стратегии игрока A . Говорят, что это ситуация идеального наблюдателя.

Естественно в качестве критерия использовать средний выигрыш, который мы получим, если выберем стратегию A_i :

$$\bar{a}_i = \sum_{j=1}^n a_{ij} Q_j,$$

которая получается как математическое ожидание случайной величины a_{ij} . Решение принимается по критерию

$$\max_i \bar{a}_i. \quad (9.1)$$

Это критерий максимального среднего выигрыша. Если задана матрица рисков, то можно для каждой строки вычислить средний риск

$$\bar{r}_i = \sum_{j=1}^n r_{ij} Q_j$$

и в качестве оптимального выбрать решение, обеспечивающее минимальный средний риск

$$\min_i \bar{r}_i. \quad (9.2)$$

Покажем, что оптимальные решения, полученные по (9.1) и (9.2), совпадают. Сложим средний выигрыш и средний риск и получим константу

$$\bar{a}_i + \bar{r}_i = \sum_{j=1}^n a_{ij} Q_j + \sum_{j=1}^n r_{ij} Q_j = \sum_{j=1}^n a_{ij} Q_j + \sum_{j=1}^n (\beta_j - a_{ij}) Q_j = \sum_{j=1}^n \beta_j Q_j = c,$$

тогда

$$\bar{a}_i + \bar{r}_i = c \Rightarrow \bar{a}_i = c - \bar{r}_i; \\ \arg \max_i \bar{a}_i = \arg \min_i \bar{r}_i,$$

где c — величина, не зависящая от i , она постоянна для данной матрицы.

В результате решения выбирается чистая стратегия A_i . Есть ли смысл смешивать стратегии? Пусть мы смешиваем наши стратегии с вероятностями p_i , тогда в результате мы получим средний выигрыш в виде математического ожидания (МО):

$$\bar{a} = \bar{a}_1 p_1 + \bar{a}_2 p_2 + \dots + \bar{a}_m p_m.$$

Мы знаем, что МО меньше максимального значения, следовательно, в игре с природой нет смысла смешивать стратегии; чистая стратегия обеспечивает наилучший результат. Слабым местом в этом подходе является то, что надо знать априорные вероятности. Если они неизвестны, то необходимо их изучить. Это можно делать путем экспериментов, которые изучают условия «природы». Говорят, что этим мы обучаем нашу систему. Такой подход называется *принципом адаптации к условиям*.

Если априорные вероятности изучить не удастся, то применяется принцип недостаточности основания Лапласа: если не знаем вероятности, то считаем, что гипотезы природы равновероятны. После этого применяем критерий идеального наблюдателя. Так как при равных вероятностях энтропия (неопределенность) максимальна, то мы рассчитываем на худший случай, т. е. применяем принцип пессимизма. Если сами значения вероятности неизвестны, но есть информация о предпочтениях гипотез, то существуют методы обработки предпочтений и получения вероятностей.

Например, если вероятности гипотез относятся как

$$n : (n-1) : (n-2) : \dots : 2 : 1 = Q_1 : Q_2 : \dots : Q_n; \quad \sum_{j=1}^n Q_j = 1,$$

то можно подсчитать сами величины вероятностей в следующем виде:

$$Q_j = \frac{2(n-j+1)}{n(n+1)}; \quad j = \overline{1, n}.$$

В некоторых случаях учитывается не только средний выигрыш, но и дисперсия, т. е. величина разброса выигрыша в каждой строке, тогда вместо (9.1) используют

$$\max_i (\bar{a}_i - \beta \cdot \sigma_{a_i}); \quad \sigma_{a_i} = \sqrt{D_{a_i}},$$

где σ_{a_i} — среднеквадратическое отклонение для величин i -й строки; β — коэффициент учета рассеяния ($0 < \beta < 1$).

9.2. МЕТОДЫ ПРИНЯТИЯ РЕШЕНИЙ В УСЛОВИЯХ АПРИОРНОЙ НЕОПРЕДЕЛЕННОСТИ

Если априорная информация неизвестна или ненадежна, то применяются другие критерии.

Критерий Вальда (W) — это максиминный критерий. В качестве оптимальной по этому критерию выбирается стратегия A_k , для которой

$$W = \max_i \min_j a_{ij}.$$

При таком критерии мы подходим к этой задаче, рассматривая на самый худший случай, как и в игре с разумным противником.

Критерий Сэвиджа (S) — это критерий минимаксного риска:

$$S = \min_i \max_j r_{ij}.$$

Этот критерий неэквивалентен критерию Вальда, т. е. стратегия, оптимальная по Сэвиджу, необязательно также будет оптимальна по Вальду.

Критерий Гурвица (H) — это комбинированный критерий, его также называют критерием пессимизма–оптимизма

$$H = \max_i (k \min_j a_{ij} + (1-k) \max_j a_{ij}),$$

где $0 < k < 1$ — коэффициент, который выполняет требования критерия быть более или менее оптимистичным. При $k = 1$ критерий $H = W$, а при $k = 0$ получаем крайний оптимизм.

Существуют и другие критерии, и однозначный выбор одного критерия невозможен. Но если одну и ту же ситуацию рассматривают по разным критериям и получают одинаковые решения, то это говорит об устойчивости данной ситуации и однозначности найденного оптимального решения. В противном случае ситуация неустойчивая и необходимо либо изучать априорную информацию, либо доказывать верность лишь одного из критериев.

9.3. ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТА ПРИ ПРИНЯТИИ РЕШЕНИЙ

Если априорной информации нет или она ненадежна, то можно с помощью эксперимента получить более надежные данные о вероятностях Q_j . Под экспериментом понимают систему мероприятий, позволяющих уточнить информацию о состоянии «природы». Например, это метеонаблюдения, маркетинговые исследования в экономике, радиолокация, военная или промышленная разведка и т. д. Насколько может помочь в принятии решения эксперимент и как сопоставить стоимость эксперимента с тем оптимальным выигрышем, который мы получим?

Соответствующую теорию можно построить исходя из знания вероятностей Q_j , а также на основе критериев при неизвестной априорной информации. Мы рассмотрим случай, когда есть априорная информация, т. е. ситуацию идеального наблюдателя. Есть ли смысл проводить эксперимент? Рассмотрим два случая.

1. Идеальный эксперимент. Результат этого эксперимента однозначно определяет, каковы условия природы. Пусть заданы выигрыши a_{ij} и априорные вероятности Q_j . Стоимость эксперимента c сопоставима с выигрышем a_{ij} , т. е. эти величины имеют одинаковую размерность. Сравним средний выигрыш без проведения эксперимента со средним выигрышем при проведении эксперимента.

Если нет эксперимента, мы обеспечим себе

$$\max_i \left(\sum_{j=1}^n a_{ij} Q_j \right) = \bar{a}_{\max}.$$

Если мы проведем эксперимент, то мы точно узнаем $P_j = P_k$, и тогда, найдя в k -м столбце максимальный выигрыш, мы найдем наш выигрыш:

$$\max_i a_{ik} = \beta_k.$$

Но нам нужно оценить эффективность эксперимента до его проведения, поэтому мы должны ориентироваться на средний ожидаемый выигрыш, который получим, если

будем проводить эксперимент. Таким образом, до эксперимента мы можем ожидать выигрыш

$$\sum_{j=1}^n Q_j \cdot \beta_j.$$

Поэтому, чтобы решить, проводить эксперимент или нет, надо определить, что больше: \bar{a}_{\max} или $\sum_{j=1}^n Q_j \cdot \beta_j$.

Получается, что необходимо проводить эксперимент, если

$$\sum_{j=1}^n Q_j \beta_j - c > \max_i \sum_{j=1}^n Q_j a_{ij}.$$

Преобразовав это неравенство, получим, что идеальный эксперимент проводить нужно, если его стоимость меньше минимального среднего риска

$$c < \min_i \bar{r}_i.$$

2. Неидеальный эксперимент. В результате проведения неидеального эксперимента мы не находим однозначно P_j , а лишь изменяем вероятности Q_j . Пусть проводится неидеальный эксперимент. В результате появляются некоторые несовместные события B_1, B_2, \dots, B_L . Вероятности этих событий зависят от условий, в которых они проводятся. Пусть известны $P(B_l/P_j)$. Эти вероятности называются прямыми. После эксперимента, давшего исход B_l , необходимо пересмотреть вероятности Q_j , т. е. вместо вероятности Q_j мы перейдем к вероятности Q_{jl} . Это так называемые апостериорные вероятности. Они получаются по формуле Байеса:

$$\tilde{Q}_{jl} = P(P_j/B_l) = \frac{Q_j \cdot P(B_l/P_j)}{\sum_{j=1}^n Q_j \cdot P(B_l/P_j)}.$$

Но результаты эксперимента могут быть и B_1 , и B_2 , и B_k , поэтому мы можем только ожидать всякие исходы B_l , которые получатся в результате эксперимента. Причем каждый исход B_l привел бы к некоторым оптимальным стратегиям A_l^* . А величина выигрыша, которая бы при этом получилась, будет:

$$\bar{a}_i = \max_i \bar{a}_{il} = \max_i \sum_{j=1}^n (\tilde{Q}_{jl} \cdot a_{ij}).$$

Эти выигрыши \bar{a}_i могут произойти с вероятностью события B_l , т. е. с вероятностью $P(B_l)$, которую можно получить по формуле полной вероятности

$$P(B_l) = \sum_{j=1}^n Q_j P(B_l / P_j).$$

Тогда средний ожидаемый выигрыш будет

$$\bar{\bar{a}} = \sum_{l=1}^L \bar{a}_l P(B_l) \Rightarrow \bar{\bar{a}} - c > \max_i \sum_{j=1}^n Q_j a_{ij} \Rightarrow \bar{\bar{a}} - c > \bar{a}_{\max}.$$

Планирование экспериментов можно рассмотреть и для случаев, когда проводят 2, 3, 4, ... эксперимента [8].

9.4. МНОГОЭТАПНОЕ ПРИНЯТИЕ РЕШЕНИЙ

Мы рассмотрели различные критерии принятия решений в условиях «одноэтапных» альтернатив. На практике в таких задачах, как проектирование изделий, управление процессами, создание сложных программных комплексов, мы можем столкнуться с принятием последовательных решений (цепочкой решений, рис. 76).

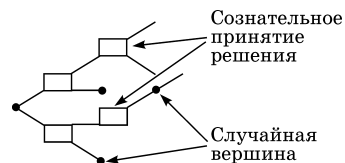


Рис. 76

Схема многоэтапных решений

Особое значение такие многоэтапные решения имеют при создании автоматизированных экспертных систем. Рассмотрим вопрос оптимизации многоэтапных решений. Многоэтапность приводит к тому, что схема принятия решения может быть представлена в

виде дерева, в каждой вершине которого осуществляется:

1) сознательный выбор между двумя и более альтернативами;

2) случайный переход из одной ветви в другую под воздействием внешних случайных факторов.

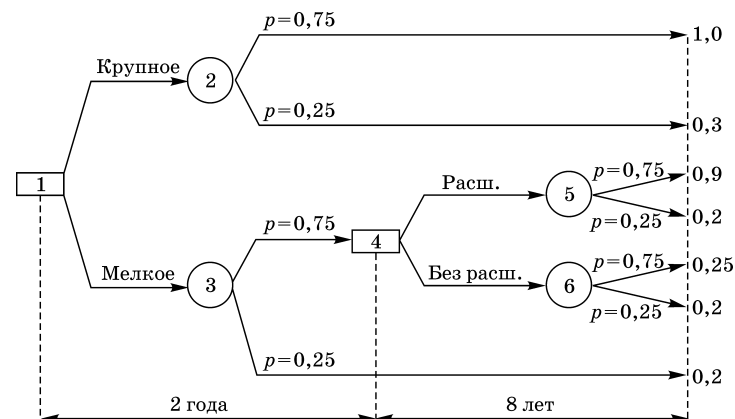


Рис. 77

Дерево решений для операции двухэтапного строительства

Рассмотрим оптимизацию многоэтапных решений на примере экономической задачи.

Пример. Фирма может принять решение о строительстве крупного или мелкого предприятия. Строительство крупного предприятия относительно дешевле в случае если будет высокий спрос на производимые товары. Мелкое предприятие можно расширить. Деятельность фирмы рассматривается в течение десяти лет, причем при строительстве мелкого предприятия вопрос о расширении будет рассматриваться через два года. Спрос заранее неизвестен (рис. 77).

Введем градацию случайного спроса: высокий спрос ожидается с вероятностью $p = 0,75$ и низкий с $p = 0,25$. Строительство крупного предприятия требует 5 млн долл.; строительство мелкого — 1 млн долл.; затраты на расширение — 4,2 млн долл.; крупное предприятие при высоком спросе дает доход 1 млн долл. ежегодно, а при низком — 300 тыс. долл.; мелкое предприятие при высоком спросе — 250 тыс. долл. ежегодно, при низком — 200 тыс. долл.; расширенное предприятие в случае высокого спроса приносит доход 900 тыс. долл. в год, при низком спросе — 200 тыс. долл.; мелкое предприятие без расширения при высоком спросе на производимый продукт приносит в те-



ГЛАВА 10

ЭКСПЕРТНЫЕ ПРОЦЕДУРЫ ДЛЯ ПРИНЯТИЯ РЕШЕНИЙ

В практических задачах принятия оптимального решения альтернативы не являются «математическими объектами», а чаще представляют собой конкретные физические системы: продукты, технологии, организацию технического мероприятия, системы и т. д. Для описания альтернатив и оценки последствий их принятия необходимо решить следующие задачи:

- построить множество возможных и допустимых альтернатив;
- сформировать набор аспектов, существенных для оценки альтернатив;
- определить критериальное пространство;
- упорядочить альтернативы по аспектам;
- получить оценку альтернатив по критериям, т. е. найти отображение Ω в критериальном пространстве (см. главу 7).

Все эти задачи являются модификацией общей задачи оценивания: сопоставления числа или нескольких чисел рассматриваемой альтернативе.

Методы решения задачи оценивания основаны на использовании экспертных процедур, поэтому их называют *методами экспертных оценок*.

10.1.

ОБЩАЯ СХЕМА ЭКСПЕРТИЗЫ

В общем случае из-за сложности оценивания систем привлекаются специалисты в данной предметной области — эксперты. Решение задач оценивания называют экс-

пертизой. Вопросы, связанные с экспертизой, рассматриваются и решаются консультантом. Он определяет Ω , а иногда и вспомогательное множество для экспертизы Ω_3 и организует всю процедуру экспертизы.

1. Консультант находит множество допустимых оценок Ω , в которых содержится исходная оценка.

2. Консультант определяет множество допустимых оценок Ω_3 , из которого осуществляют выбор эксперты.

3. Каждый эксперт выбирает свою оценку $a_i = C_i(\Omega_3)$, $i = 1, N$. При этом эксперты могут взаимодействовать между собой.

4. По заранее разработанному алгоритму (формуле) консультант обрабатывает полученную от экспертов информацию и находит результирующую оценку, являющуюся решением исходной задачи оценивания.

5. Если полученное решение не устраивает консультанта, то он предоставляет экспертам дополнительную информацию, т. е. организует обратную связь, после чего они вновь решают задачу оценивания.

10.2.

ЗАДАЧА ОЦЕНИВАНИЯ

Смысл оценивания состоит в сопоставлении альтернативе вектора из E_m . Перечислим типичные варианты этой задачи:

1. Пусть $X \in \Omega$ — альтернатива в задаче принятия решений. Имеются m критериев. Требуется альтернативе $X \in \Omega$ сопоставить вектор

$$[f_1(x), f(x), \dots, f_m(x)] \in E_m.$$

2. Пусть k_1, k_2, \dots, k_m — критерии, учитывающиеся при выборе. Их необходимо упорядочить по важности. Тогда системе $S = (k_1, k_2, \dots, k_m)$ сопоставляется перестановка натуральных чисел от 1 до m , i_1, \dots, i_m , где i_k — номер k -го критерия при упорядочивании по важности.

3. Пусть множество D разбито на l подмножеств D_1, D_2, \dots, D_l . Для элемента $x \in D$ необходимо указать, к какому из подмножеств D_i ($i = 1, l$) он относится, т. е. x сопоставляется одно из чисел от 1 до l .

4. Пусть x — отрезок, длину которого надо измерить, т. е. отрезку надо сопоставить действительное число; $f(x)$ — длина отрезка.

Задача 1 — это общая задача многокритериальной оценки.

Задача 2 — это задача ранжирования.

Задача 3 — это задача классификации.

Задача 4 — это обычная задача измерения.

Обозначим:

Ω — исходное множество допустимых значений оценок (МДО);

Ω_{ω} — МДО для экспертов;

L — взаимодействие между экспертами;

Q — обратная связь;

φ — обработка (отображение $\Omega_{\omega}^N \rightarrow \Omega$).

Назовем схемой экспертизы пятерку параметров $\Omega, \Omega_{\omega}, L, Q, \varphi$.

Подготовка экспертизы — это предварительная разработка схем экспертизы и подбор экспертов.

Реализация экспертизы — получение от экспертов информации и ее обработка.

10.3. ПОДГОТОВКА ЭКСПЕРТИЗЫ

Определение МДО (множества допустимых оценок). Перечислим типы МДО и укажем определяющие их задачи оценивания.

1. $\Omega = \{0, 1\}$. Соответствующая задача попарного сравнения заключается в выявлении лучшего из двух имеющихся объектов a и b :

$$C(\Omega) = \begin{cases} 1, & \text{если } a \text{ лучше } b, \\ 0, & \text{в противном случае.} \end{cases}$$

2. $\Omega = \{(1, 2, \dots, n), (1, 3, \dots, n), \dots, (n, n-1, \dots, 1)\}$, т. е. Ω состоит из множества перестановок длины n . Соответствующая задача ранжирования состоит в упорядочивании объектов:

$$C(\Omega) = (i_1, i_2, \dots, i_n).$$

3. $\Omega = \{1, \dots, l\}$. Соответствующая задача классификации

$$C(\Omega) = i, \text{ если } X \in S_i.$$

Определение МДО для экспертов Ω_{ω} . Типы Ω_{ω} те же, что и Ω . Но Ω_{ω} зависит от формы опроса: интервью, анкетирование, метод докладной записки эксперта — высказывание мнения в свободной форме.

Взаимодействие экспертов L .

1. Свободный обмен информацией между экспертами.

2. Обмен регламентирован.

3. Эксперты изолированы.

Пункт 2 может иметь вид «мозговой атаки»: в течение заданного времени любое мнение не может быть отвергнуто и не подлежит обсуждению.

Обратная связь. После обработки можно ознакомить экспертов с результатами и попросить дать повторную оценку.

Известен *метод Делфи*. После экспертизы проверяется согласованность, а затем, если она недостаточная, экспертам сообщается дополнительная информация и аргументация других экспертов. Затем экспертиза повторяется.

Подбор экспертов. Существуют методы оценки компетентности экспертов. Эксперты могут выставлять свои оценки компетентности себе и другим. Затем эта информация обобщается и используется в алгоритмах оценки.

10.4. МЕТОДЫ ОБРАБОТКИ ЭКСПЕРТНОЙ ИНФОРМАЦИИ

Существуют три вида методов обработки:

- статистические;
- алгебраические;
- шкалирования.

Статистические методы экспертных оценок. Результаты оценок каждого эксперта можно рассматривать как реализацию некоторой случайной величины из Ω_{ω} и применять к ним методы математической статистики. Статистические методы позволяют определить согласованность

мнений экспертов, значимость полученных оценок и т. д., т. е. качество экспертизы.

Численные оценки. Экспертиза \mathcal{E}_1 . Задача состоит в сопоставлении оцениваемой альтернативе (системе) одного числа. ($\Omega = E_1, \Omega_3 = E_1, L$ — эксперты изолированы; Q — обратная связь отсутствует)

$$\varphi(x_1, x_2, \dots, x_N) = \frac{\sum_{i=1}^N x_i \alpha_i}{\sum_{i=1}^N \alpha_i} = a, \quad (10.1)$$

где α_i — вес (коэффициент компетентности) экспертов; x_i — числовые оценки экспертов.

При отсутствии информации о компетентности экспертов $\alpha_i = 1$. Степенью согласованности мнений экспертов служит дисперсия:

$$\sigma^2 = \frac{\sum_{i=1}^N (a - x_i)^2 \alpha_i}{\sum_{i=1}^N \alpha_i}. \quad (10.2)$$

Другая экспертиза \mathcal{E}_2 повышает точность оценивания ($\Omega_3 = E_3$)

$$\begin{aligned} \varphi(x_1^1, x_2^1, x_3^1, x_1^2, x_2^2, x_3^2, \dots, x_1^N, x_2^N, x_3^N) = \\ = \sum_{i=1}^N \frac{x_1^i \gamma_1 + x_2^i \gamma_2 + x_3^i \gamma_3}{\gamma_1 + \gamma_2 + \gamma_3} \alpha_i \cdot \frac{1}{\sum_{i=1}^N \alpha_i}, \end{aligned} \quad (10.3)$$

где x_1^i, x_2^i, x_3^i — соответственно оптимистическая, наиболее вероятная и пессимистическая оценки i -го эксперта; $\gamma_1, \gamma_2, \gamma_3$ определяются эмпирически. Например, по одной методике $\gamma_1 = 1, \gamma_2 = 4, \gamma_3 = 36$, по другой $\gamma_1 = 3, \gamma_2 = 0, \gamma_3 = 2, \gamma_4 = 25$, где γ_4 — степень неуверенности эксперта в своем ответе, $\gamma_1 > \gamma_3$, так как человек склонен к занижению оценки.

Степень согласованности экспертов определяется дисперсией

$$\begin{aligned} \sigma^2 &= \sum_{i=1}^N \alpha_i \sigma_i^2 \cdot \frac{1}{\sum_{i=1}^N \alpha_i} + \sum_{i=1}^N \left(a - \frac{x_1^i \gamma_1 - x_2^i \gamma_2 - x_3^i \gamma_3}{\gamma_1 + \gamma_2 + \gamma_3} \right)^2 \cdot \alpha_i \frac{1}{\sum_{i=1}^N \alpha_i}, \\ \sigma_i^2 &= (x_3^i - x_1^i)^2 \cdot \frac{1}{\gamma_4}. \end{aligned} \quad (10.4)$$

В экспертизах \mathcal{E}_1 и \mathcal{E}_2 можно определить статистическую значимость полученных результатов. Задавшись $P_{\text{ош}}$, укажем интервал, в который оцениваемая величина попадет с вероятностью $1 - P_{\text{ош}}$.

$$\bar{a} - \Delta < a < \bar{a} + \Delta, \quad (10.5)$$

где a распределена нормально с центром \bar{a} и дисперсией (10.2).

Тогда

$$\Delta = \frac{t\sigma}{\sqrt{N}},$$

где t — стандартная ошибка, находящаяся по таблице распределения коэффициента Стьюдента $t = f(N - 1, P_{\text{ош}})$.

Пример. Десять экспертов с одинаковым весом $\alpha_i = 1$ оценивают величину T . Ими получены результаты: $T_1 = 33; T_2 = 35; T_3 = 32,2; T_4 = 34; T_5 = 38; T_6 = 34; T_7 = 37; T_8 = 40; T_9 = 36; T_{10} = 35,5$.

По формуле (10.1) получаем $T = 35,5, \sigma^2 = 4,9, \sigma = 2,2136$.

Задав $P_{\text{ош}} = 0,05$, определим $t = f(9; 0,05) = 2,262$. По формуле (10.5) $\Delta = 1,583$. Таким образом, с вероятностью 0,95 величина T находится в интервале $[33, 917; 37,083]$.

10.5. МЕТОД ДЕЛФИ ДЛЯ ЧИСЛЕННОЙ ОЦЕНКИ

Опишем применение метода Делфи для численной оценки.

$$\begin{aligned} \Omega &= E_1, \\ \Omega_3 &\left\{ Z \in E_k \mid \sum_{i=1}^k Z_i = 1; \quad Z_i \geq 0 \right\}; \end{aligned}$$

L — эксперты изолированы;

Q — экспертам предоставляется медиана q_2 :

$$P(T \leq q_2) = 0,5,$$

а также диапазон квантилей

$$\begin{aligned} \Delta q &= q_3 - q_1; \\ P(T \leq q_3) &= 0,75; \\ P(T \leq q_1) &= 0,25. \end{aligned}$$

Задаем φ следующим образом.

Весь интервал допустимых значений оцениваемой величины разбивается на k интервалов t_1, t_2, \dots, t_k ; эксперт оценивает вероятность попадания оцениваемой величины в каждый из интервалов; по результатам их оценок составляется таблица (табл. 57), где P_{ij} — оценка вероятности попадания оцениваемой величины в j -й интервал, данная i -м экспертом.

Т а б л и ц а 57

Эксперт	Интервал				
	t_1	t_2	t_3	...	t_k
1	P_{11}	P_{12}	P_{13}	...	P_{1k}
2	P_{21}	P_{22}	P_{23}	...	P_{2k}
...
N	P_{N1}	P_{N2}	P_{N3}	...	P_{Nk}

На основе этой таблицы определяется мнение экспертов о попадании оцениваемой величины в каждый из интервалов

$$P_{t_j} = \sum_{i=1}^N P_{ij} \alpha_i \cdot \frac{1}{\sum_{i=1}^N \alpha_i}. \quad (10.6)$$

Результирующей оценкой является медиана распределения

$$P(T \leq q_2) = 0,5.$$

Ее показывают экспертам, а также показывают диапазон $\Delta q = q_3 - q_1$.

Затем снова эксперты дают оценки P_{ij} и т. д. до тех пор, пока диапазон Δq не уменьшится в 1,6 раза по сравнению с предварительным.

10.6. СТРОГОЕ РАНЖИРОВАНИЕ

Ω — множество всех перестановок,

$\Omega_s = \Omega$;

L — эксперты изолированы друг от друга;

Q — обратная связь отсутствует.

Отображение φ определяется так: результаты опроса сводятся в таблицу рангов данных i -м экспертом j -му объекту (табл. 58).

$$r_s = \sum_{i=1}^N r_{is}.$$

Т а б л и ц а 58

Эксперт	Объект			
	1	2	...	n
1	r_{11}	r_{12}	...	r_{1n}
2	r_{21}	r_{22}	...	r_{2n}
...
N	r_{N1}	r_{N2}	...	r_{Nn}
Σ рангов	r_1	r_2	...	r_n

На первое место ставится объект с меньшей величиной r , $r_{ij} \neq r_{lk}$ (в строках нет одинаковых рангов).

Степень согласованности мнений экспертов определяется при помощи коэффициента конкордации:

$$W = \frac{12 \sum_{j=1}^n \left[r_j - \frac{1}{2} N(n+1) \right]^2}{N^2(n^3 - n)}, \quad (10.7)$$

где N — количество экспертов; n — количество объектов; r_j — ранги объектов, полученных в результате экспертизы.

По сути дела W — это сумма квадратов отклонений рангов r_j от среднего значения

Т а б л и ц а 59

Признак	Номер эксперта										Σr_{ij}
	1	2	3	4	5	6	7	8	9	10	
Шум	6	1	6	6	6	6	4	5	6	6	52
Цвет футеровки	4	5	4	5	5	3	5	6	4	5	46
Цвет пламени	2	2	2	3	3	2	1	1	1	2	19
Цвет дыма	1	4	3	2	2	4	3	3	3	3	28
Качество дыма	3	3	1	1	1	1	2	2	2	1	17
Искры	5	6	5	4	4	5	6	4	5	4	48

$$r_{\text{ср}} = \sum_{j=1}^n \frac{r_j}{n} = \frac{N(n+1)}{2}. \quad (10.8)$$

Пример. Проводится экспертиза по оценке технологического процесса выплавки стали в конверторе. Задан список из шести признаков, влияющих на процесс. Десять экспертов ранжировали признаки по важности (табл. 59).

В результате подсчета получим $r_{\text{ср}} = 35$; $W = 0,69$.

При четном $N(n+1)$ может быть $W = 0$.

10.7. НЕСТРОГОЕ РАНЖИРОВАНИЕ

Если объекты делят 4–5-е места, то каждый получает 4,5. В этом случае

$$W = \frac{12 \sum_{j=1}^n \left[r_j - \frac{1}{2} N(n+1) \right]^2}{N^2(n^3 - n) - N \sum_{i=1}^N \sum_{j=1}^{k_i} (t_{ij}^3 - t_{ij})}, \quad (10.9)$$

где k_i — количество групп равных рангов, введенных i -м экспертом; t_{ij} — количество дробных рангов в j -й группе, введенной i -м экспертом.

Можно учесть компетентность α_i :

$$r_j = \sum_{i=1}^N r_{ij} \alpha_i.$$

10.8. МЕТОД ПОПАРНЫХ СРАВНЕНИЙ

В этом методе эксперт i фиксирует предпочтение S -го объекта над j -м объектом в виде чисел $\{1, 0\}$

$$d_{sj}^i = \begin{cases} 1, & \text{если } S \text{ превосходит объект } j; \\ 0, & \text{в противном случае.} \end{cases}$$

Далее эти результаты записываются в матрицу

$$D = \{d_{sj}\}, \text{ где } d_{sj} = \begin{cases} 1, & \text{при } d_{sj}(+) > d_{sj}(-); \\ 0, & \text{в противном случае;} \end{cases}$$

$$d_{sj}(-) = \sum_{i \in M^-} d_{sj}^i;$$

$$d_{sj}(+) = \sum_{i \in M^+} d_{sj}^i,$$

M^+ — множество экспертов, считающих, что S доминирует j ;

M^- — множество экспертов, считающих наоборот;

$$M^- = M - M^+.$$

Окончательный ранг объекта определяется по сумме элементов строки

$$r_s = \sum_{j=1}^n d_{sj}. \quad (10.10)$$

Пример.

$$D = \{d_{sj}\} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

$$r_s = (2, 3, 0, 1).$$

Следовательно, объекты O_s необходимо упорядочить по ценности так: O_2, O_1, O_4, O_3 .

Применяются и другие методы обработки экспертной информации (см., например [42]).

ГЛАВА 11

ОПЕРАЦИИ МАССОВОГО ОБСЛУЖИВАНИЯ

Многие операции приходится производить в своеобразных системах, называемых *системами массового обслуживания* (СМО).

Примеры СМО: телефонные станции, ЭВМ, кассы, ремонтные мастерские, библиотеки, склады, билетные кассы, магазины, парикмахерские и т. п.

Каждая СМО состоит из какого-то числа обслуживающих единиц, которые мы будем называть *каналами* обслуживания. СМО могут быть *одноканальными* и *многоканальными*.

11.1. МОДЕЛЬ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

Каждая СМО предназначена для выполнения потока заявок, поступающих в СМО обычно в случайные моменты времени. Обслуживание заявки происходит в течение некоторого момента времени (вообще говоря, случайно), после чего канал освобождается и готов принять следующую заявку. Иногда на входе СМО ввиду случайного характера заявок скапливается «очередь», иногда же СМО недогружена (простаивает). Каждая СМО может характеризоваться пропускной способностью.

Теория массового обслуживания устанавливает зависимость между характером потока заявок, количеством каналов, их производительностью, правилами работы СМО и эффективностью обслуживания.

В качестве характеристик эффективности обслуживания в зависимости от условий задачи применяются:

1. Среднее количество заявок, которые может обслужить СМО в единицу времени (пропускная способность).

2. Средний процент заявок, покидающих СМО необслуженными.

3. Среднее время ожидания в очереди.

4. Средний доход, приносимый СМО в единицу времени и т. д.

Случайный характер заявок обуславливает то, что в СМО происходит некоторый случайный процесс. Теория массового обслуживания математически описывает этот процесс.

Анализ СМО существенно облегчается, если случайный процесс в СМО является марковским.

11.2. МАРКОВСКИЙ СЛУЧАЙНЫЙ ПРОЦЕСС С ДИСКРЕТНЫМИ СОСТОЯНИЯМИ

Определение. Случайный процесс, протекающий в системе S , называется *марковским*, если для каждого момента времени t_0 вероятность любого состояния системы в будущем (при $t > t_0$) зависит только от ее состояния в настоящем (при $t = t_0$) и не зависит от того, когда и каким образом система пришла в это состояние (т. е. как процесс развивался в прошлом). Случайный процесс называется *процессом с дискретными состояниями*, если возможные состояния системы S_1, S_2, S_3, \dots можно перечислить, т. е. множество S счетное.

Кроме таких процессов имеются процессы с непрерывным изменением состояния, например изменение электрического тока в лампе и т. п.

Будем изображать каждое состояние S_1, S_2, S_3, \dots прямоугольником, а возможные переходы за один шаг — стрелкой. Такие графические схемы называются *графами состояний* (рис. 79).

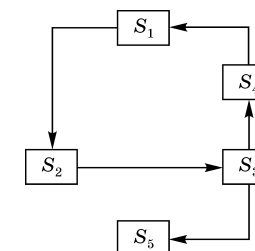


Рис. 79
Пример графа состояний

S_1 – S_5 — ЭВМ соответственно работает исправно; неисправна; ожидает осмотра; осматривается; ремонтируется; списана.

S_1-S_5 — ЭВМ соответственно работает исправно, не исправна, осматривается, ремонтируется, списана.

Случайный процесс называется *процессом с дискретным временем*, если переход системы из состояния в состояние возможен только в заранее фиксированные моменты времени t_1, t_2, \dots

Случайный процесс называется *процессом с непрерывным временем*, если переход из одного состояния в другое возможен в любой момент времени.

Рассмотрим марковский случайный процесс с дискретными состояниями и дискретным временем. Известны

$$\left\{ S_1, S_2, \dots, S_n \right\}, \left\{ t_1, t_2, \dots, t_k \right\}.$$

Моменты t_1, t_2, \dots будем называть «шагами», или «этапами». Случайный процесс заключается в том, что в последовательные моменты времени t_1, t_2, \dots система может быть в тех или иных состояниях из S_1, S_2, S_3, \dots

Например:

$$S_1 \rightarrow S_3 \rightarrow S_4 \rightarrow S_2 \rightarrow S_{10} \rightarrow S_4 \rightarrow \dots$$

или

$$S_1 \rightarrow S_1 \rightarrow S_1 \rightarrow S_4 \rightarrow S_1 \rightarrow \dots$$

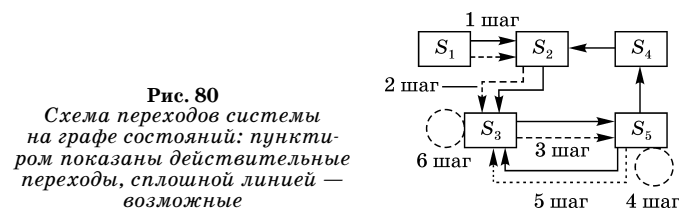
Обозначим $S_1^{(k)}$ — событие, состоящее в том, что после k шагов система находится в состоянии S_i . Тогда для любого k события $S_1^{(k)}, S_2^{(k)}, S_3^{(k)}, \dots, S_n^{(k)}$ образуют полную группу событий.

Процесс можно еще записать так: $S_1^{(0)}, S_1^{(1)}, S_3^{(2)}, S_1^{(3)}, \dots$ Такая случайная последовательность называется *простой марковской цепью*, если для каждого шага вероятность перехода из любого состояния S_i в любое S_j не зависит от того, когда и как система пришла в состояние S_i .

Опишем марковскую цепь при помощи вероятностей состояний. Тогда

$$P_1(1) = P\{S_1^{(1)}\}, \quad P_2(1) = P\{S_2^{(1)}\} \quad \text{и т. д.,}$$

очевидно $\sum_{i=1}^n P_i(k) = 1$ — условие полноты.



Будем называть $P_i(k)$ *вероятностями состояний*.

Найдем вероятности состояний системы для любого k . Марковскую цепь можно представить себе так, будто точка, изображающая систему S , случайным образом блуждает по графу состояний, «перескакивая» из состояния в состояние или задерживаясь в них (рис. 80).

Для любого дискретного момента времени t_1, t_2, \dots существует вероятность перехода системы из одного состояния в другое (переходные вероятности).

Марковская цепь называется *однородной*, если переходные вероятности не зависят от номера шага, в противном случае — *неоднородной*.

Рассмотрим однородную марковскую цепь, имеющую n состояний. Обозначим P_{ij} вероятность перехода из S_i в S_j (P_{ij} не зависит от k), P_{ii} — вероятность задержки в состоянии S_i

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{bmatrix}. \quad (11.1)$$

Существуют ij , для которых $P_{ij} = 0$. Используя события $S_1^{(k)}$, можно P_{ij} записать в виде

$$P_{ij} = P\{S_j^{(k)} | S_i^{(k-1)}\},$$

таким образом сумма членов, стоящих в каждой строке матрицы (11.1), равна 1.

Имея размеченный граф состояний и зная начальное состояние системы, можно найти вероятности состояний $P_1(k), P_2(k), \dots, P_n(k)$ после любого k -го шага.

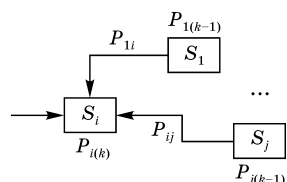


Рис. 81
Граф состояний для систем с дискретными состояниями и дискретным временем

Легко показать (рис. 81), что вероятности состояний $P_i(k)$ после k -го шага определяются рекуррентной формулой через вероятности состояний после $(k-1)$ шага

$$P_i(k) = \sum_{j=1}^n P_j(k-1)P_{ji}, \quad i = \overline{1, n}. \quad (11.2)$$

Аналогично для неоднородной цепи Маркова

$$P_i(k) = \sum_{j=1}^n P_j(k-1)P_{ji}^{(k)}, \quad i = \overline{1, n}, \quad (11.3)$$

где $P_{ij}^{(k)}$ — вероятность перехода системы из состояния S_i в состояние S_j на k -м шаге

$$P_{ij}^{(k)} = P\{S_j^{(k)} | S_i^{(k-1)}\}.$$

11.3. МАРКОВСКИЙ ПРОЦЕСС С ДИСКРЕТНЫМ СОСТОЯНИЕМ И НЕПРЕРЫВНЫМ ВРЕМЕНЕМ. СИСТЕМА УРАВНЕНИЙ КОЛМОГОРОВА

На практике часто переходы системы происходят в случайные моменты времени, которые заранее указать нельзя. Например, выход из строя аппаратуры, телефонный звонок и т. д.

Покажем, как выражаются вероятности состояний для этого процесса, который мы будем называть *непрерывной цепью Маркова*.

Оказывается, что все характеристики СМО легко получаются, если знать вероятности всех состояний СМО.

Пусть $P_i(t)$ — вероятность того, что в момент t система S будет находиться в состоянии S_i ($i = \overline{1, n}$). Очевидно, для любого t

$$\sum_{i=1}^n P_i(t) = 1, \quad \forall t.$$

Здесь вместо переходных вероятностей P_{ij} нужно ввести плотность вероятностей перехода λ_{ij} .

Пусть в момент t система была в состоянии S_i . Рассмотрим отрезок времени Δt , примыкающий к t , тогда

$$\lambda_{ij} = \lim_{\Delta t \rightarrow 0} \frac{P_{ij}(\Delta t)}{\Delta t},$$

где $P_{ij}(\Delta t)$ — вероятность того, что система за время Δt перейдет из S_i в S_j ($i \neq j$). Очевидно

$$P_{ij}(\Delta t) \approx \lambda_{ij}\Delta t.$$

Если все плотности λ_{ij} не зависят от t , то марковский процесс называется *однородным (стационарным)*, в противном случае — *неоднородным (нестационарным)*.

Граф с проставленными у стрелок плотностями вероятностей перехода называется *размеченным графом состояний* (рис. 82).

Плотность вероятности перехода λ_{ij} еще называют *интенсивностью*, так как она показывает среднее количество переходов (событий) в единицу времени.

Пусть мы знаем размеченный граф состояний, тогда можно определить вероятности состояний $P_1(t), P_2(t), \dots, P_n(t)$ как функции времени. Вероятности $P_i(t)$, $i = \overline{1, n}$ удовлетворяют системе дифференциальных уравнений Колмогорова, решая которую, можно получить $P_i(t)$, $i = \overline{1, n}$.

Пример. Возьмем четыре состояния S_1, S_2, S_3, S_4 . Найдём $P_1(t)$ (рис. 83).

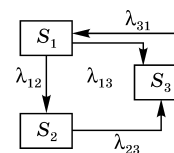


Рис. 82
Граф состояний для систем с дискретными состояниями и непрерывным временем

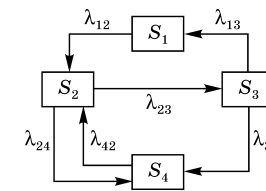


Рис. 83
Пример графа состояний для систем с непрерывным временем

Придадим t приращение Δt и найдем вероятность того, что в момент $(t + \Delta t)$ система находится в S_1 . Это событие может произойти двумя способами:

1. В момент t система была в состоянии S_1 , а за время Δt не вышла из него.

2. В момент t система была в состоянии S_3 , а за время Δt перешла из него в S_1 .

Вероятность первого варианта, очевидно, будет найдена как произведение $P_1(t)$ на вероятность того, что система не перейдет из S_1

$$P_1(t) \cdot (1 - \lambda_{12}\Delta t).$$

Вероятность второго варианта

$$P_3(t) \cdot \lambda_{31}\Delta t.$$

По правилу сложения вероятностей

$$P_1(t + \Delta t) = P_1(t)(1 - \lambda_{12}\Delta t) + P_3(t)\lambda_{31}\Delta t.$$

Разделим на Δt и перейдем к пределу при $\Delta t \rightarrow 0$

$$\begin{aligned} \frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} &= -P_1(t)\lambda_{12} + P_3(t)\lambda_{31}; \\ \lim_{\Delta t \rightarrow 0} \frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} &= \lim_{\Delta t \rightarrow 0} (-P_1(t)\lambda_{12} + P_3(t)\lambda_{31}); \\ \frac{dP_1(t)}{dt} &= -P_1(t)\lambda_{12} + P_3(t)\lambda_{31}. \end{aligned}$$

Рассмотрим S_2 и найдем $P_2(t + \Delta t)$ — вероятность того, что в момент $(t + \Delta t)$ система будет находиться в S_2 . Здесь это событие может произойти тремя способами:

1. В момент t система была в состоянии S_2 , а за время Δt не вышла из него.

2. В момент t система была в состоянии S_1 , а за время Δt перешла из него в S_2 .

3. В момент t система была в состоянии S_4 , а за время Δt перешла из него в S_2 .

$$\text{Для (1): } P_2(t) \cdot [1 - \lambda_{23}\Delta t - \lambda_{24}\Delta t].$$

$$\text{Для (2): } P_1(t) \cdot \lambda_{12}\Delta t.$$

$$\text{Для (3): } P_4(t) \cdot \lambda_{42}\Delta t.$$

$$\begin{aligned} P_2(t + \Delta t) &= P_2(t)(1 - \lambda_{23}\Delta t - \lambda_{24}\Delta t) + P_1(t)\lambda_{12}\Delta t + P_4(t)\lambda_{42}\Delta t; \\ \frac{dP_2(t)}{dt} &= -P_2(t)\lambda_{23} - P_2(t)\lambda_{24} + P_1(t)\lambda_{12} + P_4(t)\lambda_{42}. \end{aligned}$$

Аналогично для S_3 , S_4 получаем

$$\left. \begin{aligned} \frac{dP_1}{dt} &= -\lambda_{12}P_1 + \lambda_{31}P_3; \\ \frac{dP_2}{dt} &= -\lambda_{23}P_2 - \lambda_{24}P_2 + \lambda_{12}P_1 + \lambda_{42}P_4; \\ \frac{dP_3}{dt} &= -\lambda_{31}P_3 - \lambda_{34}P_3 + \lambda_{23}P_2; \\ \frac{dP_4}{dt} &= -\lambda_{42}P_4 + \lambda_{24}P_2 + \lambda_{34}P_3. \end{aligned} \right\} \quad (11.4)$$

Это и есть система уравнений Колмогорова для вероятностей состояний. Интегрируя их, можно найти $P_i(t)$, $i = 1, n$. Начальные условия берутся в зависимости от начального состояния системы. Если при $t = 0$ система была в S_1 , то надо принять при $t = 0$: $P_1 = 1$; $P_2 = P_3 = P_4 = 0$.

Заметим, что в системе (11.4) достаточно трех уравнений, так как

$$P_4 = 1 - (P_1 + P_2 + P_3).$$

Структура (11.4) имеет закономерность, которая позволяет записать систему уравнений Колмогорова по виду графа состояний.

Алгоритм составления уравнений Колмогорова. В левой части каждого уравнения стоит производная вероятности состояния, а правая часть содержит столько членов, сколько стрелок связано с данным состоянием. Если стрелка направлена из состояния, то соответствующий член имеет знак «-», если в состояние, то знак «+». Каждый член равен произведению плотности вероятности перехода, соответствующей данной стрелке, умноженной на вероятность того состояния, из которого исходит стрелка.

Это правило справедливо для любых марковских цепей и позволяет механически записывать уравнения Колмогорова.

Пример. Дан граф состояний (рис. 84).

$$t = 0; P_1 = 1; P_2 = P_3 = P_4 = 0;$$

$$\frac{dP_1}{dt} = -(\lambda_{12} + \lambda_{13})P_1;$$

$$\frac{dP_2}{dt} = P_1\lambda_{12} + \lambda_{32}P_3;$$

$$\frac{dP_3}{dt} = -\lambda_{32}P_3 + \lambda_{13}P_1 - \lambda_{43}P_3;$$

$$\frac{dP_4}{dt} = \lambda_{43}P_3.$$

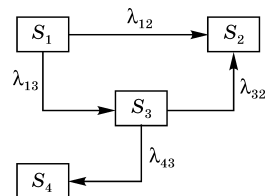


Рис. 84
Граф состояний для системы с дискретными состояниями и непрерывным временем

11.4. ПОТОК СОБЫТИЙ. ПРОСТЕЙШИЙ ПОТОК И ЕГО СВОЙСТВА

Поток событий — последовательность однородных событий, следующих одно за другим в случайные моменты времени (рис. 85).

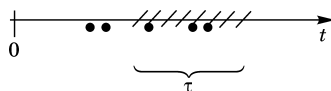


Рис. 85
Иллюстрация потока событий

1. Поток событий называется *стационарным*, если вероятность попадания того или иного числа событий на участок времени длиной τ зависит только от длины участка и не зависит от его местонахождения. (Стационарность может быть локальной, когда указанное свойство выполняется только на каком-то интервале времени.)

2. Поток событий называется потоком *без последствий*, если для любых непересекающихся участков времени число событий, попадающих на один из них, не зависит от того, сколько событий попало на другой (другие).

3. Поток называется *ординарным*, если вероятность попадания на элементарный участок двух событий или более пренебрежимо мала по сравнению с вероятностью попадания ординарных событий. (Например, поток в ЗАГС не является ординарным).

Поток, удовлетворяющий свойствам 1–3 называется *простейшим пуассоновским потоком*.

Можно доказать, что при взаимном наложении достаточно большого числа потоков, обладающих последствием (лишь бы они были стационарными и однородными), образуется простейший поток.

Если поток не имеет последствий, ординарен, но не стационарен, он называется *нестационарным пуассоновским потоком*. В таком потоке интенсивность λ (среднее число событий в единицу времени) зависит от времени $\lambda = \lambda(t)$.

Пуассоновский (стационарный и нестационарный) поток тесно связан с известным распределением Пуассона, в соответствии с которым *число событий потока, попадающих на любой участок, распределено по закону Пуассона*.

Рассмотрим участок времени от t_0 до $t_0 + \tau$. Тогда, как известно из теории вероятностей, вероятность попадания на участок τ ровно m событий

$$P_m = \frac{a^m}{m!} e^{-a}, \quad (m = 0, 1, \dots),$$

где a — среднее число событий, приходящихся на участок τ (рис. 86).

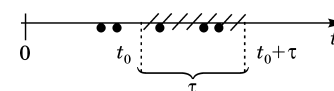


Рис. 86
Схема для пуассоновского потока событий

Для стационарного пуассоновского потока

$$a = \lambda\tau,$$

т. е. не зависит от того, где находится участок τ .

Для нестационарного пуассоновского потока

$$a = \int_{t_0}^{t_0+\tau} \lambda(t) dt.$$

В теории вероятностей доказано, что выполнение свойств 1–3 приводит к пуассоновскому закону распределения, таким образом, простейший поток всегда пуассоновский.

Рассмотрим простейший поток событий с интенсивностью λ . Найдем закон распределения T — *времени между событиями* (см. рис. 87).

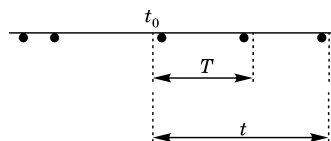


Рис. 87
Иллюстрация
для определения времени
попадания хотя бы одного
события потока

Сначала найдем функцию распределения случайной величины T :

$$F(t) = P(T < t),$$

($T < t$) будет тогда, когда на участок длины t попадет хотя бы одно событие

$$F(t) = 1 - P_0,$$

где P_0 — вероятность того, что не попадет ни одного события

$$\begin{aligned} P_0 &= \frac{a^0}{0!} e^{-a} = e^{-a} = e^{-\lambda t}; \\ F(t) &= 1 - e^{-\lambda t}, \quad (t > 0); \\ f(t) &= \frac{\partial F(t)}{\partial t} = \lambda e^{-\lambda t}, \quad (t > 0). \end{aligned} \quad (11.5)$$

Этот закон распределения называется *показательным*, λ — показатель *показательного закона* (рис. 88). Определим его параметры: математическое ожидание и дисперсию. Видно, что математическое ожидание для показательного закона распределения равно среднеквадратическому отклонению.

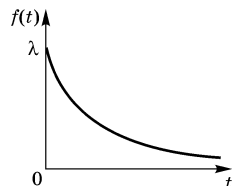


Рис. 88
График показательного
закона распределения
времени между двумя
соседними событиями
пуассоновского потока

$$\begin{aligned} m_t &= \int_0^{\infty} t f(t) dt = \lambda \int_0^{\infty} t e^{-\lambda t} dt = \frac{1}{\lambda}; \\ D_t &= \int_0^{\infty} t^2 f(t) dt - m_t^2 = \frac{1}{\lambda^2}; \\ \sigma_t &= \frac{1}{\lambda}; \\ m_t &= \sigma_t = \frac{1}{\lambda}. \end{aligned} \quad (11.6)$$

Последнее равенство используется для проверки того, является ли поток пуассоновским.

На рис. 89 показан вид функции плотности $f(t)$ при различных λ .

Вычислим вероятность появления интервалов, меньших математического ожидания $1/\lambda$:

$$P\left\{t < \frac{1}{\lambda}\right\} = \int_0^{\frac{1}{\lambda}} \lambda \cdot e^{-\lambda t} dt = 1 - \frac{1}{e} = 0,63;$$

т. е. в простейшем пуассоновском потоке короткие интервалы более часты, чем длинные, что создает более тяжелые условия работы, так как заявки группируются.

Для нестационарного потока закон распределения промежутка T уже не будет показательным, так как закон зависит от вида функции $\lambda(t)$.

Найдем вероятность того, что на участке Δt появится какое-то событие потока, т. е. участок не будет пустым. Так как поток ординарен, то

$$P_1(\Delta t) \approx 1 - P_0(\Delta t),$$

где $P_1(\Delta t)$ — вероятность того, что на Δt появится ровно одно событие; $P_0(\Delta t)$ — ни одного события

$$\begin{aligned} P_0(\Delta t) &= \frac{a^0}{0!} e^{-a} = e^{-a} = e^{-\lambda \Delta t}, \\ P_1(\Delta t) &\approx 1 - e^{-\lambda \Delta t}. \end{aligned}$$

Разложим $e^{-\lambda \Delta t}$ в ряд

$$\begin{aligned} P_1(\Delta t) &\approx 1 - (1 - \lambda \Delta t), \\ P_1(\Delta t) &\approx \lambda \Delta t. \end{aligned}$$

Для нестационарного потока

$$P_1(\Delta t) \approx \lambda(t) \Delta t.$$

Полученные формулы позволяют очень просто находить переходные вероятности (плотности) для составления уравнений Колмогорова (см. п. 11.3.).

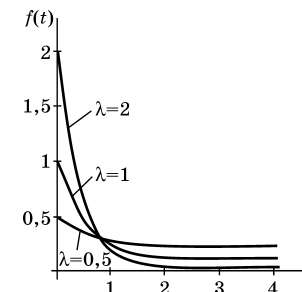


Рис. 89
Показательный закон
распределения при различных
параметрах λ

Кроме пуассоновских потоков в теории массового обслуживания рассматриваются потоки Пальма и Эрланга, являющиеся обобщением пуассоновских потоков.

Поток Пальма — это поток, в котором промежутки между последовательными событиями T_i есть независимые и одинаково распределенные случайные величины, таким образом если они распределены по показательному закону, то поток Пальма — пуассоновский поток.

Поток Эрланга k -го порядка получается из простейшего, если в простейшем сохранить каждую k точку, а остальные выбросить. Простейший поток — это поток Эрланга 1-го порядка.

В дальнейшем при изучении марковских процессов в системах с дискретными состояниями и непрерывным временем (непрерывные марковские цепи), нам удобно будет рассматривать переходы системы из состояния в состояние как результат воздействия каких-то потоков событий.

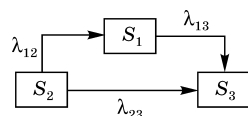


Рис. 90
Граф состояний для систем с пуассоновскими потоками переходов

Пусть в системе S переходы из состояния в состояние происходят под действием пуассоновских потоков с определенными интенсивностями (эти интенсивности ставят на графе состояний, рис. 90).

После этого можно составить систему уравнения Колмогорова, считая, что $P_i(\Delta t) \approx \lambda \Delta t$, т. е. λ_{ij} играет здесь роль плотностей. Система уравнения Колмогорова — это система линейных дифференциальных уравнений с постоянными (или с переменными, если $\lambda = \lambda(t)$) коэффициентами. Такие системы легко решаются с использованием ЭВМ.

11.5. ПРЕДЕЛЬНЫЕ ВЕРОЯТНОСТИ СОСТОЯНИЙ

Для практических расчетов нет необходимости знать функции $P_i(t)$. Достаточно знать так называемые *предельные вероятности* P_i .

Пусть имеется система S с дискретными состояниями S_1, S_2, \dots, S_n , в которых протекает марковский случайный

процесс с непрерывным временем. Пусть все интенсивности постоянны: $\lambda_{ij} = \text{const}$.

Записав и проинтегрировав систему уравнений Колмогорова, мы получим вероятности состояний как функции времени

$$P_1(t), P_2(t), \dots, P_n(t);$$

$$\forall t, \sum_{i=1}^n P_i(t) = 1.$$

Возникает вопрос, что будет с $P_i(t)$ при $t \rightarrow \infty$.

Пределы $\lim_{t \rightarrow \infty} P_i(t) = P_i$ (если они существуют) называются *предельными вероятностями состояний*.

Можно доказать следующее положение:

если число состояний системы конечно, из каждого состояния можно перейти (за конечное число шагов) в другое, то предельные вероятности состояний существуют и не зависят от начального состояния системы (рис. 91, 92).

Очевидно, что для предельных вероятностей должно быть:

$$\sum_{i=1}^n P_i = 1,$$

т. е. при $t \rightarrow \infty$ в системе устанавливается предельный стационарный режим. В этом режиме $P_i(t) = P_i$ не зависят от времени — это среднее относительное время системы в данном состоянии.

Если $P(S_1) = 0,2$; $P(S_2) = 0,3$; $P(S_3) = 0,5$, то: 20% времени система находилась в состоянии S_1 ,

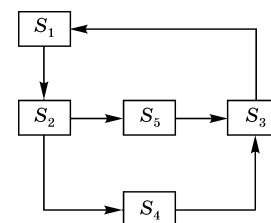


Рис. 91
Пример графа состояний для системы, имеющей предельные вероятности состояний (условие выполняется)

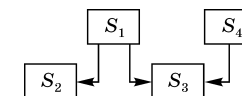


Рис. 92
Пример графа состояний для системы, не имеющей предельных вероятностей состояний (условие не выполняется)

30% времени система находилась в состоянии S_2 ,

50% времени система находилась в состоянии S_3 .

Как же вычислить предельные вероятности состояний?

Оказывается, для этого надо в уравнениях Колмогорова положить все левые части (производные) равными нулю, а P_i считать постоянными.

Действительно, в установившемся режиме $\frac{dP_i}{dt} = 0$, отсюда получим систему линейных алгебраических уравнений с нормировочным условием $\sum_{i=1}^n P_i = 1$.

Решив ее, найдем все предельные вероятности состояний. Покажем это для практически важных примеров моделей СМО.

11.6. ПРОЦЕСС ГИБЕЛИ И РАЗМНОЖЕНИЯ

Марковская непрерывная цепь называется процессом гибели и размножения, если граф состояний имеет вид, представленный на рис. 93.

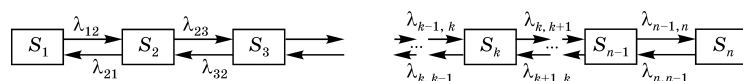


Рис.93
Схема процесса гибели и размножения

Пример. Каждый узел технической системы может выходить из строя и ремонтироваться. Тогда S_{i+1} — состояние, когда вышло из строя i узлов.

Напишем систему алгебраических уравнений для предельных вероятностей состояний:

для S_1 : $\lambda_{12}P_1 = \lambda_{21}P_2$;

для S_2 : $\lambda_{23}P_2 + \lambda_{21}P_2 = \lambda_{12}P_1 + \lambda_{32}P_3$.

Первое равенство можно сократить, получим

$$\lambda_{23}P_2 = \lambda_{32}P_3,$$

...

$$\lambda_{k-1,k}P_{k-1} = \lambda_{k,k-1}P_k$$

получим

$$\forall (k=2, \dots, n), \begin{cases} \lambda_{k-1,k}P_{k-1} = \lambda_{k,k-1}P_k, \\ P_1 + P_2 + \dots + P_n = 1. \end{cases}$$

Решая последовательно эту систему уравнений, получаем

$$\begin{aligned} P_2 &= \frac{\lambda_{12}}{\lambda_{21}} P_1; \\ P_3 &= \frac{\lambda_{23}}{\lambda_{32}} P_2 = \frac{\lambda_{23}}{\lambda_{32}} \cdot \frac{\lambda_{12}}{\lambda_{21}} P_1; \\ P_k &= \frac{\lambda_{k-1,k} \lambda_{k-2,k-1} \dots \lambda_{12}}{\lambda_{k,k-1} \lambda_{k-1,k-2} \dots \lambda_{21}} P_1; \\ k &= 2, \dots, n. \end{aligned} \quad (11.7)$$

Таким образом, все P_i выражены через P_1 . Подставим их в полученное условие

$$\sum_{i=1}^n P_i = 1,$$

получим

$$P_1 = \frac{1}{1 + \sum_{k=2}^n \frac{\lambda_{k-1,k} \lambda_{k-2,k-1} \dots \lambda_{12}}{\lambda_{k,k-1} \lambda_{k-1,k-2} \dots \lambda_{21}}}. \quad (11.8)$$

Остальные P_i выражаются через P_1 .

11.7. ЦИКЛИЧЕСКИЙ ПРОЦЕСС

Циклический процесс имеет граф состояний (рис. 94).

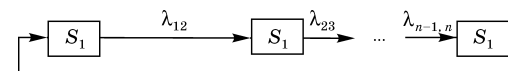


Рис. 94
Схема циклического процесса

Выведем предельные вероятности. Система алгебраических уравнений Колмогорова будет такой:

$$\left. \begin{aligned} \lambda_{23}P_2 &= \lambda_{12}P_1; \\ \lambda_{34}P_3 &= \lambda_{23}P_2; \\ &\dots \\ \lambda_{k,k+1}P_k &= \lambda_{k-1,k}P_{k-1}; \\ &\dots \\ \lambda_{n1}P_n &= \lambda_{12}P_1; \\ \sum_{i=1}^n P_i &= 1; \end{aligned} \right\} \begin{aligned} P_2 &= \frac{\lambda_{12}}{\lambda_{23}} P_1; \\ P_3 &= \frac{\lambda_{23}}{\lambda_{34}} P_2 = \frac{\lambda_{12}}{\lambda_{23}} \frac{\lambda_{23}}{\lambda_{34}} P_1 = \frac{\lambda_{12}}{\lambda_{34}} P_1; \\ &\dots \\ P_k &= \frac{\lambda_{12}}{\lambda_{k,k-1}} P_1. \end{aligned} \quad (11.9)$$

Подставляем формулу в $\sum_{i=1}^n P_i = 1$, получаем

$$P_1 = \frac{1}{1 + \lambda_{12} \left(\frac{1}{\lambda_{23}} + \frac{1}{\lambda_{34}} + \dots + \frac{1}{\lambda_{n1}} \right)} \quad (11.10)$$

и получаем P_k , $k = \overline{2, n}$.

Перейдем от интенсивностей λ_{ij} к среднему времени \bar{t}_i пребывания системы в состоянии S_i .

Из S_i в циклической схеме исходит одна стрелка. Найдем математическое ожидание времени T_i , в течение которого система пробудет в состоянии S_i . Так как процесс марковский, закон распределения T_i не зависит от того, сколько времени система уже была в состоянии S_i , т. е. закон такой же, если бы система только что пришла в состояние S_i . Поэтому промежуток T между событиями имеет показательный закон распределения. Параметр его $\lambda_{i,i+1}$, тогда

$$\bar{t}_i = \frac{1}{\lambda_{i,i+1}},$$

$$\lambda_{i,i+1} = \frac{1}{\bar{t}_i}, \quad \forall i.$$

В силу цикличности $\lambda_{n1} = \frac{1}{\bar{t}_n}$, тогда подставив это в формулы для вероятностей, получим

$$P_k = \frac{\bar{t}_k}{\sum_{i=1}^n \bar{t}_i}, \quad k = \overline{1, n}, \quad (11.11)$$

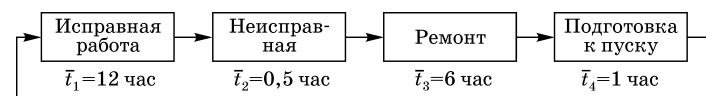


Рис. 95

Пример графа состояний для СМО «Газонокосилка»

т. е. предельные вероятности состояний в циклической схеме относятся как среднее время пребывания системы подряд в каждом из состояний.

Пример графа состояний для СМО «Газонокосилка» изображен на рис. 95.

11.8. КЛАССИФИКАЦИЯ СМО

Различают два основных типа СМО.

1. Системы с отказами. Здесь заявки, попадающие в момент, когда все каналы заняты, получают «отказ», покидают СМО и в обслуживании далее не участвуют.

2. Системы с ожиданием (с очередью). Заявки в случае занятости каналов становятся в очередь и ожидают, пока не освободится какой-либо канал. Затем после освобождения какая-либо заявка обслуживается (не обязательно первоочередная).

Системы с ожиданием бывают *упорядоченные* (т. е. обслуживание происходит в порядке очереди) и *неупорядоченные* (в случайном порядке). Бывают СМО с *приоритетом*, когда более важные заявки обслуживаются в первую очередь.

СМО с ожиданием делятся на системы с *неограниченным* и с *ограниченным ожиданием*. В первой любые заявки, поступившие в СМО, будут рано или поздно обслужены, во второй есть предел для ожидания.

В зависимости от типов СМО применяются различные критерии эффективности их работы:

1. В СМО с отказами:

- *абсолютная пропускная способность* — среднее число заявок, которые может обслужить СМО за единицу времени Q ;

- *относительная пропускная способность СМО* — средняя доля поступивших заявок, обслуживаемая системой

$$q = \frac{\text{обслуженные}}{\text{все заявки (обслуженные + необслуженные)}}.$$

Также рассматривается *среднее число занятых каналов и среднее относительное время простоя системы и каналов*.

2. Для СМО с неограниченным ожиданием абсолютная и относительная пропускная способность теряет смысл, так как любая заявка когда-нибудь все равно обслуживается. Вместо них используются:

- среднее число заявок в очереди;
- среднее число заявок в системе (в очереди + обслуживаемые);
- среднее время ожидания в очереди;
- среднее время пребывания заявки в системе и др.

Для СМО с ограниченным ожиданием все указанные характеристики могут быть использованы.

Подводя итог описанию процессов, происходящих в СМО, можно сказать, что для анализа процесса в СМО нужно знать:

- n — число каналов;
- интенсивность потока заявок;
- производительность каналов (среднее число обслуживаемых в единицу времени заявок);
- условия образования очереди.

Используем полученные формулы для анализа конкретных СМО. При этом все потоки считаем простейшими.

11.9.

ОДНОКАНАЛЬНАЯ СМО С ОТКАЗАМИ

Это простейшая СМО, в котором $n = 1$. В общем случае $\lambda = \lambda(t)$. Заявка в случае занятости канала покидает СМО. Обслуживание заявки в общем случае продолжается в течение случайного времени обслуживания $T_{об}$, распределенного по показательному закону с параметром μ :

$$f(t) = \mu e^{-\mu t}, \quad t > 0.$$

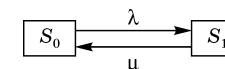
Таким образом, поток обслуживаний также простейший. Требуется найти:

- 1) абсолютную пропускную способность СМО Q ;
- 2) относительную пропускную способность q .

Канал обслуживания может находиться в двух состояниях: S_0 — свободен, S_1 — занят.

Граф состояний изображен на рис. 96.

Рис.96
Граф состояний для одноканальной СМО с отказами



Очевидно, что $P_0(t) + P_1(t) = 1$.

Тогда с учетом этого запишем единственное дифференциальное уравнение Колмогорова (второе будет лишним):

$$\begin{aligned} \frac{dP_0}{dt} &= -\lambda P_0 + \mu P_1; \\ \frac{dP_1}{dt} &= -(\mu + \lambda)P_1 + \lambda P_0. \end{aligned}$$

Пусть $P_0(0) = 1$, $P_1(0) = 0$. Таким образом, в $t = 0$ заявок нет.

Решим дифференциальное уравнение аналитически

$$\begin{aligned} P_0 &= uv; \\ u'v + v'u &= -(\mu + \lambda)uv + \mu; \\ v[u' + (\mu + \lambda)u] + v'u - \mu &= 0; \\ u' + (\mu + \lambda)u &= 0; \\ \frac{du}{dt} &= -(\mu + \lambda)u; \quad \frac{du}{u} = -(\mu + \lambda)dt; \\ \ln u &= -(\mu + \lambda)t; \\ u &= e^{-(\mu + \lambda)t}; \\ v'e^{-(\mu + \lambda)t} &= \mu; \\ dv &= \mu e^{(\mu + \lambda)t} dt; \quad v = \mu \frac{1}{\mu + \lambda} e^{(\mu + \lambda)t} + c; \\ P_0 &= e^{-(\mu + \lambda)t} \left(\mu \frac{1}{\mu + \lambda} e^{(\mu + \lambda)t} + c \right); \quad \text{н. у. } P_0(0) = 1; \\ 1 &= c + \frac{\mu}{\mu + \lambda}; \quad c = \frac{\lambda}{\mu + \lambda}. \end{aligned}$$

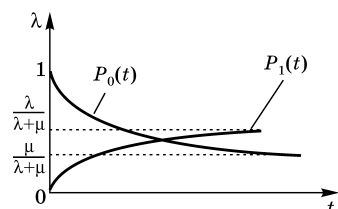


Рис. 97
Графики вероятностей состояний для одноканальной СМО с отказами

Окончательно получаем искомые вероятности как функции времени в виде

$$P_0 = \frac{\mu}{\mu + \lambda} + \frac{\lambda}{\mu + \lambda} e^{-(\mu + \lambda)t};$$

$$P_1 = 1 - P_0 = \frac{\lambda}{\mu + \lambda} - \frac{\lambda}{\mu + \lambda} e^{-(\mu + \lambda)t}.$$

Эта зависимость имеет вид, показанный на рис. 97.

Из графика видно, что предельные вероятности P_0 и P_1 — это значения, к которым стремятся $P_0(t)$ и $P_1(t)$, таким образом

$$P_0 = \frac{\mu}{\mu + \lambda}; \quad P_1 = \frac{\lambda}{\mu + \lambda}.$$

Очевидно, предельные вероятности можно получить, если сразу составить систему алгебраических уравнений Колмогорова.

P_0 — есть вероятность того, что заявка, пришедшая в момент t , будет обслужена, следовательно, это и есть относительная пропускная способность

$$q = P_0,$$

$$q = \frac{\mu}{\lambda + \mu}.$$

Вероятность отказа — это в данном случае P_1 .

Так как только q -я доля из пришедших λ заявок обслуживается системой, то абсолютная пропускная способность Q получается в виде

$$Q = \lambda q; \quad Q = \frac{\lambda \mu}{\lambda + \mu}.$$

Очевидно, что вероятность отказа будет

$$P_{\text{отк}} = 1 - \frac{\mu}{\lambda + \mu} = \frac{\lambda}{\lambda + \mu}.$$

11.10. МНОГОКАНАЛЬНАЯ СМО С ОТКАЗАМИ

Пусть число каналов СМО равно n . Перечислим все состояния СМО:

S_0 — все каналы свободны;

S_1 — занят один канал;

...

S_k — занято k каналов, $((n - k) — свободны);$

...

S_n — заняты все n каналов.

Граф состояний будет иметь вид, показанный на рис. 98.

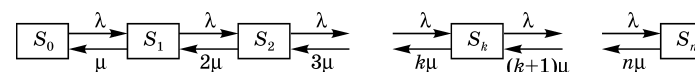


Рис. 98

Граф состояний для многоканальной СМО с отказами

По стрелкам слева направо систему переводит один и тот же поток с интенсивностью λ . Если система была в S_k (занято k каналов), то любая заявка переводит ее в S_{k+1} .

По стрелкам справа налево $S_1 \rightarrow S_0$ после окончания обслуживания одного канала система перейдет в S_0 . Если занято k каналов, то общий поток обслуживаний будет в k раз интенсивней.

Это частный случай процесса «гибели и размножения» (см. п. 11.6).

Составим систему уравнений Колмогорова:

$$\left. \begin{aligned} \frac{dP_0}{dt} &= -\lambda P_0 + \mu P_1; \\ \frac{dP_1}{dt} &= -(\lambda + \mu)P_1 + \lambda P_0 + 2\mu P_2; \\ &\dots \\ \frac{dP_k}{dt} &= -(\lambda + k\mu)P_k + \lambda P_{k-1} + (k+1)\mu P_{k+1}; \\ &\dots \\ \frac{dP_n}{dt} &= -n\mu P_n + \lambda P_{n-1}. \end{aligned} \right\} \quad (11.12)$$

Система (11.12) называется *уравнениями Эрланга*.

обозначив $\frac{\lambda}{\mu} = \rho$, на основе формулы для схемы гибели и размножения (11.7), (11.8), получаем

$$\begin{cases} P_1 = \frac{\rho}{1!} P_0; & P_2 = \frac{\rho^2}{2!} P_0; & \dots & P_n = \frac{\rho^n}{n!} P_0; \\ P_{n+1} = \frac{\rho^{n+1}}{n \cdot n!} P_0; & P_{n+2} = \frac{\rho^{n+2}}{n^2 \cdot n!} P_0; & \dots & P_{n+m} = \frac{\rho^{n+m}}{n^m \cdot n!} P_0; \end{cases}$$

$$P_0 = \left[\left(1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} \right) + \underbrace{\frac{\rho^{n+1}}{n \cdot n!} + \frac{\rho^{n+2}}{n^2 \cdot n!} + \dots + \frac{\rho^{n+m}}{n^m \cdot n!}}_{\text{геометрическая прогрессия}} \right]^{-1}. \quad (11.14)$$

По формуле суммы членов геометрической прогрессии со знаменателем q

$$S = \frac{a_1(1-q^m)}{1-q},$$

$$P_0 = \left[1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} \cdot \frac{\frac{\rho}{n} - \left(\frac{\rho}{n}\right)^{m+1}}{1 - \frac{\rho}{n}} \right]^{-1}. \quad (11.15)$$

Все вероятности состояний найдены. Найдем некоторые характеристики такой СМО:

1. Отказ будет тогда, когда заняты все каналы и нет места в очереди

$$P_{\text{отк}} = P_{n+m} = \frac{\rho^{n+m}}{n^m \cdot n!} P_0.$$

2. Относительная пропускная способность q дополняет вероятность отказа до единицы

$$q = 1 - P_{\text{отк}} = 1 - \frac{\rho^{n+m}}{n^m \cdot n!} P_0.$$

3. Абсолютная пропускная способность будет

$$Q = \lambda \cdot q = \lambda \cdot \left(1 - \frac{\rho^{n+m}}{n^m \cdot n!} P_0 \right). \quad (11.16)$$

4. Найдем *среднее число заявок, связанных с системой*. Сначала найдем *среднее число занятых каналов*. Для СМО с отказами оно совпадало со средним числом заявок, находящихся в системе. Для нашего случая эти величины отличаются на количество заявок, находящихся в очереди.

Пусть \bar{Z} — количество заявок, связанных с системой; \bar{K} — среднее число занятых каналов:

$$\bar{K} = \frac{Q}{\mu};$$

$$\bar{K} = \rho \cdot \left(1 - \frac{\rho^{n+m}}{n^m \cdot n!} P_0 \right).$$

Среднее число заявок в очереди получим как математическое ожидание:

$$\begin{aligned} \bar{r} &= \sum_{r=1}^m r P_{n+r} = \\ &= \frac{\rho^{n+1} \cdot P_0}{n \cdot n!} \left[1 + 2 \frac{\rho}{n} + 3 \left(\frac{\rho}{n} \right)^2 + \dots + m \left(\frac{\rho}{n} \right)^{m-1} \right]; \\ \frac{\rho}{n} &= \chi; \\ \bar{r} &= \frac{\rho^{n+1}}{n \cdot n!} P_0 [1 + 2\chi + 3\chi^2 + \dots + m\chi^{m-1}]; \\ [1 + 2\chi + 3\chi^2 + \dots + m\chi^{m-1}] &= \frac{d}{d\chi} (\chi + \chi^2 + \chi^3 + \dots + \chi^m) = \\ &= \frac{d}{d\chi} \left(\frac{\chi - \chi^{m+1}}{1 - \chi} \right) = \frac{1 - \chi^m(m+1 - m\chi)}{(1 - \chi)^2}; \\ \bar{r} &= \frac{\rho^{n+1}}{n \cdot n!} P_0 \frac{1 - (m+1)\chi^m + m\chi^{m+1}}{(1 - \chi)^2}. \end{aligned}$$

Очевидно, среднее число заявок в системе будет $\bar{Z} = \bar{K} + \bar{r}$.

5. Среднее время ожидания заявки в очереди. Здесь рассмотрим ряд гипотез:

- заявка пришла, и очереди нет, есть свободный канал, $t_{\text{ож}} = 0$;

- заявка пришла, и точно все n каналов заняты, очереди нет. Заявке придется ждать время $\frac{1}{n\mu}$, так как поток освобождений n каналов имеет интенсивность $n\mu$;
- если заявка застанет все n каналов занятыми и одну заявку в очереди, то ей придется ждать

$$\frac{2}{n\mu} = \frac{1}{n\mu} + \frac{1}{n\mu} \text{ и т. д.}$$

Если r заявок в очереди, то $\frac{r}{n\mu}$;

- если в очереди m заявок, то она вообще не будет ждать:

$$\begin{aligned} \bar{t}_{\text{ож}} &= \frac{1}{n\mu} P_n + \frac{2}{n\mu} P_{n+1} + \dots + \frac{m}{n\mu} P_{n+m-1} = \\ &= \frac{\rho^n P_0}{n \cdot n! \mu} \left[1 + \frac{2\rho}{n} + \frac{3\rho^2}{n^2} + \dots + \frac{m\rho^{m-1}}{n^{m-1}} \right]; \\ \bar{t}_{\text{ож}} &= \frac{\bar{r}}{\lambda}; \\ \bar{t}_{\text{ож}} &= \frac{\rho^n P_0}{n \cdot n! \mu} \cdot \frac{1 - (m+1)\chi^m + m\chi^{m+1}}{(1-\chi)^2}. \end{aligned}$$

6. Среднее время пребывания заявки в системе

$$\begin{aligned} \bar{t}_{\text{сист}} &= \bar{t}_{\text{ож}} + \frac{\text{относительная пропускная способность}}{\text{интенсивность обслуживания}}; \\ \bar{t}_{\text{сист}} &= \bar{t}_{\text{ож}} + \frac{q}{\mu} \left(\frac{1}{\mu} - \text{среднее время обслуживания} \right). \end{aligned}$$

Действительно, так как заявки обрабатываются параллельно, то, если одна заявка пришла, она обрабатывается за время $\frac{1}{\mu}$, но среди всех только q -я часть обрабатывается действительно, поэтому $\frac{q}{\mu}$.

Таким образом, мы получили формулы для многоканальной СМО с ожиданием:

$$\begin{aligned} 1. P_{\text{отк}} &= \frac{\rho^{n+m}}{n^m \cdot n!} P_0. \\ 2. q &= 1 - P_{\text{отк}}. \\ 3. Q &= \lambda \cdot \left(1 - \frac{\rho^{n+m}}{n^m \cdot n!} P_0 \right). \\ 4. \bar{Z} &= \rho \cdot \left(1 - \frac{\rho^{n+m}}{n^m \cdot n!} P_0 \right). \\ 5. \bar{r} &= \frac{\rho^{n+1}}{n \cdot n!} P_0 \cdot \frac{1 - (m+1)\chi^m + m\chi^{m+1}}{(1-\chi)^2}. \\ 6. \bar{K} &= \bar{Z} + \bar{r}. \\ 7. \bar{t}_{\text{ож}} &= \frac{\rho^n}{n \cdot \mu \cdot n!} P_0 \cdot \frac{1 - (m+1)\chi^m + m\chi^{m+1}}{(1-\chi)^2} = \frac{\bar{r}}{\lambda}. \\ 8. t_{\text{сист}} &= \bar{t}_{\text{ож}} + \frac{q}{\mu}. \\ P_0 &= \left[1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \dots + \frac{\rho^n}{n!} \cdot \frac{\left(\frac{\rho}{n}\right) - \left(\frac{\rho}{n}\right)^{m+1}}{1 - \left(\frac{\rho}{n}\right)} \right]^{-1}. \end{aligned} \quad (11.17)$$

11.12. ОДНОКАНАЛЬНАЯ СМО С ОЖИДАНИЕМ

Очевидно, что одноканальная СМО с ожиданием есть частный случай только что рассмотренной многоканальной СМО с ожиданием. Граф состояний имеет вид, показанный на рис. 100. Заметим, что вероятности состояний для одноканальной системы можно получить, используя формулы (11.14) и (11.15):

$$n=1: P_k = \rho^k P_0; P_0 = [1 + \rho + \rho^2 + \dots + \rho^{m+1}]^{-1} = \frac{1 - \rho}{1 - \rho^{m+2}}.$$

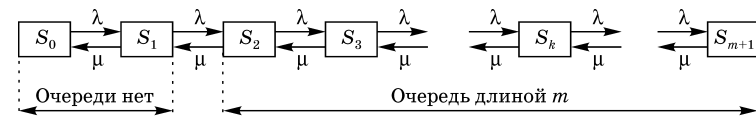


Рис. 100

Граф состояний для одноканальной СМО с очередью

Число мест в очереди ограничено m . Тогда вероятности всех состояний имеют вид

$$\left. \begin{array}{l} P_1 = \rho \cdot P_0 \\ P_2 = \rho^2 \cdot P_0 \\ \dots \\ P_k = \rho^k \cdot P_0 \\ \dots \\ P_{m+1} = \rho^{m+1} \cdot P_0 \end{array} \right| \begin{array}{l} P_0 = [1 + \rho + \rho^2 + \dots + \rho^{m+1}]^{-1} \\ P_0 = \frac{1 - \rho}{1 - \rho^{m+2}} \end{array} \quad (11.18)$$

Получим характеристики такой СМО:

1. Вероятность отказа:

$$P_{\text{отк}} = P_{m+1} = \frac{\rho^{m+1}(1-\rho)}{1-\rho^{m+2}}. \quad (11.19)$$

2. Относительная пропускная способность:

$$q = 1 - P_{\text{отк}} = 1 - \frac{\rho^{m+1}(1-\rho)}{1-\rho^{m+2}}. \quad (11.20)$$

3. Абсолютная пропускная способность:

$$Q = \lambda \cdot q.$$

4. Среднее число заявок в очереди найдем из формулы для многоканальной СМО при $n = 1$. Так как $\frac{\rho}{n} = \chi$, при $n = 1$ $\chi = \rho$,

$$\bar{r} = \frac{\rho^2(1-\rho)}{1-\rho^{m+2}} \cdot \frac{1-\rho^m(m+1-m\rho)}{(1-\rho)^2} = \frac{\rho^2[1-\rho^m(m+1-m\rho)]}{(1-\rho^{m+2})(1-\rho)}. \quad (11.21)$$

5. Среднее число заявок под обслуживанием найдем из предыдущих формул при $n = 1$:

$$\bar{K} = \rho \cdot \left(1 - \frac{\rho^{n+m}}{n^m \cdot n!} P_0 \right) = \rho \cdot \left(1 - \frac{\rho^{m+1} \cdot (1-\rho)}{1-\rho^{m+2}} \right) = \frac{\rho - \rho^{m+2}}{1-\rho^{m+2}}. \quad (11.22)$$

6. Среднее число заявок, связанных с СМО:

$$\bar{Z} = \bar{r} + \bar{K}.$$

7. Среднее время ожидания заявки в очереди.

Для многоканальной СМО

$$\bar{t}_{\text{ож}} = \frac{\rho^n P_0}{n\mu n!} \cdot \frac{1-(m+1)\chi^m + m\chi^{m+1}}{(1-\chi)^2}.$$

При $n = 1$ $\chi = \rho$,

$$\begin{aligned} \bar{t}_{\text{ож}} &= \frac{\rho(1-\rho)}{\mu(1-\rho^{m+2})!} \cdot \frac{1-(m+1)\rho^m + m\rho^{m+1}}{(1-\rho)^2} = \\ &= \frac{\rho}{\mu(1-\rho^{m+2})!} \cdot \frac{1-\rho^m(m+1-m\rho)}{(1-\rho)}. \end{aligned} \quad (11.23)$$

Сравнив это выражение с выражением для \bar{r} , видим, что

$$\bar{t}_{\text{ож}} = \frac{1}{\rho \cdot \mu} \bar{r} = \frac{\bar{r}}{\lambda}, \quad (11.24)$$

т. е. среднее время ожидания равно среднему числу заявок в очереди, деленному на интенсивность потока заявок,

$$\bar{t}_{\text{сист}} = \bar{t}_{\text{ож}} + \frac{q}{\mu} = \frac{\bar{r}}{\lambda} + \frac{q}{\mu}.$$

11.13. СМО С ОЖИДАНИЕМ И БЕСКОНЕЧНОЙ ОЧЕРЕДЬЮ

Мы рассмотрели СМО с ожиданием и очередью длиной m . Снимем это ограничение и устремим $m \rightarrow \infty$. Граф состояний будет бесконечным. Получим вероятности состояний одноканальной СМО с бесконечной очередью. Для этого в выражениях (11.18)–(11.23) положим $m \rightarrow \infty$. Видим, что в выражении для P_0 сумма будет конечной, только если $\rho < 1$. Таким образом, при $\rho \geq 1$ не существует предельного установившегося режима, и при $t \rightarrow \infty$ очередь стремится к бесконечности.

Пусть $\rho = \frac{\lambda}{\mu} < 1$ и предельный режим существует, тогда (используя сумму убывающей геометрической прогрессии) получим

$$\left. \begin{aligned}
 P_0 &= 1 - \rho, \\
 P_1 &= \rho \cdot (1 - \rho), \\
 &\dots \\
 P_k &= \rho^k \cdot (1 - \rho). \\
 \text{Каждая заявка будет обслужена, поэтому} \\
 1. P_{\text{отк}} &= 0, \\
 2. q &= 1, \\
 3. Q &= \lambda \cdot q = \lambda, \\
 4. \bar{r}_{\infty} &= \frac{\rho^2}{1 - \rho}, \\
 5. \bar{K}_{\infty} &= \rho, \\
 6. \bar{Z}_{\infty} &= \bar{r}_{\infty} + \rho = \frac{\rho^2}{1 - \rho} + \rho = \rho \left(\frac{\rho}{1 - \rho} + 1 \right) = \frac{\rho}{1 - \rho}, \\
 7. \bar{t}_{\text{ож}} &= \frac{1}{\mu} \cdot \frac{\rho}{1 - \rho}, \\
 8. \bar{t}_{\text{сист}} &= \frac{1}{\mu} \cdot \frac{\rho}{1 - \rho} + \frac{1}{\mu} = \frac{1}{\mu} \cdot \frac{1}{1 - \rho}.
 \end{aligned} \right\} (11.25)$$

11.14. ЗАМКНУТЫЕ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ

Мы рассмотрели СМО, когда заявки поступали извне. Это *разомкнутые (открытые)* СМО. Иногда потоки заявок зависят от состояния самой СМО. Рассмотрим пример такой *замкнутой* СМО.

Бригада из m рабочих обслуживает n станков ($m < n$). При выходе из строя станка один из рабочих его обслуживает (если есть свободные рабочие). Если вышли из строя $(m + 1)$ станков, то один станок не может быть немедленно обслужен, он становится в очередь. Считаем что распределение времени обслуживания — пуассоновское. Рабочий в среднем тратит время на обслуживание

$$\bar{t}_{\text{об}} = \frac{1}{\mu},$$

где μ — интенсивность потока обслуживаний.

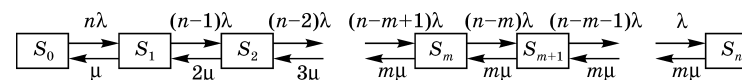


Рис. 101
Граф состояний для замкнутой СМО

Пусть λ — интенсивность потока неисправностей одного станка. Перечислим состояния:

S_0 — все станки работают, рабочие свободны;

S_1 — один станок неисправен, один рабочий его ремонтирует;

...

S_m — m станков неисправны, m рабочих их ремонтируют;

...

S_n — n станков неисправны, m станков ремонтируются, а $n - m$ станков ожидают ремонта.

Граф состояний будет иметь вид, показанный на рис. 101.

Видим, что это схема гибели и размножения. Используя общее решение (11.7), (11.8), получим предельные вероятности состояний.

$$P_1 = \frac{n}{1} \rho \cdot P_0; \quad P_2 = \frac{n(n-1)}{2!} \rho^2 \cdot P_0; \quad P_3 = \frac{n(n-1)(n-2)}{3!} \rho^3 \cdot P_0;$$

...

$$P_m = \frac{n(n-1)\dots(n-m+1)}{m!} \rho^m \cdot P_0;$$

$$P_{m+1} = \frac{n(n-1)\dots(n-m)}{m \cdot m!} \rho^{m+1} \cdot P_0;$$

$$P_{m+2} = \frac{n(n-1)\dots(n-m-1)}{m^2 \cdot m!} \rho^{m+2} \cdot P_0;$$

...

$$P_n = \frac{n(n-1)\dots 1}{m^{n-m} \cdot m!} \rho^n \cdot P_0;$$

$$P_0 = \left[1 + \frac{n}{1!} \rho + \frac{n(n-1)}{2!} \rho^2 + \dots + \frac{n(n-1)\dots(n-m+1)}{m!} \rho^m + \right. \\
 \left. + \frac{n(n-1)\dots(n-m)}{m \cdot m!} \rho^{m+1} + \dots + \frac{n(n-1)\dots 1}{m^{n-m} \cdot m!} \rho^n \right]^{-1}.$$

Вычислим среднее число занятых рабочих

$$\bar{K} = 0 \cdot P_0 + 1 \cdot P_1 + 2 \cdot P_2 + \dots + m \cdot (P_m + P_{m+1} + \dots + P_n);$$

$$\bar{K} = P_1 + 2 \cdot P_2 + \dots + (m-1) \cdot P_{m-1} + m \cdot (1 - P_0 - P_1 - \dots - P_{m-1}).$$

Среднее число неисправностей, обслуживаемое бригадой в единицу времени (абсолютная пропускная способность), получается как произведение среднего числа занятых рабочих на количество обслуживаемых заявок в единицу времени

$$Q = \bar{K} \cdot \mu.$$

Среднее число неисправных станков, — это среднее число станков, связанных с процессом обслуживания, — ϖ (аналогично параметру \bar{Z}). Вычислим ϖ непосредственно как математическое ожидание

$$\varpi = 1 \cdot P_1 + 2 \cdot P_2 + \dots + m \cdot P_m + (m+1) \cdot P_{m+1} + \dots + n \cdot P_n.$$

Но можно получить другую формулу, не требующую знания всех вероятностей.

Действительно, количество работающих станков ($n - \varpi$). Эти работающие станки создают интенсивность потока неисправностей $(n - \varpi)\lambda$. В стационарном режиме эти неисправности устраняются рабочими, которые устраняют Q неисправностей в единицу времени. Значит,

$$(n - \varpi)\lambda = Q = \bar{K} \cdot \mu;$$

$$n - \varpi = \frac{\bar{K} \cdot \mu}{\lambda};$$

$$\varpi = n - \frac{\bar{K} \cdot \mu}{\lambda} = n - \frac{\bar{K}}{\rho}.$$

Среднее число станков, ожидающих в очереди

$$\bar{r} = \varpi - \bar{K}.$$

Вероятность того, что рабочие не будут заняты

$$P_{\text{своб}} = P_0.$$

11.15. СМО С ВЗАИМОПОМОЩЬЮ МЕЖДУ КАНАЛАМИ

Иногда можно построить СМО, где незанятые каналы помогают занятому в обслуживании. Это может быть и в открытых, и в замкнутых системах массового обслуживания.

При анализе операций в подобных СМО необходимо учитывать следующее:

- насколько убыстряется обслуживание при взаимопомощи;
- какой регламент (алгоритм) взаимопомощи дает наилучшие результаты.

Если над обслуживанием заявки работает не один канал, а k каналов, то интенсивность потока обслуживания будет увеличиваться с ростом k . Обозначим эту функцию $\mu(k)$.

Очевидно, что неограниченное увеличение числа помогающих каналов не всегда ведет к пропорциональному увеличению скорости обслуживания. При некотором $k = k_{\text{крит}}$ дальнейшее увеличение числа занятых каналов бессмысленно (рис. 102).

Для анализа поэтому надо знать $\mu(k)$.

Пусть $\mu(k) = ak$ для $k \leq k_{\text{крит}}$;

$\mu(k) = \mu_{\text{max}}$ для $k > k_{\text{крит}}$.

Если $n \leq k_{\text{крит}}$, то интенсивность обслуживания заявки несколькими каналами пропорциональна числу каналов (рис. 102б).

Обратимся к анализу алгоритмов взаимопомощи.

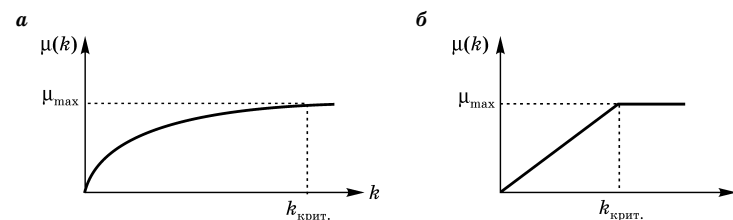


Рис. 102

Функции интенсивности обслуживания при взаимопомощи:

а — нелинейная; б — линейная.

Самый простой алгоритм «все как один» на заявку. В этом случае все n каналов обрабатывают появившуюся заявку до полного обслуживания, затем вторую и т. д. В этом случае СМО как бы становится одноканальной, только с более высокой интенсивностью обслуживания.

Выгодно ли это? Это зависит от $\mu(k)$, от типа СМО (с отказами, с очередью) и от критериальной величины.

Очевидно, для систем с отказами среднее время пребывания заявки в СМО уменьшается. Это хорошо, зато увеличивается вероятность отказа, так как все каналы заняты, если хотя бы одна заявка пришла.

Для системы с ожиданием также получаем одноканальную СМО с увеличенной интенсивностью обслуживания. Здесь среднее время пребывания заявки в системе меньше, но средняя длина очереди и среднее время ожидания в очереди может получиться больше.

Таким образом, алгоритм «все как один» не очень хорош. Желательно сделать так, чтобы взаимопомощь между каналами не мешала принимать и обслуживать новые заявки, появившиеся за время, пока все каналы были заняты.

Назовем распределенной взаимопомощью следующий алгоритм.

В момент прихода первой заявки за обслуживание принимаются все n каналов, если приходит еще одна, то часть каналов переключается на нее, если еще одна, то часть на нее и т. д. до тех пор, пока не окажутся занятыми n каналов.

Далее происходит отказ в СМО с отказами или заявки становятся в очередь в СМО с очередями. Здесь нет простоя, если хотя бы одна заявка поступила.

Возникает вопрос: какую часть каналов надо перераспределять на другие поступившие заявки. Это зависит от вида $\mu(k)$. Если $\mu(k)$ линейна и $k_{\text{крит}} > n$, то все равно какую часть каналов выделить на обслуживание вновь поступившей заявки, лишь бы все каналы были заняты (суммарная интенсивность обслуживания все равно будет $n\mu$). В общем случае для нахождения оптимального распределения каналов по заявкам нужно решать задачу

$$\left. \begin{aligned} \sum_{i=1}^l \mu(k_i) &\rightarrow \max; \\ \sum_{i=1}^l k_i &= n. \end{aligned} \right\} \quad (11.26)$$

Эта задача является частным случаем задачи распределения ресурсов (см. п. 5.4) и, как известно, имеет теоретическое решение.

Если кривая $\mu(k)$ неубывающая и выпуклая вверх, то нужно распределять каналы по заявкам как можно более равномерно, если выпуклая вниз, то наиболее неравномерно — все работают над одной заявкой. (Если зависимость линейная, то все равно.)

Рассмотрим работу n -канальной СМО при равномерной взаимопомощи между каналами.

11.15.1. ВЗАИМОПОМОЩЬ В СМО С ОТКАЗАМИ

- S_0 — СМО свободна;
 S_1 — 1 заявка обслуживается всеми n каналами;
 S_2 — 2 заявки обслуживаются всеми n каналами;
 ...
 S_k — k заявок обслуживаются всеми n каналами;
 ...
 S_n — n заявок обслуживаются всеми n каналами.
 Граф состояний будет иметь вид, показанный на рис. 103.

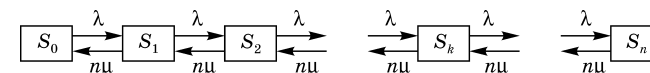


Рис. 103

Граф состояний для СМО с отказами и взаимопомощью

То есть граф такой же, как для одноканальной СМО с $\mu^* = n\mu$ и ограниченной очередью, имеющей $(n - 1)$ мест (см. п. 11.12). Поэтому можно подставить в полученные ранее для этой СМО формулы вместо ρ

$$\rho^* = \frac{\lambda}{\mu^*} = \frac{\lambda}{n\mu}$$

и вместо m

$$m^* = n - 1.$$

11.15.2. ВЗАИМОПОМОЩЬ В СМО С ОЧЕРЕДЬЮ

Рассмотрим СМО с очередью и максимальным числом мест в очереди m . Пусть между каналами равномерная взаимопомощь и $\mu(k) = k\mu$.

S_0 — СМО свободна;

S_1 — 1 заявка обслуживается всеми n каналами;

S_2 — 2 заявки обслуживаются всеми n каналами;

...

S_k — k заявок обслуживаются всеми n каналами, очереди нет;

...

S_n — n заявок обслуживаются всеми n каналами, очереди нет;

...

S_{n+1} — n заявок обслуживаются всеми n каналами, 1 заявка стоит в очереди;

...

S_{n+m} — n заявок обслуживаются всеми n каналами, в очереди m заявок.

Граф состояний будет иметь вид, показанный на рис. 104.



Рис. 104

Граф состояний для СМО с очередью и взаимопомощью

Мы получили граф того же вида, что и в предыдущем случае, но с увеличенным на m числом состояний. То есть здесь также можно воспользоваться формулами для одноканальной СМО с ограниченной очередью с $\mu^* = n\mu$ и числом мест в очереди $m^* = n + m - 1$. Если записать все характеристики, то увидим, что для равномерной взаимопомощи все характеристики изменяются в желательном для нас направлении. Так как интенсивность и число мест в очереди увеличиваются, то $P_{\text{отк}}$ уменьшается, а q и Q увеличиваются.

Однако организация такой дисциплины взаимопомощи осуществима не для всех СМО.

11.16. СМО С ОШИБКАМИ

На практике часто удобно использовать модель СМО, в которой заявки обслуживаются не с полной достоверностью, а с вероятностью $p \neq 1$, т. е. некоторые заявки в действительности не обслуживаются (брак в работе СМО, который нельзя принципиально устранить).

Например, СМО — операция по отражению налетов противника, справочное бюро, телефонная станция и т. п.

В разомкнутой СМО с ошибками появление ошибки на обслуживании практически не сказывается, так как число источников заявок очень велико и интенсивность потока заявок из-за ошибок практически не меняется. Поэтому, чтобы учесть ошибки в разомкнутой системе, нужно умножить на p относительную пропускную способность q , где p — вероятность безошибочного обслуживания. Соответственно $Q^* = Qp$. Другие характеристики, такие как время ожидания в очереди, число заявок в очереди, очевидно, не меняются.

Для замкнутой системы заявка, обслуженная с ошибкой, вновь становится в очередь на обслуживание и увеличивает время обслуживания всех заявок.

Пример.

Рассмотрим замкнутую СМО с ошибками. Пусть один рабочий обслуживает n станков. Интенсивность потока неисправностей одного станка λ , среднее время наладки

$\bar{t}_{\text{об}} = \frac{1}{\mu}$. С вероятностью p обслуживание удачно. С вероят-

ностью $(1 - p)$ обслуживание с браком. Требуется определить предельные вероятности состояний.

Здесь наличие ошибки приводит к тому, что $\mu^* = p\mu$, так как число успешных обслуживаний за единицу времени будет именно таким.

Получим граф состояний (см. рис. 105).

Мы получили граф состояний, как в одноканальной замкнутой СМО.

По полученным ранее формулам можно получить предельные вероятности.

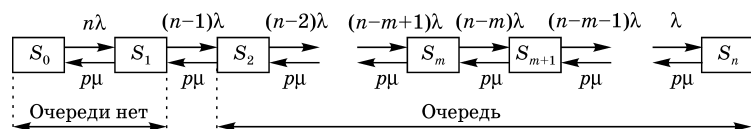


Рис. 105

Граф состояний для СМО с ошибками

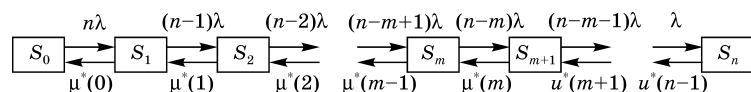


Рис. 106

Граф состояний для СМО с ошибками, зависящими от очереди

Важной является модель СМО, когда вероятность обслуживания зависит от длины очереди. Пусть в этом же примере при отсутствии очереди интенсивность потока

$$\text{обслуживания } \mu(0) = \frac{1}{t_{\text{об}}}.$$

При наличии очереди r интенсивность $\mu(r)$ обычно больше, чем $\mu(0)$ ($\mu(r) > \mu(0)$). Одновременно изменяется $p(r) > p(0)$, тогда интенсивность верного обслуживания будет

$$\mu^*(r) = \mu(r)p(r), \quad (r = 0, \dots, n-1).$$

Граф состояний будет иметь вид, показанный на рис. 106.

Получаем вероятности состояний, как для схемы гибели и размножения:

$$P_1 = \frac{n\lambda}{\mu^*(0)} P_0;$$

$$P_2 = \frac{n(n-1)\lambda^2}{\mu^*(0) \cdot \mu^*(1)} P_0;$$

...

$$P_n = \frac{n(n-1)\dots 1 \cdot \lambda^n}{\mu^*(0)\dots \mu^*(n-1)} P_0;$$

$$P_0 = \left[1 + \frac{n\lambda}{\mu^*(0)} + \frac{n(n-1)\lambda^2}{\mu^*(0) \cdot \mu^*(1)} + \dots + \frac{n(n-1)\dots 1 \cdot \lambda^n}{\mu^*(0)\dots \mu^*(n-1)} \right]^{-1}.$$

11.17. СМО С ПРОИЗВОЛЬНЫМ РАСПРЕДЕЛЕНИЕМ ВРЕМЕНИ ОБСЛУЖИВАНИЯ. ФОРМУЛА ХИНЧИНА–ПОЛЛАЧЕКА

Пуассоновский поток является важнейшей моделью для анализа большинства СМО. Роль его в теории массового обслуживания аналогична роли нормального закона распределения в теории вероятностей, так как сумма большого числа потоков даже отличных от пуассоновских дает пуассоновский поток.

Анализ СМО, когда потоки отличаются от пуассоновских, сильно усложняется.

Остановимся на одном важнейшем случае, когда анализ позволяет получить характеристики СМО при произвольном вероятностном распределении потоков в СМО.

Пусть в одноканальной СМО с бесконечным числом мест в очереди ($m = \infty$) поток заявок пуассоновский, а поток обслуживания произвольный, но известно среднее значение времени обслуживания и его постоянная дисперсия

$$\bar{t}_{\text{об}} = \frac{1}{\mu}, \quad \sigma_{t_{\text{об}}}^2 = \int_0^\infty \left(t - \frac{1}{\mu} \right)^2 \cdot g(t) dt,$$

где $g(t)$ — произвольная плотность распределения длительностей обслуживания.

Тогда для любого вида $g(t)$ средняя длина очереди определяется выражением

$$\bar{r} = \frac{\rho^2}{2(1-\rho)} \left\{ 1 + \left(\frac{\sigma_{t_{\text{об}}}}{\bar{t}_{\text{об}}} \right)^2 \right\} = \frac{\rho^2(1 + \mu^2 \sigma_{t_{\text{об}}}^2)}{2(1-\rho)} = \frac{\rho^2 + \lambda^2 \sigma_{t_{\text{об}}}^2}{2(1-\rho)}. \quad (11.27)$$

Это так называемая формула Хинчина–Поллачека.

Среднее число заявок в системе, очевидно, будет

$$\bar{K} = \bar{Z} + \bar{r};$$

$$\bar{K} = \rho + \bar{r} = \rho + \frac{\rho^2 + \lambda^2 \sigma_{t_{\text{об}}}^2}{2(1-\rho)}.$$

Для дисциплины «первый пришел — первый обслуживается» среднее время пребывания заявки в системе будет [8]

$$\bar{t}_{\text{сист}} = \bar{t}_{\text{ож}} + \bar{t}_{\text{об}} = \frac{1}{\lambda} \left[\rho + \frac{\rho^2 + \lambda^2 \sigma_{t_{\text{об}}}^2}{2(1-\rho)} \right].$$

Относительное среднее время простоя системы

$$\bar{t}_{\text{простоя}} = P_0 = 1 - \rho.$$

Формула Хинчина–Поллачека верна для любой дисциплины управления очередью, например, «первый пришел — первый ушел», «последний пришел — первый обслуживается», лишь бы выбор очередной заявки для обслуживания не зависел от времени обслуживания.

Рассмотрим два важных случая:

1. Время обслуживания постоянно (самый легкий для обслуживания случай). В этом случае

$$\begin{aligned} \sigma_{t_s} &= 0; \\ \bar{r} &= \frac{\rho^2}{2(1-\rho)}; \\ \bar{K} &= \rho + \frac{\rho^2}{2(1-\rho)}. \end{aligned} \quad (11.28)$$

2. Время обслуживания случайное, и распределение имеет экспоненциальный характер, т. е. поток обслуживания — пуассоновский (самый тяжелый для обслуживания случай). На основании формулы (11.6)

$$\sigma_{t_{\text{об}}} = \bar{t}_{\text{об}} = \frac{1}{\mu}.$$

Подставляя это выражение в формулу Хинчина–Поллачека, получаем

$$\begin{aligned} \bar{r} &= \frac{\rho^2}{2(1-\rho)}(1+1) = \frac{\rho^2}{1-\rho}; \\ \bar{K} &= \rho + \frac{\rho^2}{1-\rho} = \frac{\rho}{1-\rho}. \end{aligned} \quad (11.29)$$

Таким образом, формулы для пуассоновского потока обслуживания (см. п. 11.13) можно получить как частный случай формулы Хинчина–Поллачека.

Сравнивая выражения (11.28) и (11.29), замечаем, что средняя длина очереди для пуассоновского потока в 2 раза больше, чем для потока с постоянным временем обслуживания.

Поэтому, если мы рассчитываем на пуассоновский поток обслуживания, а на самом деле поток отличается от пуассоновского, то средняя длина очереди будет несколько завышена (максимум в 2 раза), но это страхует нас от ошибки в другую сторону.

Сказанное приводит нас к выводу, что, рассчитывая характеристики СМО в предположении пуассоновских потоков, мы получаем предельные (наихудшие) характеристики, т. е. они гарантируют нам, что в реальных СМО характеристики будут не хуже.

В этом заключается роль пуассоновской модели потоков для СМО.

Анализ более сложных моделей СМО, применение метода имитационного моделирования для анализа, а также практические применения моделей СМО смотри в [1], [8], [10].

ЛИТЕРАТУРА

1. *Таха Хэмди, А.* Введение в исследование операций. — 6-е изд. — М. : Изд. дом «Вильямс», 2001. — 912 с.
2. *Васин, А. А.* Исследование операций : учеб. пособие для вузов / А. А. Васин, П. С. Краснощеков, В. В. Морозов. — М. : Изд-во «Академия», 2008. — 464 с.
3. *Пантелеев, А. В.* Методы оптимизации в примерах и задачах : учеб. пособие / А. В. Пантелеев, Т. А. Летова. — М. : Высш. шк., 2002. — 544 с.
4. *Акулич, И. Л.* Математическое программирование в примерах и задачах : учеб. пособие. — 2-е изд. — СПб. : Изд-во «Лань», 2009. — 352 с.
5. *Кузнецов, Ю. Н.* Математическое программирование : учеб. пособие / Ю. Н. Кузнецов, В. И. Кузубов, А. Б. Волощенко. — 2-е изд. — М. : Высш. шк., 1980. — 300 с.
6. *Ляшенко, И. Н.* Линейное и нелинейное программирование : учеб. пособие / И. Н. Ляшенко, Е. А. Карагодова, Н. В. Черникова [и др.]. — Киев : Вища школа, 1975. — 370 с.
7. *Вентцель, Е. С.* Исследование операций. — М. : Советское радио, 1972. — 550 с.
8. *Черноруцкий, И. Г.* Методы оптимизации и принятия решений : учеб. пособие. — СПб. : Изд-во «Лань», 2001. — 384 с.
9. *Косоруков, О. А.* Исследование операций : учебник / О. А. Косоруков, А. В. Мищенко // под общ. ред. д. э. н., проф. Н. П. Тихомирова. — М. : Изд-во «Экзамен», 2003. — 448 с.
10. *Дегтярев, Ю. И.* Исследование операций : учеб. пособие для вузов по спец. АСУ. — М. : Высш. шк., 1986. — 320 с.
11. *Базара, М.* Нелинейное программирование. Теория и алгоритмы / М. Базара, К. Шетти. — М. : Мир, 1982. — 410 с.
12. *Морозов, В. В.* Исследование операций в задачах и упражнениях : учеб. пособие / В. В. Морозов, А. Г. Сухарев, В. В. Фёдоров. — 2-е изд., испр. — М. : Выс. шк., 2009. — 288 с.

13. *Кудрявцев, Е. М.* Исследование операций в задачах, алгоритмах и программах. — М. : Радио и связь, 1984. — 184 с.
14. *Краснощеков, П. С.* Оптимизация в автоматизированном проектировании / П. С. Краснощеков, В. В. Морозов, Н. М. Попов. — М. : Изд-во «МАКС Пресс», 2008. — 323 с.
15. Решение задач по курсу «Исследование операций». Нелинейное программирование : метод. пособие / сост. Б. А. Есипов. — Куйбышев : КуАИ, 1984. — 26 с.
16. *Есипов, Б. А.* Разработка и исследование нового эвристического алгоритма решения задачи коммивояжера : труды НИИ приборостроения / Б. А. Есипов, Е. А. Тишкин. — Самара : СГАУ, 2001. — С. 17–22.
17. Пакет прикладных программ для решения задач по исследованию операций на ЭВМ. Целочисленное программирование : метод. пособие / сост. Б. А. Есипов, А. В. Баландин, В. Г. Гашников. — Куйбышев : КуАИ, 1982. — 32 с.
18. Пакет прикладных программ для решения задач по исследованию операций на ЭВМ. Линейное частично целочисленное и нелинейное программирование : метод. пособие / сост. Б. А. Есипов, А. В. Баландин, В. Г. Гашников. — Куйбышев : КуАИ, 1982. — 32 с.
19. Математические методы и модели в экономике : метод. пособие / сост. Б. А. Есипов. — Самара : МИР, 2003. — 40 с.
20. *Аоки, М.* Введение в методы оптимизации. — М. : Наука, 1977. — 344 с.
21. *Зангвилл, У. И.* Нелинейное программирование. — М. : Сов. радио, 1973.
22. *Саати, Т.* Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. — М. : Мир, 1973. — 302 с.
23. *Фиакко, А.* Нелинейное программирование. Методы последовательной безусловной минимизации / А. Фиакко, Г. МакКормик. — М. : Мир, 1972. — 238 с.
24. Сборник задач по математике для вузов. — Ч. 4. — Методы оптимизации : учеб. пособие / под ред. А. В. Ефимова. — 2-е изд. — М. : Наука, 1990. — 304 с.
25. *Орлов, А. И.* Теория принятия решений : учебник. — М. : Изд-во «Экзамен», 2006. — 573 с.
26. *Шапкин, А. С.* Математические методы и модели исследования операций : учебник / А. С. Шапкин, Н. П. Мазаева. — 3-е изд. — М. : Изд-во «Дашков и К», 2006. — 400 с.
27. *Шикин, Е. В.* Исследование операций : учебник / Е. В. Шикин, Г. Е. Шикина. — М. : ТК «Велби»; Изд-во «Проспект», 2006. — 280 с.
28. *Волков, И. К.* Исследование операций : учеб. пособие для вузов / И. К. Волков, Е. А. Загоруйко // под ред. В. С. За-



- рубина, А. П. Крищенко. — 2-е изд. — М. : Изд-во МГТУ им. Н. Э. Баумана, 2002. — 436 с.
29. Кремер, Н. Ш. Исследование операций в экономике : учеб. пособие для вузов / Н. Ш. Кремер, Б. А. Путко, И. М. Тришин [и др.] / под ред. проф. Н. Ш. Кремера. — М. : ЮНИТИ, 2006. — 407 с.
30. Зайченко, Ю. П. Исследование операций. — Киев : Вища школа, 1986. — 390 с.
31. Васин, А. А. Теория игр и модели математической экономики / А. А. Васин, В. В. Морозов. — М. : Изд-во «МАКС Пресс», 2005. — 272 с.
32. Костевич, Л. С. Исследование операций. Теория игр : серия ВУЗ / Л. С. Костевич, А. А. Лапко. — М. : Высш. шк., 2008. — 368 с.
33. Френкин, Б. Р. Теорема Неймана о минимаксе — общеизвестная и неизвестная // Математическое просвещение. Сер. 3. Вып. 9, 2005. — С. 78–85.
34. Оуэн, Г. Теория игр : пер с англ. / под ред. А. А. Корбута. — 4-е изд. — М. : Изд-во ЛКИ, 2008. — 216 с.
35. Васин, А. А. Введение в теорию игр с приложениями в экономике : учеб. пособие / А. А. Васин, В. В. Морозов. — М., 2003. — 278 с.
36. Корнеев, В. П. Методы оптимизации : учебник. — М. : Высш. шк., 2007. — 664 с.
37. Шикин, Е. В. Математические методы и модели в управлении : учеб. пособие / Е. В. Шикин, А. Г. Чхартишвили. — М. : Дело, 2000. — 440 с.
38. Колесник, Г. В. Теория игр. — 2-е изд., испр. и доп. — М. : Книжный дом «ЛИБРОКОМ», 2010. — 152 с.
39. Корбут, А. А. Дискретное программирование / А. А. Корбут, Ю. Ю. Финкельштейн. — М., Наука, 1969. — 368 с.
40. Петросян, Л. А. Теория игр : учеб. пособие для ун-тов / Л. А. Петросян, Н. А. Зенкевич, Е. А. Семина. — М. : Высш. шк.; Книжный дом «Университет», 1998. — 304 с.
41. Катулев, А. Н. Исследование операций и обеспечение безопасности. Прикладные задачи / А. Н. Катулев, Н. А. Северцев, Г. М. Соломаха; под ред. П. С. Краснощекова. — М. : Физматлит, 2005. — 239 с.
42. Макаров, И. М. Теория выбора и принятия решений : учебное пособие / И. М. Макаров, Т. М. Виноградская, А. А. Рубчинский [и др.] — М. : Наука, 1982. — 328 с.

ОГЛАВЛЕНИЕ

Введение	5
----------------	---

Глава 1

Методология системного анализа

и исследование операций	7
1.1. Системный анализ, система, оптимизация	7
1.2. Схема операционного проекта	8
1.3. Особенности математического моделирования операций	12
1.4. Постановка задачи исследования операций в детерминированном случае, в условиях риска и неопределенности	13
1.5. Пример математического моделирования операции (задача о краске)	14

Глава 2

Линейное программирование (ЛП)	18
2.1. Общая и основная задачи ЛП	19
2.2. Геометрическая интерпретация задачи ЛП	21
2.3. Идея симплекс-метода решения задачи ЛП	24
2.4. Симплекс-таблица, стандартный алгоритм симплекс-преобразования	26
2.5. Алгоритм отыскания опорного решения задачи ЛП	29
2.6. Алгоритм отыскания оптимального решения задачи ЛП	30
2.7. Алгоритм получения первого базисного решения с использованием симплекс-процедуры (метод искусственного базиса)	33
2.8. Вырожденная задача ЛП	35
2.9. Двойственная задача ЛП	36

Глава 3

Транспортные задачи (ТЗ)	39
3.1. Математическая модель ТЗ по критерию стоимости	39
3.2. Нахождение опорного плана транспортной задачи	40
3.3. Оптимизация плана ТЗ, распределительный метод	43
3.4. Метод потенциалов решения ТЗ	46
3.5. Решение ТЗ с неправильным балансом	50
3.6. ТЗ по критерию времени	52

Глава 4

Дискретное программирование	56
4.1. Особенности задач дискретного программирования	58
4.2. Примеры моделей задач дискретного программирования	59
4.3. Алгоритм решения задачи о ранце	70



4.4. Решение задач ЛЦП методом отсечений Гомори	74
4.5. Метод ветвей и границ (МВГ)	79
4.6. МВГ для решения задачи коммивояжера	81
4.7. Алгоритм МВГ для задачи ЛЦП	85
4.8. Алгоритмы решения задач булева программирования ...	89

Глава 5

Динамическое программирование (ДП)	96
5.1. Принцип оптимальности Р. Беллмана	98
5.2. Решение графовых задач на основе принципа Беллмана	100
5.3. Функциональное уравнение Беллмана	101
5.4. Задачи распределения ресурсов	103
5.5. Расширение модели задач динамического программирования	109
5.6. Пример решения задачи распределения ресурсов	111
5.7. Эффективность динамического программирования ...	113

Глава 6

Нелинейное программирование	115
6.1. Особенности задач нелинейного программирования ...	115
6.2. Прямые методы одномерной оптимизации нелинейных функций без ограничений	118
6.3. Градиентные методы многомерной оптимизации	120
6.4. Метод деформируемого многогранника Нелдера–Мида	123
6.5. Задача НЛП с ограничениями-равенствами	126
6.6. Выпуклое НЛП	129
6.7. Теорема Куна–Таккера для выпуклого НЛП	130
6.8. Квадратичное программирование	132
6.9. Методы возможных направлений	137
6.10. Метод проекции градиента	144
6.11. Методы штрафных и барьерных функций	147
6.12. Метод скользящего допуска	153

Глава 7

Особенности современной теории принятия оптимальных решений	156
7.1. Общая постановка задачи принятия решения	158
7.2. Классификация задач принятия решений	161
7.3. Многокритериальная оптимизация	163
7.4. Определение множества Парето	166
7.5. Методы условной многокритериальной оптимизации ..	168

Глава 8

Игровые модели принятия решений	173
8.1. Основные понятия теории игр	174
8.2. Игры двух лиц с нулевой суммой	179
8.3. Игры двух лиц с произвольной суммой	194

ОГЛАВЛЕНИЕ

299

8.4. Игры многих лиц	205
8.5. Многошаговые игры	221

Глава 9

Элементы теории статистических оптимальных решений	231
9.1. Принятие решений при известных априорных вероятностях	233
9.2. Методы принятия решений в условиях априорной неопределенности	235
9.3. Планирование эксперимента при принятии решений ...	236
9.4. Многоэтапное принятие решений	238

Глава 10

Экспертные процедуры для принятия решений	242
10.1. Общая схема экспертизы	242
10.2. Задача оценивания	243
10.3. Подготовка экспертизы	244
10.4. Методы обработки экспертной информации	245
10.5. Метод Делфи для численной оценки	247
10.6. Строгое ранжирование	249
10.7. Нестрогое ранжирование	250
10.8. Метод попарных сравнений	251

Глава 11

Операции массового обслуживания	252
11.1. Модель систем массового обслуживания	252
11.2. Марковский случайный процесс с дискретными состояниями	253
11.3. Марковский процесс с дискретным состоянием и непрерывным временем. Система уравнений Колмогорова	256
11.4. Поток событий. Простейший поток и его свойства ...	260
11.5. Предельные вероятности состояний	264
11.6. Процесс гибели и размножения	266
11.7. Циклический процесс	267
11.8. Классификация СМО	269
11.9. Одноканальная СМО с отказами	270
11.10. Многоканальная СМО с отказами	273
11.11. Многоканальная СМО с ожиданием	275
11.12. Одноканальная СМО с ожиданием	279
11.13. СМО с ожиданием и бесконечной очередью	281
11.14. Замкнутые системы массового обслуживания	282
11.15. СМО с взаимопомощью между каналами	285
11.16. СМО с ошибками	289
11.17. СМО с произвольным распределением времени обслуживания. Формула Хинчина–Поллачека	291
Литература	294

Борис Алексеевич ЕСИПОВ

**МЕТОДЫ ИССЛЕДОВАНИЯ
ОПЕРАЦИЙ**

Учебное пособие

Художественный редактор *С. Ю. Малахов*
Редактор *Р. И. Исаева*
Корректоры *В. В. Вересиянова, А. М. Плетнева*
Подготовка иллюстраций *Н. А. Платонова*
Выпускающие *Е. А. Антипова, О. В. Шилкова*

ЛР № 065466 от 21.10.97
Гигиенический сертификат 78.01.07.953.П.004173.04.07
от 26.04.2007 г., выдан ЦГСЭН в СПб

Издательство «ЛАНЬ»
lan@lpbl.spb.ru; www.lanbook.com
192029, Санкт-Петербург, Общественный пер., 5.
Тел./факс: (812)412-29-35, 412-05-97, 412-92-72.
Бесплатный звонок по России: 8-800-700-40-71

Подписано в печать 20.02.13.
Бумага офсетная. Гарнитура Школьная. Формат 84×108^{1/32}.
Печать офсетная. Усл. п. л. 13,44. Тираж 1500 экз.

Заказ № .

Отпечатано в полном соответствии
с качеством предоставленных диапозитивов
в ОАО «Издательско-полиграфическое предприятие «Правда Севера».
163002, г. Архангельск, пр. Новгородский, д. 32.
Тел./факс (8182) 64-14-54; www.ippps.ru