Федеральная служба воздушного транспорта России
Академия гражданской авиации

Ю. Е. Хорошавцев

ОСНОВЫ АСУ ТРАНСПОРТНЫМИ СИСТЕМАМИ

Учебное пособие

Санкт-Петербург
1999

ФЕДЕРАЛЬНАЯ АВИАЦИОННАЯ СЛУЖБА РОССИИ

АКАДЕМИЯ ГРАЖДАНСКОЙ АВИАЦИИ

Ю.Е. ХОРОМАВЦЕВ

ОСНОВЫ [illegible] ТРАНСПОРТНЫМИ СИСТЕМАМИ

Учебное пособие

Санкт-Петербург

1999

Утверждено в качестве учебного пособия для студентов специальности "Организация и управление на воздушном транспорте"

Х 87(03)

Хорошавцев Ю.Е. Основы АСУ транспортными системами: Учебное пособие/ Академия ГА, С.-Петербург, 1993.

Содержит начальные сведения о математических методах, используемых при разработке автоматизированных систем управления (АСУ).

Рассмотрены основные принципы построения АСУ. Приводятся примеры современных систем управления базами данных, даются сжатые представления об экспертных системах.

Основное внимание уделяется математическим методам оптимального управления. Достаточно подробно излагаются методы теории массового обслуживания.

Изложение материала ориентировано на привитие навыков решения практических задач, связанных с эксплуатацией транспортных систем.

Предназначено для студентов факультета эксплуатации транспорта всех специализаций.

Ил.26, табл.20, библ.7 назв.

I. ВВЕДЕНИЕ В АСУ

I.I. Методологическая основа и задачи, стоящие перед АСУ

Практическая необходимость решения все более сложных задач, связанных с автоматизацией управления производством (транспортными системами, выпуском продукции, организацией обслуживания и т.д.) с целью получения максимальной эффективности, а также возможности, обусловленные развитием вычислительной техники, привели к разработке инженерных методов обработки информации, которые и лежат в основе автоматизированных систем управления (АСУ).

От АСУ требуется получение управленческих решений, которые она вырабатывает (генерирует) на основе входных данных (в простейших случаях она их только сортирует). Под решением понимается какой-то выбор из ряда возможностей, имеющихся в данной задаче. Например, направить так или иначе потоки грузов, распределить имеющиеся средства и т.д. Обычно ищут оптимальные решения, т.е. наилучшие в смысле принятых критериев; причем надо помнить, что сами критерии выбираются субъективно в зависимости от тех или иных требований к решению (подешевле, понадежнее и т.д.). Приведем несколько примеров.

I. План снабжения.

Имеются сырьевые базы и предприятия, потребляющие сырье. Они связаны транспортными коммуникациями со своими тарифами. Требуется найти такой план снабжения, чтобы он максимизировал прибыль.

2. Постройка участка магистрали.

Требуется спланировать очередность работ, распределение ресурсов (машин, людей) так, чтобы сроки строительства были минимальными.

3. Выпуск продукции.

Требуется найти план выпуска различных видов продукции с максимальной прибылью.

Приведенные примеры имеют общие черты. Заданы условия, характеризующие обстановку. В рамках этих условий требуется принять решение, т.е. осуществить выбор, чтобы задуманное мероприятие было оптимальным в смысле принятого заранее критерия. Иными словами, математическая формулировка задачи должна содержать описание условий выбора, а также критерия оптимальности выбора.

С ростом масштабов и сложности планируемых действий математические методы при разработке АСУ приобретают все большее значение. Работа небольшого аэродрома вполне может быть обеспечена силами одних диспетчеров; работа крупного аэроузла требует уже автоматизированной системы управления, работающей согласно четкому алгоритму. Для выработки такого алгоритма необходимо применение современных математических методов.

А С У – это комплекс аппаратных и программных средств, предназначенных для выработки управленческих решений.

Методологической основой для принятия решений в современных АСУ является комплекс научных методов, объединенных под названием "исследование операций". Иногда говорят о математическом программировании, имея в виду планирование.

Для того, чтобы спроектировать АСУ, надо ответить на две группы вопросов. Первая группа – как собрать и представить необходимую информацию, содержащуюся в различных источниках. Вторая – какие методы обработки информации использовать, чтобы на основе собранных данных получить решение, желательно оптимальное.

Задачи, стоящие при разработке АСУ, следующие:

1. Организация сбора и обработки информации, относящейся к

2. Построение адекватной математической модели, описывающей управляемый процесс. Желательно, но не обязательно, чтобы модель позволяла получить оптимальное решение в смысле нахождения варианта действий, приносящих максимальный выигрыш от операции.

Модель должна содержать параметры (a), характеризующие сам процесс или систему (производительность, тарифы перевозок, нормативы и т.п.), а также искомые неизвестные – элементы решения (x) (площади, занимаемые под складирование; объемы перевозок из i-го пункта в j-й и т.д.).

3. Определение значения параметров модели. Очень часто они не являются фиксированными величинами, а подвержены влиянию случайных факторов. В таких случаях обычно находятся их среднестатистические оценки.

4. Разработка методов оптимизации решений. Каждый метод ориентирован на свои математические модели.

5. Расчет эксплуатационных характеристик транспортных или обслуживающих систем (пропускной способности, времени ожидания и т.д.). Оптимизация этих характеристик, т.е. решение так называемых обратных задач, ввиду сложности в данном курсе не рассматривается.

1.2. Классификация АСУ. Принципы построения. Структура. Аппаратные средства

По своему назначению автоматизированные системы делятся на АСУ:

- технологическими процессами;
- планированием производства (включая сбыт и снабжение);
- планированием и управлением эксплуатацией коммуникационных средств (транспорта, связи);

- административно-хозяйственной деятельностью;

- автоматизированными рабочими местами (АРМ), служащими для автоматизации инженерной деятельности.

Каждая АСУ предназначена для решения определенных (одной или нескольких) задач. В соответствии с этим для нее определена своя номенклатура сведений, т.е. исходной информации, необходимой для получения решения. Решение производится в соответствии с заложенными алгоритмами и программой, построенными на основе математических методов (желательно, но не обязательно оптимизационных).

В общем случае АСУ включает в себя:

- систему управления базами данных (СУБД) различной степени развитости;

- вычислитель;

- устройства ввода-вывода (интерфейс пользователя).

Исходная информация для получения решения в АСУ объединяется в базы данных (БД). СУБД позволяет собирать, хранить и представлять по задаваемым признакам информацию. СУБД строится на базе компьютеров. Обычно СУБД включает в себя набор файлов данных, имеющих стандартный формат, и управляющую программу. Одним их распространенных сейчас является формат dBASE в системе MS DOS. В системе WINDOWS широкое распространение получила система ACCESS.

Для решения детерминированных задач с жестко определенным набором данных СУБД в АСУ может вырождаться просто в массивы исходных данных.

Вычислитель в соответствии с записанной программой и введенными исходными данными решает задачу.

Устройства ввода-вывода выдают решение в удобном для пользователя виде: распечатку таблиц, графиков, диаграмм, изображение маршрутов, формирование расписаний и т.д.

Аппаратные средства АСУ могут быть: стандартными и специальными.

К стандартным средствам относятся ЦВМ различных классов, от больших до ПЭВМ. Однако большие ЦВМ обычно решают сложные технологические задачи (управление электростанцией, гидроузлом, портом и т.д.). Они имеют специфические средства сопряжения с другими устройствами (измерителями, сигнализаторами и т.д.), системы связи с операторами (диспетчерские пульты, АРМ и т.д.) и обычно входят в специальные АСУ.

Для того, чтобы организовать потоки информации для работы АСУ, компьютеры объединяют в сети. Линии связи и специальные устройства (шлюзы, серверы, маршрутизаторы и т.д.) также дополняют аппаратные средства АСУ.

2. ИНФОРМАЦИОННАЯ БАЗА АСУ

2.1. Базы данных. Системы управления базами данных СУБД

Под информационной базой обычно понимают упорядоченный набор исходных данных, необходимых для информационного обеспечения при решении практических задач. Современные АСУ, оперирующие большими массивами исходных данных, содержащихся в различных источниках, при обращении к ним придерживаются определенных правил поведения. Для того, чтобы собирать, сортировать, сопоставлять, обрабатывать однотипную информацию, ее необходимо представить в виде баз данных.

База данных - совокупность сведений, объединенных и классифицированных по определенным признакам, позволяющим представлять информацию на компьютере в табличном виде. Такое представление возможно тогда, когда данные структурированы, т.е. разбиты по группам с однотипным содержанием (например, тарифы перевозки, габариты груза, пункты назначения и т.п.). Каждая запись в базе данных (БД) состоит из последовательности заполняемых колонок (полей). БД иногда называют электронными таблицами, поскольку их отображение на дисплее (или распечатка) имеет обычный табличный вид.

Структура БД отображается в файлах баз данных. Задаются число полей, их имена, размер (в байтах), тип, которые регламентируются системой управления базами данных (СУБД).

СУБД - управляющие программы, позволяющие оперировать с базами данных. Наиболее основными функциями СУБД являются:

- создание файлов БД;
- запись информации;
- просмотр и редактирование;
- поиск и фильтрация данных;
- формирование отчетных форм по задаваемым шаблонам.

Существует много СУБД, служащих для информационного обеспечения различного рода деятельности (экономических задач, конструкторских разработок и т.д.). В качестве примера можно привести системы PARADOX, FoxPro, Clipper. В системе WINDOWS распространенной является СУБД ACCESS.

Ранее перечисленные СУБД являются реляционными, которые построены по принципу установления связей между отдельными таблицами (файлами) и обработки их как будто они являются одним объектом. Такое постоянно случается, когда требуемая информация содержится в различных источниках – файлах БД. При этом, как правило, одной записи одного файла соответствует несколько записей другого. Например, в файле A указаны авиакомпании с их реквизитами, а в файле B – типы располагаемых для перевозок самолетов (табл. 2.I).

Таблица 2.I

Файл A

№	Лицензия	Авиакомпания	
I	. T .	Пулково	. .
2	. T .	Аэрофлот	. .
3	. F .	Штопор	. .
.	...	...	. .

Файл B

№	Тип ВС	Авиакомпания	
I	Ту-204	Пулково	. .
2	А-3I0	Байкал	. .
3	Ил-96	Пулково	. .
.	...	...	. .

В первом поле файла A символами . T . и . F . обозначены используемые логические утверждения (есть лицензия или ее нет). В примере, при объединении информации, относящейся к а/п Пулково, запись I файла A связывается с записями I и 3 файла B.

Сразу отметим, что определить сколько нужно файлов (таблиц) и какой структуры, чтобы упаковать всю информацию, – задача самостоятельная и нетривиальная. Для ее решения предложены различные методы [6], в основе которых лежит классификация, группирование информации с целью (например, минимизации объема занимаемой памяти компьютера).

Установление связи может быть типа:

- одна запись одного файла с одной записью другого;
- одна запись - с несколькими записями другого.

С помощью связей можно компоновать информацию из нескольких файлов (таблиц) в одной отчетной форме. Для установления связей необходимо выполнить условие связи: по какому признаку связываются файлы. В примере табл. 2.1 таким признаком является значение поля "Авиакомпания". Для связи типа одна связь с несколькими иногда требуется выполнение индексирования файлов. Индексирование заключается в создании индексных файлов (по крайней мере, одного), выполняющих служебную, вспомогательную роль. Индексный файл содержит информацию (индексы), упорядочивающую записи в файле БД (в частности, в порядке возрастания кодов стандарта ASCII). При этом физическая последовательность записей не нарушается. Индексирование значительно убыстряет работу СУБД. Оно может производиться как в порядке возрастания, так и убывания кодов.

Помимо связи файлов, индексирование играет важную роль при поиске, сортировке и выборке информации. Индексированные файлы обрабатываются быстрее.

В СУБД ACCESS структура файлов БД другая: в одном файле может размещаться несколько таблиц. Но связи между ними устанавливаются также по ключу (признаку условия связи).

2.2. Программные средства СУБД

Программные средства ориентированы на выполнение пользовательских функций при работе с базами данных. Эти средства учитывают структуру БД и приспособлены лишь к стандартным форматам. Одним из наиболее распространенных является формат dBASE.

В формате dBASE структура файла БД описывается набором поименованных полей (колонок таблицы) следующих типов:

- даты;
- числового;

- символьного;
- логического;
- примечаний (МЕМО – поле).

Тип поля задает характер записываемой в него информации. Несоответствие типа данных с типом поля приводит к ошибке записи. Кроме типа задается длина поля и точность представления вещественного числа (для числового поля). Таким образом, данные ограничены размером поля (например, символьные – до 254 байт).

Для создания файла БД используется команда CREATE с атрибутами.

При работе СУБД существующие (созданные) файлы необходимо "открыть". Открывают их по команде USE с атрибутами.

Формат dBASE поддерживает реляционные базы данных и позволяет устанавливать связи более чем с десятью одновременно открытыми файлами. Открываются файлы в так называемых "рабочих областях" по команде SELECT . Установка связей между файлами осуществляется командой SET RELATION с атрибутами. Заметим, что число одновременно открытых файлов может быть ограничено установкой в системном файле CONFIG.SYS.

В формате dBASE файлы баз данных имеют стандартный тип: данных (.dbf), примечаний (.dbf), индексов (.ntx) и т.д.

Файл данных имеет форму таблицы, каждая строка которой – одна запись файла, а каждый столбец – набор одноименных однотипных полей записей. Одно данное (элемент таблицы) находится на пересечении строки и столбца.

СУБД обрабатывает информацию путем выполнения определенных команд и функций (описание их приводится в специальных руководствах). Эти команды объединяются в программный модуль, дополняемый библиотеками стандартных процедур.

Формат dBASE позволяет использовать файлы данных в вычислительных сетях. При работе в сети имеются специальные команды совместимости, препятствующие одновременной модификации данных несколькими пользователями, блокировки, запрета доступа, запрета на редактирование и т.д. Обычно функции распорядителя (изменения, удаления, добавления информации) отводятся "администратору" БД. Пассивные пользователи могут только "потреблять" информацию, но никак ее не модифицировать. Вычислительные сети наиболее эффективны там, где источники информации находятся в различных подразделениях, службах авиакомпании. Каждая служба пользуется своими таблицами БД: объединение, обмен информацией производится с помощью сети.

Примером одной из старейших АСУ, работающих в формате dBASE, служит известная АСУ "Безопасность".

С точки зрения пользователя, принципы построения СУБД в операционной системе WINDOWS похожи на то, как это сделано в MS DOS. Рассмотрим для примера некоторые особенности работы с СУБД ACCESS, ни в коей мере не желая дублировать специальные руководства.

При работе с ACCESS пользователь обращается к комплексной многофункциональной системе, в которой он создает свои файлы данных и в которой он с ними работает (обеспечивая доступ к системе WINDOWS). Создать самостоятельную программу в виде исполняемого файла с тем, чтобы с ней работать, затруднительно. Занимая много места в памяти, при работе с несложными БД, ACCESS может оказаться "пушкой при стрельбе по воробьям". Тем не менее это мощная система, возможности которой постоянно совершенствуются в новых версиях и в которой отражены основные черты СУБД в WINDOWS.

ACCESS формирует таблицы БД, описания их форм представления и компонует в единый файл. Как в целом в ОС WINDOWS основным средством функционального взаимодействия с компьютером является мышь. С ее помощью выбираются и задаются режимы работы ACCESS и выполняемые функции. В данной СУБД не надо знать языка программирования, чтобы иметь возможность работать с базами данных. Пользователь непосредственно с помощью меню выполняет все нужные действия. Например, создает, конструирует (модифицирует), открывает таблицы: формирует "запросы" (объединенные таблицы). При установлении связей между таблицами, таблицы должны иметь одноименные поля, как и в dBASE. *Процесс объединения выполнен наглядно с помощью* графического построения. С "запросом" можно работать как с новой таблицей (например, установить сортировку и т.д.).

Для работы с конкретной СУБД пользователь сам подберет для себя специальную литературу, давать же обобщающее подробное ее описание - безнадежное и бесполезное занятие. Автор прекрасно понимает, что на рынке компьютеров и программного обеспечения все меняется сказочно быстро. Поэтому можно лишь посоветовать читателю постоянно следить за развитием компьютерной техники.

3. ИСПОЛЬЗОВАНИЕ ЭКСПЕРТНЫХ СИСТЕМ В АСУ

3.I. Структура типовой экспертной системы. Функции блоков

Экспертные системы (ЭС) относят к системам искусственного интеллекта. А именно к системам, располагающим собственными знаниями, так называемыми базами знаний. База знаний - это совокупность правил, математических (логических) процедур, позволяющих решать задачи в данной проблемной области (т.е. определенного типа). Отличительной особенностью ЭС от обычных вычислительных программ, реализующих те или иные методы решения задач, является то, что база знаний может развиваться. В уже созданной ЭС, без участия разработчика, база знаний может видоизменяться, пополняться новыми знаниями (т.е. правилами и т.д.), получаемыми от эксперта. Эксперт - специалист по решению конкретного вида задач (например, специалист по маркетингу) может совершенно не знать программирования, но тем не менее, в диалоге с вычислительной машиной (ЭС) передает ей свои знания. Для облегчения диалога и во избежание ошибок при этом обычно присутствует инженер по знаниям - специалист в области программирования, которому, впрочем, необязательно знать устройство ЭС. Процедура пополнения знаний происходит в виде вопросов со стороны ЭС и ответов эксперта. Естественно, что задание вопросов запрограммировано в ЭС в так называемой "оболочке". Пока ЭС не наполнена знаниями, она представляет собой оболочку.

Решения ЭС могут быть объяснены самой системой на качественном уровне. По запросу пользователя ЭС может дать комментарий на конечное или промежуточное решение. Естественно, комментарии заносятся в базу знаний при диалоге ЭС с экспертом (а не сама ЭС их выдумывает).

Структура ЭС включает в себя (рис. 3.I):

- интерпретатор;
- базу данных;
- базу знаний;
- блок ввода-вывода.

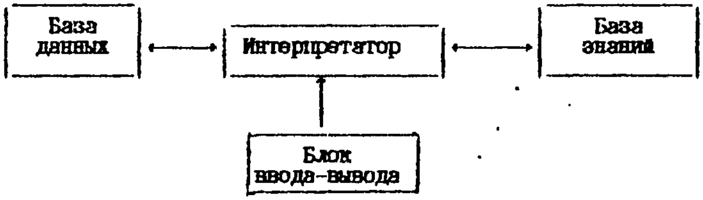


Рис. 3.1

Интерпретатор служит для преобразования данных, полученных от эксперта в совокупность правил (решающих правил), пополняющих базу знаний ЭС.

База данных содержит всю необходимую для решения задачи исходную информацию.

База знаний содержит в формализованном виде сведения о том, как решить задачу (правила, алгоритмы, процедуры и т.д.).

Следует иметь в виду, что изначально алгоритм решения задачи может быть неизвестен, а исходные данные могут носить сугубо субъективный характер ("скорее холодный, чем горячий"). Для учета оценочного характера данных может использоваться алгебра нечетких (размытых) множеств. (Порой эту алгебру считают разновидностью теории вероятности, в которой присутствуют субъективные вероятности в отличие от частотно задаваемых; в этой алгебре понятие вероятности заменяется функцией принадлежности).

Блок вывода служит для представления решения пользователю. Иногда желательно, чтобы решение оформлялось грамматически, т.е. выглядело в виде законченных предложения на естественном языке (например, русском). Это сложная самостоятельная задача, она решается различными методами. В частности, распространен метод шаблонов: шаблонная фраза содержит пробелы, заполняемые ЭС при решении задачи. Таким образом, ЭС находит лишь нужные слова и подставляет их в заготовленные места. Обычно формирование более сложных фраз возлагается на лингвистический процессор. Одним из методов построения сложных выражений является метод предикатов (высказываний). Примером фразы, построенной из шаблона, может служить следующая: "Поставку груза в $\{$Певек; Магадан; ...$\}$ производить $\{$поездом; самолетом; пароходом; ...$\}$ с периодичность $\{$месяц; квартал; полугодие; ...$\}$". Выражения в фигурных скобках определяют возможные значения, выбираемые экспертной системой при решении задачи.

3.2. Режимы работы экспертной системы. Наполнение знаний и решение задач

Выделяют 2 основных режима работы ЭС:

- наполнение (приобретение) знаний;
- решение задач.

Приобретение знаний - это как бы обучение системы. Для приобретения знаний в общении с ЭС участвует эксперт, который наполняет (уже существующую, готовую) ЭС знаниями, позволяющими ей в режиме решения самостоятельно находить ответы на поставленные вопросы (давать рекомендации). Обычно предполагается, что имеются квалифицированные специалисты, обладающие требуемой эрудицией. Однако при наполнении знаниями системы в областях, находящихся на стыке профессий, режим приобретения знаний затрудняется: нужно "увязывать" разнородные знания. Такая стыковка возможна опять-таки с привлечением некоторых экспертов-эрудитов, а также возможна формальными методами (автоматическая). Одним из них является известная задача синтеза знаний по координатным (т.е. здесь по частным) функциям выбора (правилам, составляющим знания). Синтез может осуществляться, в частности, с использованием принципа транзитивности отношения в правилах (если $a > b$ и $b > c$, то $a > c$).

Режим приобретения знаний может осуществляться путем диалога (интерактивного взаимодействия) эксперта с компьютером на языке, приближенном к естественному. Чередование вопросов и ответов является внешней формой процедуры приобретения знаний.

Приобретение знаний проходит несколько этапов: идентификации, концептуализации, формализации, выполнения, тестирования. В них последовательно от общих представлений о решаемом классе проблем до конкретных правил, от предположительного вида решений до описания всех конкретных вариантов решения формируется база знаний. "Наскоком" знания не приобретаются. Непродуманное наполнение знаниями ЭС приводит к невозможности получения либо разумного, либо вообще какого-либо решения.

Наполнение знаниями возможно также путем прямого составления программных блоков, содержащих правила, на внутреннем языке программирования данной ЭС.

Наиболее простые знания (и наиболее часто) задаются в виде логических правил вида

$$\text{IF } a \text{ THEN } b \text{ ELSE } c\,, \qquad (3.1)$$

т.е., если выполнено условие a, то производится действие b, в противном случае – c .

Последовательное применение логических правил в итоге приводит к единственному следствию (если нет ошибок). Например, пусть для белки эксперт придумал единственное правило, как забраться на вершину. А именно, если встречается растущая по пути ветка – надо выбирать самую толстую. Многократное применение этого правила приведет к вершине. Другой эксперт может дополнить базу знаний: например, вводит еще одно логическое условие – ответвление должно расти вверх. Экспертная система получила эти правила (или сначала одно) и сама сложила из них алгоритм и построила рабочую программу.

На практике число возможных значений a, b, c велико, а сами они могут иметь различный тип, например, числовой, символьный, логический, и описывать различные сущности. Поэтому используются различные модели представления знаний, позволяющие навести порядок во множествах переменных задачи. Часто знания описывают в виде семантической сети – фрейма. С помощью фрейма знания структурируются.

В режиме решения задач в общении с ЭС участвует пользователь – обычно не специалист в области вычислительной техники и программирования.

В зависимости от того, ориентирована ЭС на решение узкого круга задач или она универсальна, меняется форма взаимодействия с ней пользователя. Универсальные системы, как правило, представляются в виде "оболочек", т.е. программной среды, позволяющей приобретать знания практически в любой области, а затем в режиме решения давать ответы на запросы пользователя. Такие системы строятся на простых правилах логического типа и "всеядны", но "глубокие размышления" им не под силу. Вместе с тем, универсальные ЭС экономически наиболее оправданы.

В режиме решения данные о задаче обрабатываются сначала лингвистическим процессором. Он преобразует входные данные, представленные на естественном языке, в команды, функции и переменные задачи на внутреннем языке системы и, наоборот, делает понятными пользователю сообщения системы. Интерпретатор на основе входных данных, продукционных правил, извлекаемых из базы знаний, дает решение задачи. Если решение непонятно пользователю, то система может объяснить, как ответ был получен. Обычно объяснительный блок сообщает следующее: какие правила применялись, как в них использовалась информация, какие промежуточные выводы получались.

Архитектура экспертных систем различается в первую очередь по способу представления знаний и методами работы интерпретатора.

4. ПРИНЯТИЕ РЕШЕНИЙ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

4.1. Постановка задачи принятия решений в условиях неопределенности

На АСУ возлагаются задачи выработки управленческих решений (рекомендаций) пользователю для последующих действий. Эти рекомендации могут вытекать либо из последовательного перебора решающих правил в виде логического следствия, как это реализуется в экспертных системах, либо определяться методами оптимального управления.

Оптимальное управление всегда понимается в смысле принятого критерия (показателя эффективности) W. Сам критерий выбирается субъективно исходя из тех требований, какие предъявляются к искомому решению. Решение оптимальное по одному критерию обычно неоптимально по другому.

В детерминированном случае W зависит от двух групп величин: заданных условий задачи α и элементов решения x. Реальные задачи подвержены влиянию случайных факторов ξ, которые вносят элемент неопределенности в решение задачи. Условно для критерия эффективности можно записать, что

$$W = W(\alpha, x, \xi). \qquad (4.1)$$

При поиске оптимальных решений нужно научиться учитывать случайный характер ξ. Ведь расчет, произведенный при одних исходных значениях, даст результат, отличный от того, который получится при других, обусловленных влиянием фактора случайности. Наличие неопределенности факторов ξ ставит задачу о выборе решения в условиях неопределенности (например, оптимального планирования работы авиакомпании при неизвестных еще тарифах на обслуживание: химработы, патрулирование и т.д).

В том случае, когда ξ представляет собой случайные величины, определенные в теории вероятностей, то мы имеем стохастические задачи. Величины являются случайными, когда случайны их конкретные значения, но неслучайны вероятностные характеристики.

Одним из наиболее распространенных методов их решения является использование статистических оценок случайных величин, замена их средними значениями. Случайный фактор ξ, вызывающий флуктуации α, исключается при замене случайных величин их средними значениями (математическими ожиданиями в теории вероятностей). Таким образом, задача (4.I) приводится к виду

$$W = W(\bar{\alpha}, X),$$

где $\bar{\alpha}$ - средние значения параметров задачи.

Другой возможностью решения стохастической задачи является выбор в качестве показателя эффективности среднего значения самого W:

$$\bar{W} = M\left[W(\alpha, X, \xi)\right], \qquad (4.2)$$

где M - знак математического ожидания.

Напомним, что если ξ - случайная величина с плотностью распределения вероятности $f(\xi)$, то ее математическое ожидание равно

$$M[\xi] = \bar{\xi} = \int_{-\infty}^{\infty} \xi f(\xi)\, d\xi. \qquad (4.3)$$

Вдумчивый читатель может спросить: а не все ли равно, сначала найти средние значения входных переменных и для них вычислить W или сначала вычислить W, а затем осреднить? Действительно, в линейных задачах это безразлично, но в общем случае разница есть. Пренебрегая математической строгостью доказательства поясним это на примере переменного электрического тока - его среднее значение равно нулю, однако, мощность далеко не нулевая.

4.2. Метод максимального правдоподобия

Метод максимального правдоподобия (МП) служит для нахождения оценок параметров распределения (выборочного среднего, дисперсии и т.д.) случайных величин. Ес и из предварительного анализа известен вид функции плотности вероятности, но неизвестны параметры этой функции, т.е. как они вычисляются по данным наблюдений, метод максимального правдоподобия оказывается очень эффективным.

Пусть плотность вероятности с точностью до неизвестных параметров задана функцией $f(x;\ \theta_1,\ \theta_2,\ \ldots,\ \theta_k)$. В качестве оценок неизвестных параметров $\theta_1,\ \theta_2, \ldots,\ \theta_k$ по данному методу принимают такие значения $\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_k$, которые доставляют максимум так называемой функции правдоподобия. Можно сказать, что ищутся формулы для вычисления оценок $\tilde{\theta}_1,\ \tilde{\theta}_2, \ldots \tilde{\theta}_k$ исходя из условия, что функция плотности вероятности с определенными таким образом параметрами наилучшим образом согласуется с результатами наблюдений.

Пусть имеется выборка случайной величины X: $x_1,\ x_2, \ldots x_n$. Найдем вероятность того, что элементы выборки примут полученные значения, т.е.

$$P\left[X_1 = x_1,\ X_2 = x_2, \ldots,\ X_n = x_n\right] = f(x_1;\theta_1,\ldots,\theta_k)\cdot f(x_2;\theta_1,\ldots,\theta_k)\times$$

$$\times\ f(x_n;\theta_1,\ldots,\theta_k)(dx)^n .$$

Функция правдоподобия отсюда определяется так:

$$L(\theta_1\theta_2,\ldots,\theta_k) = f(x_1;\theta_1,\ldots,\theta_k)f(x_2;\theta_1,\ldots,\theta_k)\times\ldots$$
$$\ldots\times f(x_n;\theta_1,\ldots,\theta_k). \tag{4.4}$$

Плотность вероятности обычно выбирается исходя из общих физических соображений (и желания чего-нибудь попроще), но есть и строгие (плюс трудоемкие) методы ее определения, излагаемые в теории вероятности.

Из необходимых условий максимума функции $L(\theta_1,\theta_2,\ldots\theta_k)$ находим, что оценки $\tilde{\theta}_1,\ \tilde{\theta}_2, \ldots,\ \tilde{\theta}_k$ являются решениями следующей системы k уравнений:

$$\frac{\partial L\ (\theta_1\ \theta_2,\ldots,\ \theta_k)}{\partial\ \theta_j} = 0\ , \qquad j = 1,\ 2,\ldots k.$$

Подобные задачи возникают, когда выбран вид функции, описывающей закон распределения вероятностей, но в этой функции надо определить некоторые из ее параметров (коэффициенты $\theta_1, \ldots, \theta_k$) таким образом, чтобы функция наиболее точно соответствовала результатам наблюдений.

Для упрощения вычислений иногда удобно рассматривать логарифм функции правдоподобия

$$\ln L(\theta_1 \ \theta_2, \ldots, \theta_k),$$

ведь логарифмируя функцию, мы не меняем положения ее максимума.

Для примера найдем оценку дисперсии нормально распределенной генеральной совокупности со средним m. Из теории вероятностей мы помним, что она равна

$$\tilde{\delta}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - m)^2.$$

Теперь подтвердим это по методу МП. Как известно, плотность нормального распределения

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}.$$

Функция правдоподобия для выборки объема n из нормально распределенной генеральной совокупности определяется по формуле (4.4):

$$L(\theta_2) = L(\sigma^2) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x_i - m)^2}{2\sigma^2}} =$$

$$= \frac{1}{(2\pi)^{n/2}(\sigma^2)^{n/2}} e^{-\frac{\sum_{i=1}^{n}(x_i - m)^2}{2\sigma^2}}.$$

Отбрасывание константы $1/(2\pi)^{n/2}$ и логарифмирование $L(\sigma^2)$ не изменит значения $\tilde{\sigma}^2$, доставляющего максимум этой функции: Следовательно МП - оценку для σ^2 можно определить как значение, доставляющее максимум функции:

$$\ln L(\sigma^2) = -\frac{n}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - m)^2 .$$

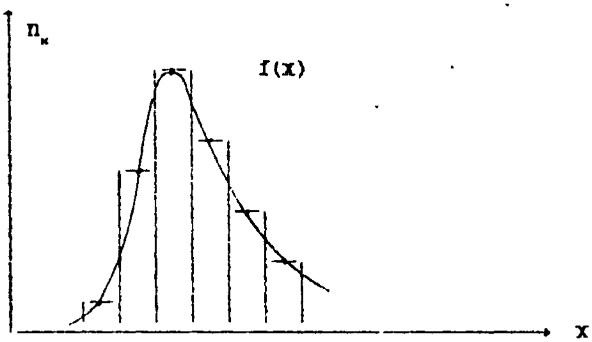
Дифференцируя по σ^2 и приравнивая нулю из условия максимума (вообще-то экстремума) полученное выражение, приходим к уравнению

$$\frac{d\ln L}{d(\sigma^2)} = -\frac{n}{2}\cdot\frac{1}{\sigma^2} + \frac{\Sigma(x_i - m)^2}{2(\sigma^2)^2} = 0.$$

Отсюда следует, что $\tilde{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - m)^2$.

Часто бывает, что при решении практических задач по виду функции плотности распределения сразу известны формулы для вычисления ее параметров. Так для нормального распределения формулы для m и σ всем знакомы. Но бывает, что после аппроксимации плотности распределения некоторой функцией, не ясно как определяются ее параметры по значениям выборки наблюдений. Тут становится полезным метод максимального правдоподобия.

На практике возникают задачи, подобные следующей. Пусть загрузка по рейсам транспортного самолета описывается статистическим рядом и гистограммой (рис. 4.1).



$n'_к$ – относительные частоты наблюдений

Рис. 4.I

Аппроксимация ряда функцией распределения $f(x)$ дает несимметричную кривую (недогрузка редка), для описания которой нормальное распределение уже не подходит. Требуется более сложная формула, включающая параметр, отвечающий за несимметричность. Для того, чтобы его выразить через результаты наблюдений, удобно воспользоваться методом максимального правдоподобия.

4.3. Дисперсионный факторный анализ. Формулировка проверяемой гипотезы

Иногда необходимо знать, влияет ли одна случайная величина на другую. В более частной постановке требуется выяснить, значимо ли влияет изменение некоторого фактора на некоторый показатель. Это вопрос формулируется в виде гипотезы H_0, утверждающей, что две случайные величины X и Y независимы. Если гипотеза в результате проверки не подтвердится, значит – зависимы. В математической статистике гипотеза H_0 формирует некоторое утве[illegible], относящееся к одной [illegible]ли нескольким случайным величи[illegible]о утверждение либо истинно, либо ложно.

Существует м[illegible]тво разнообразных методов, позволяющих решать подобные [illegible]чи. Мы рассмотрим кратко всего лишь несколько.

Однофакторный дисперсионный анализ позволяет ответить на вопрос: насколько существенное влияние оказывает изменение некоторого фактора (или группы факторов в многофакторной модели) на исследуемую величину.

Задача однофакторного дисперсионного анализа формулируется следующим образом. Пусть нормально распределенная случайная величина X наблюдается при l постоянных (различных) значениях некоторого фактора. Результаты наблюдений образуют l выборок (групп) объемом n_k, $k = 1, 2, \ldots, l$, принадлежащих l генеральным совокупностям, имеющим равные, но неизвестные дисперсии σ^2 и математические ожидания m_k, $k = 1, 2, \ldots, l$.

Генеральная совокупность - это теоретически бесконечная выборка, в которой значения параметров распределения совпадают с теоретическими.

Проверяется гипотеза H_0:

$$m_1 = m_2 = \ldots = m_l = m.$$

Если проверка опровергнет гипотезу, т.е. математические ожидания групп не равны, то заключаем, что фактор значимо влияет на результаты наблюдений.

Действительно, если изменение фактора вызывает изменение средних значений анализируемого показателя, то это означает, что фактор значимо существенен, а это и требуется проверить. Значимость подразумевает степень достоверности вывода о том, справедлива или нет гипотеза H_0. Степень значимости задается уровнем значимости α или доверительной вероятностью p, где $p = 1 - \alpha$, $0 < \alpha < 1$ (почему используются две связанные величины p и α, а не какая-либо одна - толком непонятно). Выбор доверительной вероятности (либо α) произволен: обычно p выбирают из ряда: 0,9; 0,95; 0,99 (соответственно для α: 0,1; 0,05; 0,01). Чем больше доверительная вероятность, тем с большим основанием мы можем доверять результатам проверки гипотезы H_0, т.е. выводу о том, значимо ли влияет (или нет) фактор на выбранный показатель.

Обозначим x_{ik} i-е значение k - й выборки, $k=1, 2, \ldots, l$, $i=1, 2, \ldots, n_k$, $\bar{x}_k$ - выборочное среднее k -й группы:

$$\bar{x}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik} .$$

$\bar{x}$ - общее среднее:

$$\bar{x} = \frac{1}{n} \sum_{k=1}^{l} n_k \bar{x}_k = \frac{1}{n} \sum_{k=1}^{l} \sum_{i=1}^{n_k} x_{ik} ,$$

где
$$n = \sum_{k=1}^{l} n_k ;$$

n - общий объем наблюдения по всем выборкам.

Общая сумма квадратов отклонений наблюдений от общего среднего $\bar{x}$ может быть представлена так:

$$\sum_{k=1}^{l} \sum_{i=1}^{n_k} (x_{ik} - \bar{x})^2 = \sum_{k=1}^{l} n_k (\bar{x}_k - \bar{x})^2 +$$

$$+ \sum_{k=1}^{l} \sum_{i=1}^{n_k} (x_{ik} - \bar{x}_k)^2 . \tag{4.5}$$

Это основное тождество дисперсионного анализа.
Запишем его в виде

$$Q = Q_1 + Q_2, \tag{4.6}$$

где Q - общая сумма квадратов отклонений наблюдений от общего среднего;

Q_1 - сумма квадратов отклонений выборочных средних $\bar{x}_k$ от общего среднего $\bar{x}$ (т.е. между группами);

Q_2 - сумма квадратов отклонений наблюдений от выборочных средних групп (внутри групп).

Сами величины Q_1 и Q_2 являются случайными.

Без доказательства отметим, что отношение статистик

$$S_1^2 = \frac{Q_1}{l-1} \quad \text{и} \quad S_2^2 = \frac{Q_2}{n-1}$$

имеет распределение Фишера, поскольку S_1^2 и S_2^2 являются оценками дисперсий результатов наблюдений (4.5). Напомним, что статистиками являются функции элементов выборки. Таким образом можно записать

$$\frac{S_1^2}{S_2^2} = \frac{Q_1 \,/\, (l-1)}{Q_2 \,/\, (n-1)} = F_в(l-1,\ n-1). \qquad (4.7)$$

$F(k_1, k_2)$ – случайная величина, подчиняющаяся распределению Фишера, где $k_1 = l-1$, $k_2 = n-1$; k_1 и k_2 называются степенями свободы. Как любая случайная величина F характеризуется функцией плотности вероятности $f_F(x)$, математическим ожиданием $M[F]$ и т.п.

$$M[F] = \frac{k_2}{k_2 - 2}, \quad k_2 > 2.$$

Плотность распределения вероятности Фишера показана на рис. 4.2. Вероятность P определяется площадью, ограниченной F_p.

Для того чтобы проверить верна ли гипотеза H_0 о равенстве математических ожиданий, нужно вычислить (4.7) и сравнить со справочным значением $F_{1-\alpha}(l-1,\ n-1)$. Справочное значение находится в таблицах квантилей распределения Фишера по задаваемой вероятности $p = 1-\alpha$ и величинами $k_1 = l-1$ и $k_2 = n-1$.

Гипотеза H_0 принимается на уровне значимости α, если выборочное значение $F_в$, сосчитанное согласно (4.7), меньше квантили $F_{1-\alpha}$, т.е.

$$F_в < F_{1-\alpha}(l-1,\ n-1).$$

В этом случае $\bar{x}$ и $Q_2 / n - 1$ являются несмещенными оценками m и σ^2. Если

$$F_э \ge F_{1-\alpha} (l - 1, n - l),$$

то H_0 отклоняется, а среди средних $m_1, m_2, \dots, m_l$ имеются хотя бы два не равных друг другу.

Напомним, что квантиль - это число $F_{1-\alpha}$, для которого вероятность того, что $x < F_{1-\alpha}$, равна $p=1-\alpha$, т.е. можно записать:

$$P\ (x < F_{1-\alpha}) = 1 - \alpha$$

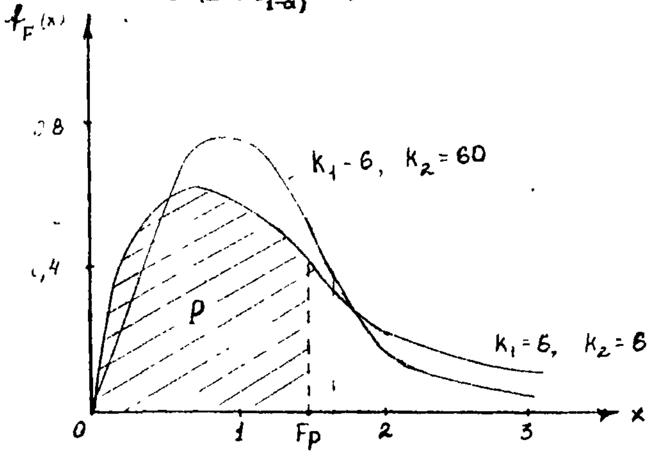


Рис. А 2

или, используя плотность вероятности $f(x)$,

$$1 - \alpha = \int_{-\infty}^{F_{1-\alpha}} f\ (x)\ dx.$$

В справочниках по математической статистики имеется большой выбор таблиц квантилей различных законов распределения (Стьюдента, χ^2 и т.д.).

Вычисление Q, Q_1 и Q_2 удобно проводить по формулам:

$$Q = A - C; \quad Q_1 = B - C; \quad Q_2 = A - B,$$

где

$$A = \sum_{k=1}^{l} \sum_{i=1}^{n_k} x_{ik}^2;$$

$$B = \sum_{k=1}^{l} \frac{1}{n_k} \left(\sum_{i=1}^{n_k} x_{ik} \right)^2;$$

$$C = \frac{1}{n} \left(\sum_{k=1}^{l} \sum_{i=1}^{n_k} x_{ik} \right)^2.$$

Если x_{ik} - сравнительно большие числа, то для упрощения вычислений расчет ведут, используя преобразованные элементы выборки

$$y_{ik} = x_{ik} - d,$$

где d - произвольная величина, для удобства выбираемая

$$d \approx \bar{x}.$$

Легко заметить, что при этом величины Q, Q_1 и Q_2 не изменяются.

В таблице 4.1 приведен пример использования факторного анализа. От трёх транспортных компаний (фактор - "Компания") поступает груз одинаковыми партиями (контейнерами). Номера наблюдений - партии, результаты наблюдения - число бракованных изделий в партии. Прочерки в таблице стоят там, где поступления груза не было. Требуется выяснить, случайно ли различие в среднем числа бракованных изделий для разных компаний (проверить H_0). Считая, что выборки получены из нормально распределенных генеральных совокупностей с равными дисперсиями, проверим гипотезу о равенстве средних при доверительной вероятности $p = 0{,}95$. Сначала вычисляются суммы по колонкам и нижним строчкам таблицы.

Таблица 4.I

№ наблюдения	№ выборки			Сумма
	I	2	3	
I	I	2	4	
2	3	3	5	
3	2	2	3	
4	I	I	–	
5	0	4	–	
6	2	–	–	
7	I	–	–	
n_k	7	5	3	n = I5
$\sum_{i=1}^{n_k} x_{ik}$	I0	I2	I2	34
$\sum_{i=1}^{n_k} x_{ik}^2$	20	34	50	A = 104
$\frac{1}{n_k}\left(\sum_{i=1}^{n_k} x_{ik}\right)^2$	14,28	28,8	48	B = 91,08

Затем вычисляются

$$C = \frac{34^2}{15} = 77,07;$$

$$Q = 104 - 77,07 = 26,93;$$

$$Q_1 = 91,08 - 77,07 = 14,01;$$

$$Q_2 = 104 - 91,08 = 12,92.$$

Находим выборочное значение статистики (4.7):

$$F_в = \frac{14,01 \;/\; (3 - 1)}{12,92 \;/\; (15 - 3)} = 6,51.$$

С другой стороны, из справочных таблиц находим, что $F_{0,95}(2;12) = 3,88$.

Так как выборочное значение $F_в$ превышает нормативное, равное 3,88, то *гипотеза о равенстве средних отклоняется*. Оказалось, что видимые и непосредственно из таблицы худшие показатели компании № 3 не случайны.

4.4. Элементы регрессивного анализа. Вычисление параметров модели методом наименьших квадратов

Одной из важных задач при рассмотрении нескольких случайных величин является нахождение связи между ними. Во многих случаях одна из переменных, например X, может быть неслучайной (т.е. принимать заданные значения), в то время как другая переменная Y имеет случайные флуктуации. Так, например, для перспективного планирования надо знать, как меняется по годам объем транспортных перевозок. Пусть имеется следующий временной ряд (табл. 4.2).

Таблица 4.2

Объем перевозок П за время t

t, год	5	10	15	20	25
П (усл.ед.)	59,3	59,8	60,1	64,9	70,2

Для построения прогноза требуется найти модель, т.е. зависимость П = f(t). Модель включает в себя вид функции и значения ее параметров. Обычно вид функции f выбирается заранее из общих физических соображений или из грубого оценочного характера зависимости. Из табл. 4.2 видно, что зависимость существенно нелинейна, с ростом t ее крутизна возрастает, поэтому можно принять квадратичный вид f по t.

Для выбранной модели остается найти числовые коэффициенты (параметры).

Предположим, что функциональная зависимость между двумя переменными известна с точностью до параметров $\beta_0, \beta_1, \ldots \beta_{k-1}$ и имеет вид

$$y = f(x;\ \beta_0, \beta_1, \ldots, \beta_{k-1}).$$

В дальнейшем неслучайная переменная будет обозначаться x. Требуется по результатам наблюдений $(x_i\ y_i)$, $i=1, 2, \ldots, n$ найти оценки неизвестных параметров $\beta_0, \beta_1, \ldots, \beta_{k-1}$.

Для решения данной задачи используют метод наименьших квадратов (МНК). В соответствии с ним в качестве оценок этих параметров (МНК – оценок) принимаются значения $\tilde{\beta}_0, \tilde{\beta}_1, \ldots, \tilde{\beta}_{k-1}$, дающие минимум функции

$$Q(\beta_0, \beta_1, \ldots, \beta_{k-1}) = \sum_{i=1}^{n} \left[y_i - f(x_i; \beta_0, \beta_1, \ldots, \beta_{k-1})\right]^2. \qquad (4.8)$$

Геометрический смысл такого метода прост. Через n точек плоскости XOY надо провести линию $f(x)$ такую, чтобы квадраты отклонений от нее (помечены стрелками) точек (x_i, y_i) были бы наименьшими (рис. 4.3). Квадраты берутся для того, чтобы знак отклонения роли не играл.

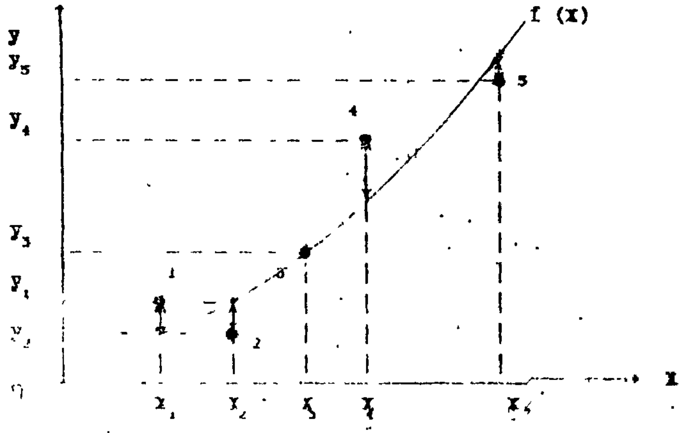


Рис. 4.3

Из необходимых условий экстремума функции $Q(\beta_0,\beta_1,\ldots,\beta_{k-1})$ находим, что МНК-оценки $\tilde\beta_0,\tilde\beta_1,\ldots,\tilde\beta_{k-1}$ являются решениями следующей системы k уравнений:

$$\frac{\partial Q(\beta_0,\beta_1,\ldots,\beta_{k-1})}{\partial \beta_j} = 0;\quad j=0,1,\ldots,k-1. \tag{4.9}$$

В общем случае эти уравнения нелинейны (по β_j) и их решение затруднительно. На практике чаще всего используют линейные по параметрам модели вида:

$$y = \beta_0 a_0(x)+\beta_1 a_1(x)+\ \ldots\ +\beta_{k-1}(x),a_{k-1}, \tag{4.10}$$

где $a_0(x),a_1(x),\ldots,a_{k-1}(x)$ – известные функции.

В этом случае система (4.9) при подстановке в нее (4.8) и (4.10) приводится к линейной алгебраической системе $\underset{\sim}{k}$ уравнений, называемой нормальной.

$$\begin{cases}
\beta_0\sum\limits_{i=1}^{n} a_0(x_i)a_0(x_i)+\beta_1\sum\limits_{i} a_1(x_i)a_0(x_i)+\ldots+\beta_{k-1}\sum\limits_{i} a_{k-1}(x_i)a_0(x_i)= \\
\qquad =\sum\limits_{i} y_i a_0(x_i); \\
\beta_0\sum\limits_{i} a_0(x_i)a_1(x_i)+\beta_1\sum\limits_{i} a_1(x_i)a_1(x_i)+\ldots+\beta_{k-1}\sum\limits_{i} a_{k-1}(x_i)a_1(x_i)= \\
\qquad =\sum\limits_{i} y_i a_1(x_i); \\
\cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot\quad\cdot \\
\beta_0\sum\limits_{i} a_0(x_i)a_{k-1}(x_i)+\beta_1\sum\limits_{i} a_1(x_i)a_{k-1}(x_i)+\ldots+ \\
\qquad \beta_{k-1}\sum\limits_{i} a_{k-1}(x_i)a_{k-1}(x_i)=\sum\limits_{i} y_i a_{k-1}(x_i),
\end{cases}$$

Построим модель по табл. 4.2. Попытаемся угадать зависимость между П и t (для этого предварительно полезно по данным таблицы построить график), придав ей вид:

$$П = a + b\,t + c\,t^2,$$

где a, b, c – искомые параметры модели.

Воспользуемся методом наименьших квадратов. Для удобства вычислений перейдем к новым переменным

$$x = \frac{t - 15}{5}, \quad y = 10\,(П - 60)$$

и вычислим оценки параметров линейной модели

$$y = \beta_0 + \beta_1 x + \beta_2 x^2.$$

Сравнивая эту модель с (4.10), замечаем, что

$$a_0(x) = 1, \; a_1(x) = x, \; a_2(x) = x^2,$$

а система нормальных уравнений имеет вид:

$$\begin{cases} \beta_0 5 + \beta_1 \sum\limits_{i=1}^{5} x_i + \beta_2 \sum\limits_{i=1}^{5} x_i^2 = \sum\limits_{i=1}^{5} y_i\,; \\ \beta_0 \sum\limits_{i=1}^{5} x_i + \beta_1 \sum\limits_{i=1}^{5} x_i^2 + \beta_2 \sum\limits_{i=1}^{5} x_i^3 = \sum\limits_{i=1}^{5} x_i y_i\,, \quad n = 5; \\ \beta_0 \sum\limits_{i=1}^{5} x_i^2 + \beta_1 \sum\limits_{i=1}^{5} x_i^3 + \beta_2 \sum\limits_{i=1}^{5} x_i^4 = \sum\limits_{i=1}^{5} x_i^2 y_i\,. \end{cases}$$

Для дальнейших вычислений составим таблицу 4.3.

Таблица 4.3

t	x	П	y	xy	x^2	yx^2	x^3	x^4
. 5	-2	59,3	-7	I4	4	-28	-8	I6
. I0	-I	59,8	-2	2	I	-2	-I	I
. I5	0	60,I	I	0	0	0	0	0
. 20	I	64,9	49	49	I	49	I	I
. 25	2	70,2	I02	204	4	408	8	I6
. Σ	0		I43	269	I0	427	0	34

С учетом полученных значений коэффициентов система нормальных уравнений такова:

$$\begin{cases} 5\beta_0 + 10\beta_2 = 143; \\ 10\beta_1 = 269; \\ 10\beta_0 + 34\beta_2 = 427. \end{cases}$$

Решая эту систему, получим

$$\tilde{\beta}_0 \approx 8{,}457, \quad \tilde{\beta}_1 \approx 26{,}9, \quad \tilde{\beta}_2 \approx 10{,}07.$$

Таким образом, зависимость между y и x имеет вид

$$y = 8{,}457 + 26{,}9x + 10{,}07x^2.$$

Переход к исходным переменным дает

$$(П - 60)10 = 8{,}457 + 26{,}9\,\frac{t-15}{5} + 10{,}07\left(\frac{t-15}{5}\right)^2,$$

откуда получаем окончательно $П = 61{,}84 - 0{,}67t + 0{,}04t^2$.

К этому уравнению нужно подходить с известной долей скептицизма. Если бы мы взяли другую модель, например, ограничились бы только первыми двумя членами (линейная по t модель), то для нее нашли бы уже другие коэффициенты. Какая из моделей лучше, по внешнему виду сказать нельзя: обе модели статистические и никак не учитывают физический характер причинно-следственных связей.

В заключение добавим, что таблица коэффициентов в системе нормальных уравнений называется регрессивной матрицей.

4.5. Линейная регрессия. Построение прогноза по линейной модели

Большое прикладное значение имеет случай линейной регрессии Y на x. При этом результаты наблюдений (x_i, y_i), $i=1,2,\dots,n$ описываются моделью

$$y = \beta_0 + \beta_1 x . \qquad (4.12)$$

Модель определяет прямую линию, показывающую, как в среднем изменяется величина Y при изменении x.

Строго говоря, регрессией величины Y на величину X является условное математическое ожидание $M_x[Y]$ как функции от x:

$$M_x[Y] = \int_{-\infty}^{+\infty} y\, f_{y|x}(y)\, d\,y,$$

где $f_{y|x}$ – плотность распределения условной вероятности y при заданном x; находится из совместной плотности распределения (рис. 4.4):

$$f_{y|x}(y) = f(x,y)/f(x),\ f(x) \neq 0,\ f(x) = \int_{-\infty}^{+\infty} f(x,y)\, d\,y.$$

В том случае, когда X и Y нормально распределены на плоскости, функция регрессии линейна (рис. 4.5) и выражается формулой:

$$y - \bar{y} = r\,\frac{\sigma_y}{\sigma_x}\,(x - \bar{x}),$$

где r – коэффициент корреляции между X и Y;

σ_x, σ_y – их среднеквадратические отклонения.

Как легко заметить, это выражение по форме совпадает с (4.12).

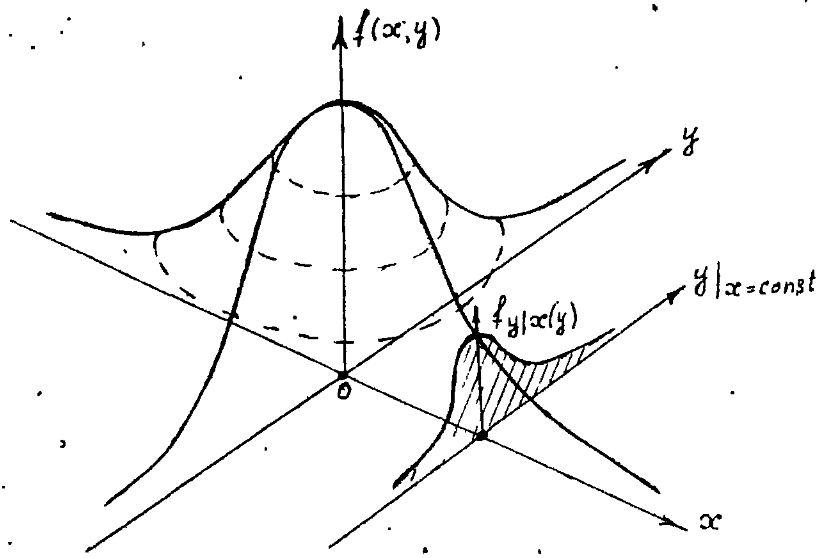


Рис. 4.4

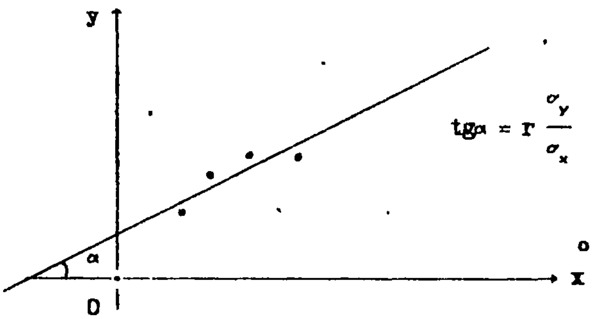


Рис. 4.5

Не надо только думать, что регрессия X на Y изобразится тем же графиком, лишь повернутым "набок", где аргументом будет Y. Линия регрессии в общем случае (кроме линейной связи) будет другой. Это легко понять для случая отсутствия связи между X и Y: тогда линии регрессии будут параллельны соответствующим осям, а значит – взаимно перпендикулярны.

В уравнении регрессии особое место занимает коэффициент корреляции r. Величины среднеквадратических отклонений σ_x σ_y помимо характеристик разброса определяются еще масштабом измерения и имеют размерности наблюдаемых величин. А вот коэффициент корреляции – величина безразмерная, именно она характеризует степень связи между случайными величинами. Значения r лежат в диапазоне

$$-1 \le r < 1.$$

Если $r = 0$, то как легко заметить, y никак не зависит от x. Стремление же модуля r к единице ($|r| \to 1$) означает усиление связи вплоть до жесткой функциональной (линейной) зависимости.

Для вычисления параметров β_0 и β_1 модели также используется метод наименьших квадратов. Его применение к модели (4.12) приводит к следующей последовательности вычислений.

$$\tilde{\beta}_0 = \bar{y} - \tilde{\beta}_1 \bar{x}, \quad \tilde{\beta}_1 = \frac{Q_{xy}}{Q_x}$$

где $$Q_{xy} = \sum_{i=1}^{n} x_i y_i - \frac{(\Sigma x_i)(\Sigma y_i)}{n} ; \qquad (4.13)$$

$$Q_x = \sum_{i=1}^{n} x_i^2 - \frac{(\Sigma x_i)^2}{n} ; \quad \bar{x} = \frac{\Sigma x_i}{n} ; \quad \bar{y} = \frac{\Sigma y_i}{n} .$$

При необходимости можно найти коэффициент корреляции по формуле

$$r = \frac{Q_{xy}}{\sqrt{Q_x Q_y}} ; \quad Q_y = \sum_{i=1}^{n} y_i^2 - \frac{(\Sigma y_i)^2}{n} .$$

Рассмотрим следующий пример. Пусть имеются сведения о тарифах на перевозки y и о стоимости авиатоплива x, сведенные в условных единицах в табл. 4.4.

Таблица 4.4

x	51	32	80	73	64	45	83	44	93
y	52,7	15,2	89,5	94,8	76	39,3	114,8	36,5	137,4

x	28	35	40	29	53	58	65	75
y	5,3	20,7	21,7	9,2	55,4	64,3	79,1	101

Требуется найти оценки параметров линейной регрессии Y на x (n = 17). По формулам (4.13) находим

$$\sum_{i=1}^{17} x_i = 948; \quad \sum_{i=1}^{17} y_i = 1012,3; \quad \bar{x} \approx 55,8; \quad \bar{y} \approx 59,5;$$

$$\sum_{i=1}^{17} x_i^2 = 59422; \quad \sum_{i=1}^{17} x_i y_i = 69165,4;$$

$$Q_x = 59422 - \frac{948^2}{17} \approx 6557;$$

$$Q_{xy} = 69165,4 - \frac{948 \cdot 1012,3}{17} \approx 12715.$$

Оценки коэффициентов регрессии равны

$$\tilde{\beta}_1 = \frac{12715}{6557} \approx 1,94;$$

$$\tilde{\beta}_0 = 59,5 - 1,94 \cdot 55,8 \approx - 48,6.$$

Уравнение регрессии имеет вид

$$y = - 48,6 + 1,94\ x.$$

В том случае, когда на Y влияет несколько случайных величин, рассматривается множественная линейная регрессия с зависимостью вида

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k.$$

В модели предполагается, что между переменными X_i нет линейной взаимосвязи $(i = 1,2,\ldots,k)$.

4.6. Обработка данных непараметрическими методами. Понятие о рангах. Ранговая корреляция

В рассмотренных ранее методах обработки статистической информации предполагалось, что распределение генеральной совокупности подчиняется нормальному закону. В некоторых случаях такое утверждение принять нельзя. Особенно в условиях, когда физическая природа процессов неизвестна (можно вспомнить статистики Бозе и Ферми из физики). В таких ситуациях применяют методы, не зависящие от распределения генеральной совокупности, называемые также непараметрическими методами.

Непараметрические методы используют не сами численные значения элементов выборки, а структурные свойства выборки. К одной из характеристик таких свойств относится ранг. Если все элементы выборки упорядочить в порядке возрастания их значений, то получим вариационный ряд. Номер элемента в этом ряду называется рангом. Если несколько элементов ряда совпадают по величине, то каждому из них присваивается ранг, равный среднему арифметическому их номеров.

Существуют и другие структурные свойства, например, положение элемента относительно выборочной медианы (делит площадь под $f(x)$ пополам): если значение элемента больше оценки медианы, то элементу присваивается знак плюс, если меньше - минус.

Непараметрические методы могут применяться при достаточно общих предположениях относительно генеральной совокупности, они более просты в вычислениях, но их достоверность ниже (говорят, что мощность критериев меньше).

С помощью непараметрических методов можно также оценить связь двух случайных величин Y и X. Например, с помощью коэффициента ранговой корреляции. Аналогом обычного коэффициента корреляции (в частности, присутствующем в модели линейной регрессии) является ранговая корреляция.

Пусть (x_i, y_i), $i = 1, 2, \ldots, n$ выборка наблюдений непрерывных случайных величин Y и X. Каждому значению x_i поставим в соответствие ранг x_i', т.е. номер элемента в вариационном ряду $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$. Аналогичным образом определим ранги y_i' элементов y_i. Каждой паре (x_i, y_i), таким образом, соответствует пара рангов (x_i', y_i'). Поскольку X и Y случайные величины, то и их ранги X' и Y' тоже можно рассматривать как случайные величины. Формально, применяя формулу для коэффициента корреляции двух обычных случайных величин

$$r = \frac{\sum\limits_{i} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum\limits_{i} (x_i - \bar{x})^2 \sum\limits_{i} (y_i - \bar{y})^2}} \qquad (4.14)$$

к выборке рангов (x_i', y_i'), приходим к выражению, определяющему коэффициент ранговой корреляции Спирмена r_s.

$$r_s = 1 - \frac{6 \sum_i (x_i' - y_i')^2}{n (n^2 - 1)}, \qquad (4.15)$$

Вычисления по формуле (4.15) проще, чем по (4.14), и не требуют соблюдения предположения о нормальном (двумерном) распределении X и Y. Формула (4.14) - всего лишь другой вид записи выражения для r из (4.13).

Коэффициент корреляции, присутствующий в линейной модели регрессии, имеет самостоятельное значение. Из модели видно, что если r=0, то величины X и Y линейно независимы. Поэтому во многих практических случаях достаточно вычислить r, чтобы сделать заключение о зависимости двух случайных величин. Формула (4.14) достаточно "прозрачна". Легко заметить, что если с ростом x_i относительно среднего значения $\bar{x}$ растет и y_i относительно своего среднего $\bar{y}$ (т.е. обе разности положительны), то и сумма в числителе возрастает. Значит $r > 0$ (если y_i убывает, то $r < 0$). Если же с ростом x_i значения y_i примерно симметрично колеблются около среднего $\bar{y}$, то число положительных и отрицательных слагаемых в числителе примерно равно, а значит вся сумма будет равна нулю, следовательно r=0.

Приведем пример. Пусть X - объем пассажирских перевозок (в тыс. пассажиро-километров); Y - грузовых перевозок (в тыс.т-км). Значения x и y приведены в табл. 4.5. Определим по ним коэффициент ранговой корреляции.

Таблица 4.5

x	68,8	63,3	75,7	67,2	71,3	72,8	76,5	63,5	69,9	71,4
y	167,0	113,3	199,9	153,6	150,8	181,2	173,1	115,4	125,6	166,2

Для решения задачи определим ранги элементов исходной выборки. Предварительно перепишем исходную выборку, упорядочив ее элементы по верхней строке (т.е. по значениям x_i). Затем в третью строчку табл. 4.6 проставим ранги для значений y_i. В результате получим табл. 4.6.

Таблица 4.6

x	63,3	63,5	67,2	68,8	69,9	71,3	71,4	72,8	75,7	76,5
y	113,3	115,4	153,6	167,0	125,6	150,8	166,2	181,2	159,9	173,1
y'	1	2	5	8	3	4	7	10	6	9

Таким образом, упорядоченной по элементам x_i выборке соответствует следующая последовательность пар рангов и их разностей (табл. 4.7):

Таблица 4.7

x_i'	1	2	3	4	5	6	7	8	9	10
y_i'	1	2	5	8	3	4	7	10	6	9
$x_i'-y_i'$	0	0	-2	-4	2	2	0	-2	3	1

Так как $\sum_{i=1}^{10} (x_i' - y_i')^2 = 42$, то согласно (4.15) находим

$$r_s = 1 - \frac{6 \cdot 42}{10\,(10^2 - 1)} \approx 0{,}745.$$

Это значение близко к единице и можно утверждать, что перевозки X и Y зависимы.

В заключение заметим, что применение статистических методов обосновано только тогда, когда условия наблюдений воспроизводимы и повторяются практически неизменными. Распространение же результатов статистического анализа на новые условия всегда опасно.

4.7. Нестохастическая неопределенность. Метод экспертных оценок

Теория вероятностей оперирует лишь со случайными величинами (или функциями). А именно такими, в которых случайны конкретные значения, но не случайны характеристики (параметры): законы распределения, математические ожидания, дисперсии и т.п. Они могут быть неизвестны, но в принципе существуют. Говорят, что случайные величины обладают статистической устойчивостью.

Проектирование АСУ, основанное на "оптимизации в среднем" при учете случайных факторов, исходит именно из неслучайного характера основных параметров случайных величин. Считается что, при многократном повторении каких-либо действий "недостача" показателя эффективности в одном случае компенсируется его "избытком" в другом.

Однако, когда речь идет не о повторяемой, массовой операции, а о единичной, уникальной, такое рассуждение становится неоправданным. Что толку в том, что операция в среднем приносит большой выигрыш, если в данном единичном случае она может дотла разорить.

На практике часто встречаются неопределенные, вроде бы случайные величины, для которых в принципе не существует понятия закона распределения или других вероятностных характеристик. Тогда говорят о нестохастической неопределенности. Она возникает в явлениях, подверженных множеству неподдающихся учету факторов, причем сами явления принципиально неповторяемые. Как изменится цена на то или иное произведение искусства (почтовую марку,...)? Всем известно, что картины импрессионистов сначала валялись на чердаках и гибли на свалках. Или какой тип процессора будет наиболее конкурентоспособен через 5 лет? Тем не менее, для проектирования АСУ производством, перевозками, сбытом продукции надо как-то отвечать на подобные вопросы. В таких случаях весьма полезным, а иногда единственно возможным, является метод экспертных оценок.

Мы уже встречались ранее с привлечением экспертов для решения задач при рассмотрении экспертных систем. Однако тогда, в основном, от экспертов требовалось узнать какими правилами надо пользоваться, какой логике следовать, чтобы придти к нужному решению. Из этих правил составлялась база знаний ЭС. Сейчас нас интересует не процесс, а результат, прогноз.

В основе метода экспертных оценок лежит неявное предположение о том, что высказывания отдельных экспертов как-то колеблются, флуктуируют около истинного значения. Поэтому достаточно как-то осреднить экспертные оценки по аналогии со случайными величинами, и мы придем к истине. Однако, как мы уже видели, такой метод не застрахован от ошибок. Можно сказать, что это плохой метод, но это лучше, чем ничего.

Примером экспертной оценки служит известная всем отметка за успеваемость (обычно по пятибалльной шкале), и часто связанное с ней чувство неудовлетворенности.

Экспертное исследование предполагает как оценку самих событий (например, получение оценок по 100 балльной шкале или определение рангов), так и оценку компетентности экспертов.

Экспертная оценка включает в себя следующие основные этапы:

1. Получение обобщенных мнений экспертов.

2. Определение степени согласованности высказываний экспертов.

3. Определение компетентности и активности экспертов.

Результаты экспертного опроса сводятся в таблицу, где указываются оценки c_{ij}, выставленные i-м экспертом j-му событию (оцениваемой ситуации). Чаще всего используются следующие показатели.

1. Показатели обобщенного мнения.

Это осредненные по всем экспертам оценки. Среднее статистическое значение $M[c_{ij}]$ – это средняя оценка (в баллах) j-го события.

$$\bar{c}_j = M[c_{ij}] = \sum_{i=1}^{m_j} c_{ij} / m_j , \qquad m_j \le m ,$$

где m_j – число экспертов, принимавших участие в оценке j-го события (из-за некомпетентности часть экспертов в данной экспертизе может не участвовать).

2. Суммарный ранг R_j события.

Все события ранжируются каждым экспертом по важности. Наиболее важное событие имеет минимальный ранг. Например, номера мест, выставляемые фигуристам на соревнованиях (помимо баллов). В результате j-е событие получает ранг R_{ij}, присвоенный i-м экспертом. Сумма рангов R_i, назначенных m_i числом экспертов j-му событию, определяется

$$R_j = \sum_{i=1}^{m_i} R_{ij} \quad , \quad m_i \le m \quad ,$$

где m_i – число экспертов, оценивших хотя бы одно событие. Если i-й эксперт не оценил какое-либо событие, то его ранг R_{ij} назначается, например, по среднему $\bar{c}_j$.

3. Показатели степени согласованности мнений экспертов.

Согласованность мнений часто оценивается коэффициентом вариации ν_j, который по смыслу похож на отношение среднеквадратического отклонения обычной случайной величины к ее среднему значению.

$$\nu_j = \frac{\sigma_j}{\bar{c}_j} = \frac{\sqrt{D_j}}{\bar{c}_j}$$

где

$$\sigma_j^2 = D_j = \sum_{i=1}^{m_j} (c_{ij} - \bar{c}_j)^2 / m_j \; .$$

Чем меньше ν_j, тем согласованнее мнение о j-м событии (при этом мы помним, что вопросы истины не решаются большинством голосов).

Другим показателем согласованности мнений является коэффициент конкордации (т.е. согласия) W. Он относится к ранговым критериям.

Определение компетентности и активности экспертов – это, по существу, самостоятельное новое экспертное исследование.

5. ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ В АСУ

5.1. Задача линейного программирования

Современные АСУ сложными объектами (такими как транспортные системы, производственные комплексы, энергетические узлы и т.д.) решают целый блок задач: собирают, сортируют, предварительно статистически обрабатывают исходные данные, а затем, используя математические методы принятия решений, формируют рекомендации по управлению этими объектами. Методы принятия решений (рекомендации) составляют специальный класс прикладных математических задач. Их особенностью является поиск, выбор не каких-либо решений вообще, а оптимальных в смысле принятых критериев. В соответствии с этим, математическое описание задачи должно включать в себя как формализованное описание условий задачи, так и критерия эффективности, называемого также целевой функцией. Рассмотрим пример.

Фирма производит изделия трех видов: U_1, U_2, U_3. На выпуск каждого вида заключены контракты, согласно которым должно быть выпущено не менее b_1 единиц изделия U_1, не менее b_2 – изделия U_2 и не менее b_3 – изделия U_3. Выпуск может быть превышен, но в определенных границах: условия спроса и складирования ограничивают количество произведенных единиц каждого типа: не более соответственно β_1, β_2, β_3 единиц. На изготовление идет сырье (четыре вида): S_1, S_2, S_3, S_4, причем запасы ограничены числами γ_1, γ_2, γ_3, γ_4.

Известно количество сырья a_{ij} каждого i-го вида ($i=1,2,3,4$), которое идет на изготовление j-го вида изделия ($j=1,2,3$). Значения a_{ij} сведены в следующую матрицу.

Сырье	Изделие		
	U_1	U_2	U_3
s_1	a_{11}	a_{12}	a_{13}
s_2	a_{21}	a_{22}	a_{23}
s_3	a_{31}	a_{32}	a_{33}
s_4	a_{41}	a_{42}	a_{43}

При реализации одно изделие U_1 приносит прибыль c_1, U_2 – прибыль c_2, U_3 – прибыль c_3. Требуется так спланировать производство – определить сколько каких изделий производить, чтобы суммарная прибыль обращалась в максимум.

Запишем условие задачи в математическом виде. Обозначим элементами решения x_1, x_2, x_3 – количества единиц изделий U_1, U_2, U_3, которые мы произведем (план). Обязательность выполнения контрактов выразится в виде ограничений-неравенств:

$$x_1 \geq b_1; \quad x_2 \geq b_2; \quad x_3 \geq b_3 .$$

Отсутствие излишней продукции даст еще три неравенства;

$$x_1 \leq \beta_1; \quad x_2 \leq \beta_2; \quad x_3 \leq \beta_3 .$$

Кроме того, должно хватить сырья. Соответственно четырем видам сырья будем иметь четыре ограничения-неравенства:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \le r_1; \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \le r_2; \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \le r_3; \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 \le r_4. \end{cases}$$

Прибыль, приносимая планом (x_1, x_2, x_3), будет равна

$$f = c_1x_1 + c_2x_2 + c_3x_3.$$

Функция $f(x_1, x_2, x_3)$ - это целевая функция.

В общем виде задача математического программирования заключается в нахождении решения (набора чисел), доставляющего экстремум некоторой целевой функции при соблюдении системы неравенств, описывающих условия - ограничения задачи. Термин "программирование" заимствован из английского языка и более точно может быть переведен как „ планирование. Методы математического программирования используются при планировании всевозможных операций (производство, транспорт и т.п.) и часто называются методами исследования операций.

Не существует общих методов решения задач математического программирования. Целесообразно рассматривать отдельные классы (виды) задач. Для каждого такого класса удается сформулировать алгоритм решения, приемлемый только для данного класса задач. Наиболее разработанными являются задачи линейного программирования.

Для задач линейного программирования характерно два условия:

- целевая функция (показатель эффективности) линейно зависит от элементов решения $x_1, x_2, \ldots, x_n$;
- ограничения, налагаемые на элементы решения, имеют вид линейных равенств или неравенств относительно $x_1, x_2, \ldots, x_n$.

Переменные x_1, x_2, $\ldots, x_n$ могут быть подчинены или нет требованию неотрицательности. Одна и та же задача линейного программирования может быть записана в различной форме. Говорят, что задача линейного программирования записана в канонической форме, если все ее ограничения, кроме $x_j \geq 0$, $j = 1,2,\ldots n$, представляют собой равенства. Если ограничения содержат неравенства, то задача записана в стандартной форме.

Математическая формулировка канонической формы следующая. Дана система линейных уравнений:

$$\begin{cases} a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n = b_1; \\ a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n = b_2; \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n = b_m \end{cases} \qquad (5.1)$$

и линейная функция

$$f = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n . \qquad (5.2)$$

Требуется найти такое неотрицательное решение

$$x_1 \geq 0; \ x_2 \geq 0; \ \ldots ; \ x_n \geq 0 \qquad (5.3)$$

системы (5.1), при котором функция f принимает наименьшее значение (минимизируется).

Уравнения (5.1) называются ограничениями данной задачи. Строго говоря, условия (5.3) тоже являются ограничениями, однако, их не принято так называть, поскольку они не характерны для данной задачи, а являются общими для всех задач линейного программирования.

Для записи задачи линейного программирования в различной форме применяются следующие приемы.

1. Точка минимума функции $f(x)$ совпадает с точкой максимума функции $-f(x)$, т.е.

$$\min \{ f(x)\} = \max \{-f(x)\}.$$

2. Ограничения в виде неравенств

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

можно представить в виде равенств, использовав новые переменные $x_{n+i} \geq 0$, называемые слабыми:

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

Если на переменную x_j не наложено условие неотрицательности, ее можно заменить двумя неотрицательными переменными x_j^+ x_j^-, положив

$$x_j = x_j^+ - x_j^- \ ; \quad x_j^+ \geq 0; \quad x_j^- \geq 0.$$

Если имеется k таких переменных, то их можно заменить $k + 1$ неотрицательными переменными x_j' и x_0; положив $x_j = x_j' - x_0$, $x_j' \geq 0$, $x_0 \geq 0$.

Система ограничений в виде равенств и неравенств образует выпуклое множество – выпуклый многогранник. Целевая функция – некоторую плоскость (гиперплоскость). Задача линейного программирования является частным случаем задачи выпуклого программирования.

5.2. Геометрический смысл задачи линейного программирования

Пусть ограничения задачи линейного программирования записаны в виде (5.I), причем все m уравнений линейно независимы. Это означает, что никакое из них не может быть получено из остальных. Очевидно, задача такова, что $n>m$. Если $n=m$, то нечего оптимизировать, а надо решать систему (5.I), если же $n<m$, то система переопределена.

Известно, что m линейно независимых уравнений всегда можно разрешить относительно каких-то m базисных переменных, выразив их через остальные, свободные. Мы как бы принимаем, что свободные переменные являются известными числами. В таком случае система (5.1) становится имеющей лишь m неизвестных, а их однозначно можно найти, решив систему из m уравнений. Число свободных переменных $k=n-m$. Для наглядности примем $k=2$ и обозначим свободные переменные x_1 и x_2. Тогда остальные: $x_3, x_4, \ldots, x_n$ – базисные. Мы можем вместо m уравнений (5.1) получить тоже m уравнений, но записанных в другой форме, разрешенных относительно $x_3, x_4, \ldots, x_n$. При этом мы помним, что все $x_i \geq 0$, $i=1,2,\ldots,n$. Теоретически для такого разрешения можно воспользоваться формулой Крамера (на практике используют метод искусственного базиса, см. далее), тогда

$$\begin{cases} x_3 = \alpha_{31}x_1 + \alpha_{32}x_2 + \beta_3; \\ x_4 = \alpha_{41}x_1 + \alpha_{42}x_2 + \beta_4; \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ x_n = \alpha_{n1}x_1 + \alpha_{n2}x_2 + \beta_n. \end{cases} \tag{5.4}$$

Система (5.4) содержит $m=n-2$ уравнений ($k=2$). Будем изображать пару значений свободных переменных точкой с координатами x_1, x_2 (рис. 5.1). Так как переменные x_1, x_2 должны быть неотрицательными, то допустимые значения свободных переменных лежат только выше оси OX_1 и правее оси OX_2. Теперь построим на плоскости x_1Ox_2 область допустимых решений (ОДР). Она определяется уравнениями (5.4) и требованиями неотрицательности всех переменных. Положим в первом уравнении (5.4) предельное значение $x_3=0$; получим уравнение граничной прямой:

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

На этой прямой $x_3 = 0$; по одну сторону от нее $x_3>0$, по другую $x_3<0$. Отметим штриховкой ту сторону, где $x_3>0$. Аналогично поступим и со всеми остальными условиями (5.4).

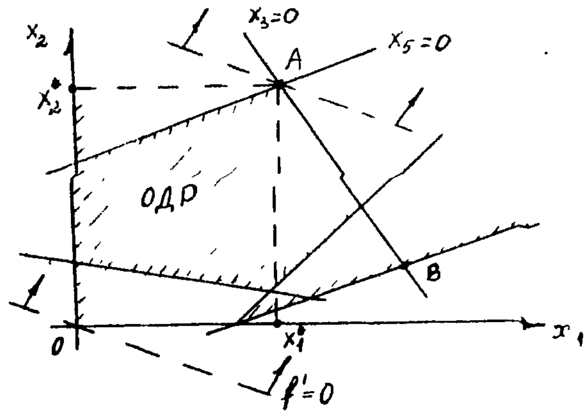


Рис. 5.I

Таким образом, мы построили n прямых: две оси координат (OX_1 и OX_2) и n-2 прямых $x_3=0$, $x_4=0,\dots,x_n=0$. Каждая из них определяет допустимую полуплоскость, где может лежать решение. Часть первого координатного угла, принадлежащая одновременно всем этим полуплоскостям, и есть ОДР. Заметим, что этих решений - бесконечное множество, т.к. любая пара значений из ОДР годится, а по x_1 и x_2 могу быть определены из системы (5.4) и базисные переменные.

Заметим, что при k=I наша задача выродилась бы в одномерную: условия - ограничения определяли бы точки на прямой, а ОДР стала бы некоторым отрезком на ней.

Для нахождения оптимального решения из множества допустимых дадим геометрическую интерпретацию условию минимизации f (5.2). Подставим выражения (5.4) в формулу (5.2). После приведения подобных членов получим:

$$f = r_1x_1 + r_2x_2 + r_0,$$

где r_1, r_2 - какие-то коэффициенты;

r_0 - свободный член, которого вначале у функции f не было: теперь при переходе к переменным x_1 и x_2 он мог и появиться.

Однако для геометрических построений мы его отбросим: ведь, очевидно, минимум f достигается при тех же значениях x_1 и x_2, что и минимум однородной функции (без свободного члена):

$$f' = r_1x_1 + r_2x_2.$$

Положим $f'=0$, т.е. $r_1x_1 + r_2x_2 = 0$.

На рис. 5.I это прямая, проходящая через начало координат. Придавая f' какие-то значения C_1, $C_2, \ldots$, прямая будет перемещаться параллельно самой себе; при перемещении в одну сторону f' будет возрастать, в другую – убывать. Пусть убывание будет в направлении отмеченном стрелками, т.е. "направо-вверх", но могло быть и наоборот: все зависит от коэффициентов r_1, r_2 (при $r_1<0$ и $r_2<0$ перемещение прямой будет по стрелке). Понятно, что чем дальше в направлении стрелок перемещается прямая, тем меньшее значение принимает f'. Очевидно, что минимум f' достигнет в точке A – крайней в ОДР. Заметим, что там, где f' стремится к минимуму, ОДР должна быть ограничена, в противном случае задача решения не имеет. Может также случиться, что прямая f' совместится с граничной прямой ОДР, тогда задача имеет бесконечное множество решений. Однако из этого бесконечного множества можно ограничиться только вершинами, тогда мы получим универсальное правило. Минимум f' достигается в одной из вершин ОДР, получающихся на пересечении 2-х прямых, построенных из условия равенства нулю базисных переменных. В общем случае имеем правило.

Оптимальное решение задачи линейного программирования достигается при такой совокупности значений переменных $x_1, x_2, \ldots, x_n$, где, по крайней мере, k из них обращаются в нуль, а остальные неотрицательны.

При формулировке правила мы обобщили рассмотренный случай с $k=2$ на произвольное $k > 2$. Так при $k = 3$ ОДР представляет собой многогранник, но по прежнему оптимальное решение достигается в одной из его вершин. В общем случае имеется k-мерное пространство и k-мерный многогранник с вершинами, называемыми (только в задаче линейного программирования) опорными точками. Из 5.I видно, что не все точки, в которых к-переменные обращаются в нуль, принадлежат ОДР (например, (•)В). Таким образом, правило задает необходимое, но не достаточное условие оптимальности.

Приведенное правило служит основой большинства рабочих методов оптимизации в линейном программировании: надо перебирать опорные точки из ОДР и вычислять для них значение f; та точка, в которой f минимально, и дает решение задачи. К сожалению, число переборов велико и равно числу комбинаторных сочетаний свободных и базисных переменных C_n^m. Для $n=30$ и $m=10$ число $C_{30}^{10} = 30045015$, а эта задача – далеко не из сложных. Поэтому требуются не "слепые" переборы, а целенаправленный поиск, в котором каждое последующее решение будет не хуже предыдущего.

Рассмотрим следующий простой пример, позволяющий решить задачу геометрически. Авиакомпания занимается перевозкой двух типов грузов. Возможности на перевозку связаны ограничениями по весу и габаритам. Вес ограничен 150 единицами груза любого типа. Единица веса груза второго типа занимает в 3 раза меньший объем, чем первого. Объем загрузки ограничен 300 условными единицами по габаритам (для которых плотность второго груза принимается за единицу). Требуется найти план перевозок (x_1 единиц (веса) груза первого типа и x_2 – второго), если известно, что перевозка груза первого типа приносит вдвое большую прибыль, чем второго (по весу).

Приведенные условия запишутся в виде неравенств

$$\begin{cases} 3x_1 + x_2 \le 300; & (5.5) \\ x_1 + x_2 \le 150. & (5.6) \end{cases}$$

Показатель эффективности – линейная функция имеет вид

$$f = 2x_1 + x_2.$$

Среди $x_1, x_2 \ge 0$ надо найти такие, которые сообщают функции f наибольшее значение.

Эта задача имеет простое геометрическое решение. Введем прямоугольную систему $x_1 0 x_2$. Тогда множество решений изобразится заштрихованной областью на рис. 5.2.

Для построения проведем сначала прямую $3x_1+x_2=300$ по двум точкам: пусть $x_1=0$, тогда $x_2=300$; при $x_2=0$ $x_1=100$. Нанесем

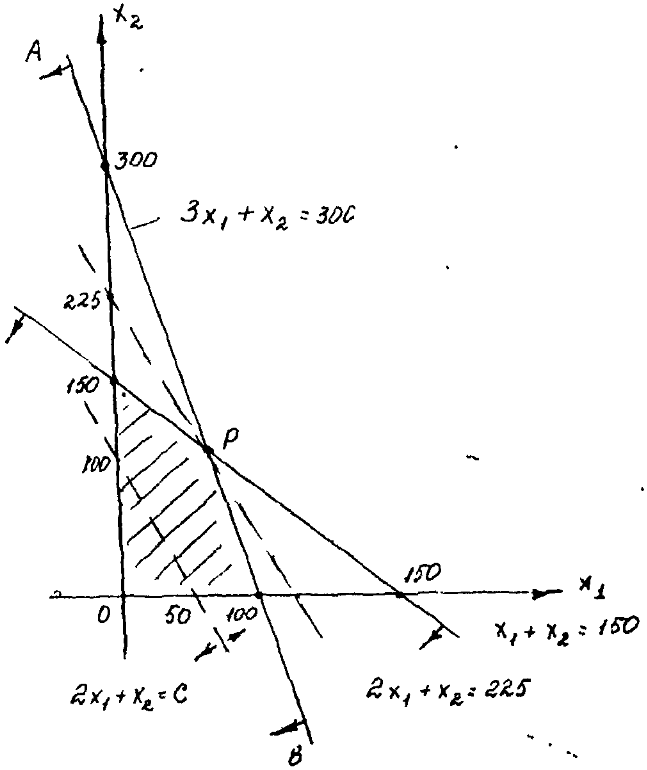


Рис. 5.2

точки (0,300) и (100,0) на графике и проведем прямую АВ. Чтобы определить какая часть плоскости определяется неравенством (5.5), подставим в него координаты точки (0,0). Получили верное неравенство, определяющее полуплоскость, содержащую начало координат. Стрелки на прямой АВ указывают эту полуплоскость. Аналогично строится прямая и для второго неравенства.

Для построения линейной функции воспользуемся следующим приемом: запишем ее в виде

$$2x_1 + x_2 = C,$$

где C можно произвольно менять для того, чтобы она приняла максимальное значение, не нарушая при этом ограничения (5.5) и

(5.6). Пусть для начала $C=100$, тогда прямая пройдет через точки (0,100) и (50,0). Изобразим ее пунктирной линией, показывая, что изменяя C линия будет перемещаться параллельно самой себе. Из рисунка следует, что решением системы неравенств при условии max f является (•) P (75,75). Действительно, с ростом числа C в уравнении для f данная прямая смещается вверх. Самое большое значение C соответствует (•) P - пересечению 2-х прямых, получающихся из (5.5) и (5.6). Это значение и определяет максимум $f = 225$.

Если бы мы перешли к канонической форме записи задачи, то неравенства заменили бы уравнениями, введя дополнительно две слабые переменные. В итоге число переменных $n=4$ и $k=n-m=2$ при двух ($m=2$) условиях-ограничениях. Таким образом, как и в случае неравенств мы опять получили геометрическую задачу на плоскости с двумя переменными. Очевидно, что решение ее будет таким же, как и полученное ранее.

5.3. Симплекс-метод

Реальные задачи линейного программирования содержат, как правило, большое число ограничений и неизвестных. Для их решения разработаны различные приемы, учитывающие специфику данного класса задач. Например, для транспортной задачи существует простой и эффективный алгоритм, обусловленный особенностями ее системы ограничений. Однако разработаны и общие методы, позволяющие найти решение любой задачи линейного программирования в обозримое число шагов (правда, иногда это будет "стрельбой из пушки по воробьям"). Наиболее распространенным является симплекс-метод.

Симплекс-метод заключается в целенаправленном переборе точек многогранника ОДР. В его основу положено правило, рассмотренное в предыдущем параграфе:

- берутся k свободных переменных и приравниваются нулю (но так, чтобы это была точка из ОДР);

- проверяется возможность минимизации f, и если она существует, то в системе ограничений меняется базис, т.е. определенная базисная переменная делается свободной, а свободная - базисной; при этом вся система переразрешается относительно нового базиса и процедура повторяется сначала;

– если такая возможность исчерпана, то мы пришли к оптимальному решению (или же выясняется, что задача решения не имеет).

Поясним симплекс-метод на примере. Для начала требуется, чтобы система уравнений, заданная в канонической форме, была приведена к виду (5.4), где все $\beta_3, \beta_4, \ldots, \beta_n$ (при k=2) были бы неотрицательны. Тогда при нулевых свободных переменных базисные будут неотрицательными, как это следует из (5.4), а значит мы имеем какую-то опорную точку на многограннике ОДР. От нее мы и будем "танцевать". Можно ли на самом деле привести систему к такому виду и как это сделать – этот вопрос мы рассмотрим позже.

Пусть заданная система ограничений и функция f приведены к такому виду:

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

Здесь неизвестные x_1, x_2, x_3 образуют базис. Соответствующее базисное решение (при $x_4=0$ и $x_5=0$) есть (1, 2, 3, 0, 0); значение f для этого решения равно 0.

Посмотрим, не является ли это решение минимальным. Поскольку x_5 входит в f с отрицательным знаком, то мы можем уменьшить f, увеличив x_5 (сохранив $x_4=0$). Однако поскольку при изменении x_5 будут меняться также x_1, x_2, x_3, нужно следить за тем, чтобы ни одно из них не стало отрицательным.

Так как увеличение x_5 приводит к увеличению x_1, то для x_1 такой опасности нет. Рассматривая же x_2 и x_3, находим, что x_5 может быть увеличено только до 2 (и не более, иначе x_2 станет отрицательным), что дает $x_1=5$, $x_2=0$, $x_3=1$. В итоге получаем новую опорную точку (5, 0, 1, 0, 2), в которой число положительных неизвестных по-прежнему равно трем. Для этого решения $f=-2$.

Новый базис теперь состоит из x_1, x_3, x_5. Чтобы осуществить соответствующее переразрешение системы ограничений, нужно выразить эти неизвестные через x_2, x_4. Начинаем с уравнения для x_2 (новой свободной переменной), из которого выражаем x_5 (новую базисную переменную):

$$x_5 = 2 + 2x_4 - x_2;$$

затем выражаем x_1, x_3 и f:

$$x_1 = 1 - x_4 + 2(2 + 2x_4 - x_2);$$

$$x_3 = 3 - 3x_4 - (2 + 2x_4 - x_2);$$

$$f = x_4 - (2 + 2x_4 - x_2).$$

Преобразуя, приходим к виду

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

$$f = -2 - x_4 + x_2.$$

Выясним, нельзя ли еще уменьшить f. Коэффициент при x_4 в выражении для f отрицателен, поэтому можно попытаться уменьшить f, увеличив x_4 (без изменения $x_2=0$). Первое и третье уравнения этому не препятствуют, а из второго видно, что x_4 можно увеличить до 1/5 (если более, то x_3 станет отрицательным). При этом $f = -11/5$. Полагая $x_4 = 1/5$, $x_2 = 0$, получим $x_1 = 28/5$, $x_3 = 0$, $x_5 = 12/5$. В итоге имеем новое допустимое решение (28/5, 0, 0, 1/5, 12/5).

Новый базис теперь состоит из x_1, x_4, x_5. Из уравнения для x_3 (новой свободной переменной) выражаем x_4 (новая базисная):

$$x_4 = \frac{1}{5} - \frac{1}{5}x_3 + \frac{1}{5}x_2 .$$

Делая соответствующие подстановки приходим к системе:

$$
\begin{cases}
x_1 = \dfrac{28}{5} - \dfrac{3}{5} x_3 - \dfrac{7}{5} x_2; \\
x_4 + \dfrac{1}{5} - \dfrac{1}{5} x_3 + \dfrac{1}{5} x_2; \\
x_5 = \dfrac{12}{5} - \dfrac{2}{5} x_3 - \dfrac{3}{5} x_2;
\end{cases}
$$

$$
f = \frac{11}{5} + \frac{1}{5} x_3 + \frac{4}{5} x_2 .
$$

Теперь в функции f обе переменные входят с положительными коэффициентами. Поэтому, если увеличить x_2 и x_3, то f будет возрастать. Таким образом заключаем, что полученное решение (28/5, 0, 0, I/5, I2/5) является оптимальным, искомый минимум $f = -\dfrac{II}{5}$; задача решена.

Изложенный ход рассуждения приводит нас к следующему алгоритму симплекс-метода:

I. По знаку коэффициентов при неизвестных в целевой функции f следует определить:

- не достигли ли мы уже оптимального решения (нет отрицательных коэффициентов) и
- если не достигли, то переменную с отрицательным коэффициентом в f из свободной переводим в базисные.

2. Определяем по системе ограничений какую переменную из базисных поменять местами с выбранной свободной; для этого надо выбирать наименьшее положительное отношение элементов столбца свободных членов к коэффициентам при новой свободной переменной.

3. Перерешаем систему относительно нового базиса и вычисляем значение f.

Теперь вернемся к вопросу: как привести систему ограничений к виду (5.4) (в общем случае, в правых частях уравнений будет по k+I слагаемому). Наиболее распространенный метод – метод искусственного базиса, в котором вводятся искусственные, вспомогательные переменные. Для иллюстрации частично разберем пример.

Дана система ограничений

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

Требуется решить ее относительно некоторого исходного базиса.

Прежде всего преобразуем систему так, чтобы свободные члены уравнений стали неотрицательными. Для этого умножим обе части первого уравнения на -1.

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

Введем искусственные неизвестные y_1, y_2 :

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

Все решения исходной системы в примере можно получить, взяв решения новой системы со вспомогательными неизвестными, приняв $y_1=0$, $y_2=0$. В новой системе y_1, y_2 образуют базис. Далее требуется перейти к другому базису, не содержащему ни одной искусственной переменной. Тогда в нем, полагая $y_1=0$ и $y_2=0$, получим систему, равносильную исходной, но уже разрешенную относительно базиса, имеющего вид (5.4). Эта вспомогательная задача решается также симплекс-методом: с его помощью требуется минимизировать вспомогательную функцию

$$F = y_1 + y_2 .$$

Очевидно, что $\min F = 0$, при этом $y_1=0$, $y_2=0$ (ни одна из переменных не может быть отрицательной).

Если выясняется, что $\min F > 0$, то это означает, как легко заметить, что исходная система несовместна. Следовательно, задача линейного программирования с такой системой ограничений неразрешима.

Когда базис выделен, далее задача решается обычным симплекс-методом.

В заключение отметим, что при переходе от одного базиса к другому, система координат, составленная из свободных переменных, трансформируется. Следовательно и ОДР в новой системе координат выглядит иначе (не надо думать, что имеется фиксированный многогранник, по которому "путешествуют" в поисках экстремума, правильнее будет сказать о "путешествии" по проекциям многогранника ОДР).

5.4. Транспортная задача линейного программирования

Имеются m пунктов отправления (ПО) $A_1, A_2, \ldots, A_m$, в которых расположены запасы однородных грузов в количестве соответственно $a_1, a_2, \ldots, a_m$ единиц. Имеются n пунктов назначения (ПН) $B_1, B_2, \ldots, B_n$, подавших заявки соответственно на $b_1, b_2, \ldots, b_n$ единиц груза. Сумма всех заявок равна сумме всех запасов:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j . \qquad (5.7)$$

Известны стоимости c_{ij} перевозки единицы груза от каждого пункта A_i до каждого B_j (задана матрица $\| c_{ij} \|$). Стоимость перевозки нескольких единиц груза пропорциональна их числу (нет оптовых скидок).

Требуется составить план перевозок, чтобы все заявки были выполнены, а стоимость всех перевозок минимальна.

Обозначим x_{ij} – количество груза, отправляемого из A_i в B_j. Совокупность чисел – матрица $\| x_{ij} \|$ будет искомым планом.

Все x_{ij} неотрицательны и удовлетворяют следующим условиям:

$$\sum_{j=1}^{n} x_{ij} = a_i; \qquad i = 1,2,\dots, m - \qquad (5.8)$$

весь отправляемый из A_i пункта груз исчерпывает его запас;

$$\sum_{i=1}^{m} x_{ij} = b_j; \qquad j = 1,2,\dots,n - \qquad (5.9)$$

весь полученный в B_j груз равен заявке.

Оптимальный план минимизирует функцию

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \longrightarrow \min. \qquad (5.10)$$

Мы видим, что перед нами каноническая форма задачи линейного программирования. Однако в силу особой структуры условий - ограничений (все коэффициенты равны единице) можно использовать более рациональный алгоритм поиска решения, чем симплекс-метод. Наиболее известный - метод "северо-западного" угла.

Прежде, чем рассмотреть его, отметим, что число уравнений (ограничений) равно $m+n$ (5.8) и (5.9). Однако наличие условия (5.7) оставляет из них только $m+n-1$ линейно независимых. Общее число переменных x_{ij} равно $m \cdot n$:

$$\| x_{ij} \| = \begin{Vmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{Vmatrix} .$$

Следовательно, число базисных переменных равно $m+n-1$, а число свободных

$$k = mn - (m + n - 1) = (m - 1)(n - 1).$$

Мы знаем, что оптимальное решение нужно искать в опорной точке, где, по крайней мере, k переменных равны нулю. Задачу удобно решать, записав ее в форме таблицы, в клетках которой проставляются перевозки x_{ij} (и справочная стоимость c_{ij}).

Чтобы не менее k переменных (а значит и клеток в таблице) равнялись нулю, очевидно, в заполняемых клетках надо ставить максимально возможное число, поскольку "вес" всех клеток одинаков. Только тогда будет занято минимальное число клеток.

Продемонстрируем метод на таблице 5.I.

Таблица 5.I

ПО \ ПН	B_1	B_2	B_3	B_4	B_5	a_i
A_1	13 (18)	7	14	7	5	30(12)
A_2	11	8	12	6	8	48
A_3	6	10	10	8	11	20
A_4	14	8	10	10	15	30
b_j	18	27	42	15	26	128

Начнем заполнение таблицы с левого верхнего ("северо-западного" угла). Удовлетворяем потребности первого потребителя за счет первого поставщика. Если потребности оказались выше возможностей первого поставщика, то подключаем второго. Если запасы первого поставщика выше потребностей первого потребителя, то остаток запасов передаем второму потребителю и т.д.

Заявка B_1 равна 18 единицам. Удовлетворим ее полностью из запасов A_1, останется запас 30-18=12. Это число запишем в скобках в колонке a_i, как и число 18 в северо-западной клетку. Оставшиеся 12 единиц отдадим B_2. Но ему этого мало, недостающие до 27 единиц, 15 единиц возьмем из A_2 и т.д. Получим таблицу 5.2, где числа оставим без скобок.

Таблица 5.2

ПО\ПН	B_1	B_2	B_3	B_4	B_5	a_i
A_1	13 18	7 12	14	7	5	30
A_2	11	8 15	12 33	6	8	48
A_3	6	10	10 9	8 11	11	20
A_4	14	8	10	10 4	15 26	30
b_j	18	27	42	15	26	128

Пустые клетки предполагаются заполнены нулями. Значения c_{ij} записаны в верхнем углу каждой клетки. Полученный план является опорным, поскольку число свободных переменных $(m-1)(n-1)=12$ равно числу нулевых клеток.

Следующим шагом в решении задачи является проверка опорного плана на оптимальность. Для этого существует метод переноса по циклам (но только по таким, которые имеют отрицательную цену). Цикл - прямоугольный (в общем случае ступенчатый) замкнутый маршрут, начинающийся в свободной угловой клетке, а все остальные угловые - базисные (занятые) клетки. В табл. 5.2 сплошными стрелками отмечен простейший вариант цикла. Ценой цикла называется алгебраическая сумма, получающаяся последовательным чередующимся суммированием и вычитанием стоимостей c_{ij}, участвующих в углах цикла. Для нашего варианта цена равна $6-12+10-8=-4$. Цена показывает насколько изменяется (если цена отрицательная - то уменьшается) стоимость перевозки единицы груза при перераспределении перевозок x_{ij}, стоящих в углах (вершинах) цикла. Например, видно, что если произвести отмеченную стрелками циклическую перестановку перевозок, уменьшив перевозки в "дорогой" клетке (2;3) со стоимостью 12, но зато увеличив перевозки в "дешевой" клетке (2;4) с c_{24}=6, то план будет оптимальнее. Чтобы план оставался опорным, мы должны при этом сделать одну из свободных клеток базисной, а одну из базисных – свободной. Перераспределение должно быть таким, чтобы баланс запасов и заявок не нарушался. Результат циклического переноса представлен в табл. 5.3.

Таблица 5.3

ПО \ ПН	B_1	B_2	B_3	B_4	B_5	a_i
A_1	12 18	7 12	14	7	5	30
A_2	11	8 15	12 22	6 11	8	48
A_3	6	10	10 20	8	11	20
A_4	14	8	10	10 4	15 26	30
b_j	18	27	42	15	26	128

По циклу можно перебрасывать лишь количество груза, равное наименьшей перевозке, стоящей в отрицательной вершине цикла. Например, если бы мы взяли цикл, обозначенный пунктиром, то перебрасывать могли бы только 11 единиц груза. "Отрицательная" вершина – это вершина (клетка) с величиной c_{ij}, которая вычитается при определении цены цикла.

Мы видим, что алгоритм решения задачи похож на симплекс-метод. Тоже базисные и свободные переменные, также замена свободных на базисные и т.д. Но в целом, алгоритм нагляднее и экономичнее: разыскивая в транспортной таблице свободные клетки с отрицательной ценой цикла и перебрасывая по этому циклу по-возможности наибольшее количество груза, мы в конечном итоге придем к оптимальному плану. Для того, чтобы это производить "автоматически", существуют различные способы, в частности, метод потенциалов, но на нем останавливаться не будем. Заметим лишь, что этот метод предполагает решение так называемой двойственной задачи, в которой определенные переменные называются потенциалами. Двойственная задача - тоже задача линейного программирования, но как бы зеркально отраженная по отношению к исходной.

Рассмотрим случай задачи с неправильным балансом, когда заявки не равны запасам. Пусть запасы больше заявок:

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

Задача сводится к транспортной с правильным балансом, если добавить фиктивный пункт B_ϕ, условно приписав ему заявку, равную избытку запасов над заявками

$$b_\phi = \sum_i a_i - \sum_j b_j .$$

При этом все стоимости перевозок $c_{i\phi}$ в фиктивной ПН полагаются нулевыми: $c_{i\phi}=0$. В матрице плана перевозок появляется дополнительный столбец $x_{i\phi}$, $i=1,2,\dots,m$, и задача сводится к обычной. Только нужно помнить, что все получающиеся перевозки $x_{i\phi}$ на самом деле не производятся, а остаются в ПО A_i.

Если запасов не хватает:

$$\sum_i a_i < \sum_j b_j ,$$

то можно ввести фиктивный ПО A_ϕ, условно приписав ему недостающий запас. Дальше задача решается аналогично рассмотренной выше. Естественно, часть потребителей недополучит груза на величину фиктивного запаса $x_{\phi j}$.

Достоинством алгоритма решения транспортной задачи является то, что при целочисленных ограничениях на запасы и заявки, решение получается тоже целочисленным.

5.5. Примеры задач, решаемых симплекс-методом. Задача о наилучшем использовании производственных площадей

Пусть имеется m производственных участков площадью S_1, $S_2, \ldots, S_m$ под выполнение n видов работ. Производительность на i-м участке j-й работы соответственно равна a_{ij}, $j=1,2,\ldots,n$ (участки могут быть необорудованными, закрытыми, отапливаемыми и т.п.). Доход от работ соответственно составляет c_j.

Требуется определить какую площадь на каждом участке следует отдать под каждый вид работ, чтобы получить максимальную прибыль, если по договорам должно быть выполнено не менее b_j работ каждого вида.

Обозначим через x_{ij} площадь, которую предполагается отвести на i-м участке под j-й вид работ, так что

$$x_{i1} + x_{i2} + \ldots + x_{in} = S_i.$$

Ожидаемый объем j-го вида работ

$$a_{1j}x_{1j} + a_{2j}x_{2j} + \ldots + a_{mj}x_{mj} \geq b_j$$

по условиям договоров.

Ожидаемая прибыль от j-го вида работ равна

$$c_j(a_{1j}x_{1j} + a_{2j}x_{2j} + \ldots + a_{mj}x_{mj}),$$

а от всех работ определяется суммой

$$\sum_{j=1}^{n} c_j\,(a_{1j}x_{1j} + a_{2j}x_{2j} + \ldots + a_{mj}x_{mj}).$$

Таким образом, задача заключается в максимизации

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

от $m \times n$ переменных x_{ij} при выполнении следующих ограничений:

$$\sum_{j=1}^{n} x_{ij} = S_i ;$$

$$\sum_{i=1}^{m} a_{ij} x_{ij} \geq b_j ;$$

$$x_{ij} \geq 0 .$$

Задача решается симплекс-методом.

5.8. Целочисленное линейное программирование

Во многих практических задачах требуется, чтобы искомые переменные x_j $(j=1,\dots,n)$ были целыми. Простое округление нецелочисленных решений может привести к далеко не оптимальным результатам. Например, купить 2, а не I самолет и т.д. Существует большое число задач, особенно комбинаторных, которые можно сформулировать как задачи целочисленного программирования. Методы их решения для различного вида задач различны. Используются как методы линейного программирования, так и сетевые (потоковые) модели.

При решении хотелось бы знать, имеет ли задача целочисленное оптимальное решение при (любых) целочисленных ограничениях b_i. Это означает, что выпуклый многогранник ОДР, определяемый ограничениями

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

$$x_j \ge 0 \ , \ \ j\in\{1,\ n\}\ ;$$

$$x_j,\ b_i \ - \text{все целые},$$

имеет целочисленные крайние (угловые) точки (рис. 5.3). А это бывает далеко не всегда. На рисунке угловые точки B и C не являются даже допустимыми решениями, поскольку ни одна из них не целочисленна.

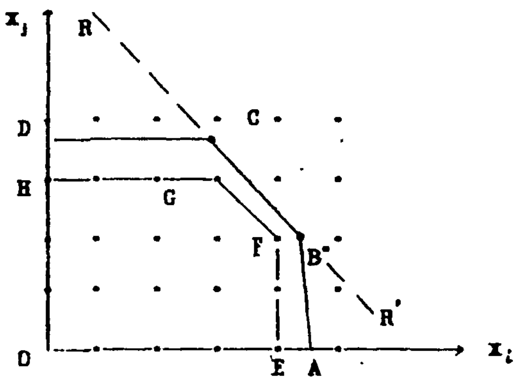


Рис. 5.3

Для существования целочисленного оптимального решения требуется абсолютная унимодулярность матрицы A:

$$A = \left\| \begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right\|_{m\times n} .$$

Матрицу A называют абсолютно унимодулярной, если все ее миноры равны 0 либо ± I. (Минор k-го порядка – это определитель, построенный из элементов, стоящих на пересечении произвольных k строк и k столбцов). На практике для ответа на вопрос о существовании решения используются специальные методы. Но даже ответив на вопрос положительно, получить оптимальное решение не просто. Решение задачи, обычно, сводится к некоторой модифицированной задаче линейного программирования, в которой добавляются ограничения типа RR', как показано на рис. 5.3. В результате исходная ОДР OABCD сокращается до OEFGH такой, в которой все крайние точки целочисленны и при этом область содержит все допустимые целочисленные точки исходной области.

Для различного типа задач разработаны специальные методы и вычислительные программы. Мы разберем лишь постановку некоторых задач.

Пусть имеется ряд неделимых предметов (контейнеров) $П_1, П_2, \ldots, П_n$, которые желательно увезти некоторым авиарейсом. Известны стоимости этих предметов: $c_1, c_2, \ldots, c_n$ и их вес $g_1, g_2, \ldots, g_n$. Перевозка лимитируется грузоподъемностью самолета Q, а все предметы в единственном числе. Задача заключается в выборе наиболее выгодного набора предметов, вес которого укладывается в Q.

Введем переменные $x_1, x_2, \ldots, x_n$, определяемые условием

$$x_i = \begin{cases} I, \text{ если выбирается предмет } П_i; \\ 0, \text{ если не выбирается;} \end{cases}$$
$$i = I, 2, \ldots, n.$$

Ограничение задачи укладывается в одном неравенстве

$$g_1x_1 + g_2x_2 + \ldots + g_nx_n \le Q.$$

Показателем эффективности является функция

$$f = \sum_{i=1}^{n} c_i x_i \longrightarrow \max,$$

На первый взгляд может показаться, что задача решается методом линейного программирования при дополнительных ограничениях $x_i \leq 1$, $i = 1, 2, \ldots, n$.

Однако решение наверняка получится "дробным". Если его округлить, то либо груз "не влезет" в Q, либо решение будет далеко не оптимальным.

Для данного типа задач (задачи о назначениях) разработаны специальные методы решения. В дальнейшем мы эту задачу решим методом динамического программирования.

5.7. Задача о назначениях

Пусть требуется распределить экипажи самолетов по рейсам так, чтобы затраты были минимальными. Ограничение заключается в том, чтобы каждый экипаж выполнял только один рейс, а все рейсы выполнялись по одному разу.

Решение задачи записывается в двумерный массив $\bar{X} = \|x_{ij}\|$, $i,j = 1,2,\ldots,m$, где m – число экипажей, равное числу рейсов. Искомая переменная

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-й экипаж назначается на } j\text{-й рейс;} \\ 0 & \text{в противном случае.} \end{cases}$$

Критерий эффективности включает в себя элементы матрицы стоимостей c_{ij} – затраты, связанные с i-м назначением на j-й рейс.

Для заданного значения m существует $m!$ допустимых решений. Необходимо минимизировать

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

при ограничениях:

а) каждый экипаж выполняет только один полет -

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad \text{для всех } i;$$

б) каждый рейс (маршрут) выполняется один раз -

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \text{для всех } j.$$

Очевидно, что задача о назначениях является частным случаем транспортной задачи при единичных значениях параметров a_i и b_j. Поэтому для решения задачи о назначениях можно воспользоваться методом потенциалов транспортной задачи. Разработан также и специальный алгоритм, называемый венгерским [3]. Мы его рассматривать не будем.

5.8. Задача о закреплении самолетов за воздушными линиями

Эта задача возникает при выборе оптимального варианта плана закрепления самолетов за данными воздушными линиями. План должен обеспечивать необходимые объемы перевозок при минимальных суммарных эксплуатационных расходах.

Пусть имеется n типов самолетов, которые нужно распределить между m авиалиниями. Месячный объем перевозок самолетом i-го типа на j-й авиалинии равен a_{ij} единицам, а связанные с этим месячные эксплуатационные расходы составляют c_{ij}. Задано также число N_i самолетов i-го типа.

Требуется определить число x_{ij} самолетов i-го типа, которые следует закрепить за j-й авиалинией для обеспечения перевозки по этой линии b_j единиц груза.

Объем перевозок по j-й линии составляет

$$a_{1j}x_{1j} + a_{2j}x_{2j} + \dots + a_{nj}x_{nj}.$$

Следовательно, ограничения задачи запишутся в виде

$$\sum_{i=1}^{n} a_{ij}x_{ij} \ge b_j, \quad j=1,2,\dots,m;$$

$$\sum_{j=1}^{m} x_{ij} = N_i, \quad i=1,2,\dots,n.$$

Минимизируемая функция имеет вид

$$f_{(x)} = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}\, x_{ij};$$

все x_{ij} – неотрицательные целые числа.

6. СЕТЕВЫЕ (ПОТОКОВЫЕ) ЗАДАЧИ

6.I. Основные определения и приложения потоковых моделей

Многие задачи линейного программирования могут быть сформулированы как сетевые. Примером таких задач служит транспортная, которую графически сокращенно можно изобразить в виде сети, с источниками S_i и стоками d_j (рис. 6.I). Однако имеются и такие сетевые задачи, которые нельзя сформулировать в виде задачи линейного программирования. Например, таковой является задача коммивояжера.

Вследствие специальной структуры сетевых задач для них получено большое число эффективных алгоритмов и изящных теорем, обеспечивающих решение широкого круга практических задач. В отличие от общей задачи линейного программирования в большинстве сетевых задач оптимальные решения целочисленны.

Введем основные определения и понятия.

Сеть состоит из множества узлов (вершин) и множества дуг (ребер), которые связывают эти узлы. Яркий пример с полезными аналогиями – электрическая сеть. Узлы характеризуют состояния в условии задачи, а дуги описывают процессы, протекающие при переходе из состояния в состояние. Узлам N_j приписывают числа $\bar{x}_j$, которые в свою очередь определяются через переменные величины x_{ij} условий задачи.

Последовательность узлов N и дуг A сети: N_1, A_{12}, N_2 A_{23}, ..., N_{k-1}, $A_{k-1}k$, N_k называют цепью. В сети выделяют два специальных узла: источник N_s и сток N_t.

Потоком из источника N_s в сток N_t в сети называется величина, определяемая множеством неотрицательных чисел x_{ij}, поставленных в соответствие дугам A_{ij}:

$$\bar{x}_j = \sum_i x_{ij} - \sum_k x_{jk} = \begin{cases} -V, \text{ если } j = S\,; \\ 0, \text{ если } j \neq S,\ j \neq t\,; \\ +V, \text{ если } j = t\,; \end{cases} \qquad (6.I)$$

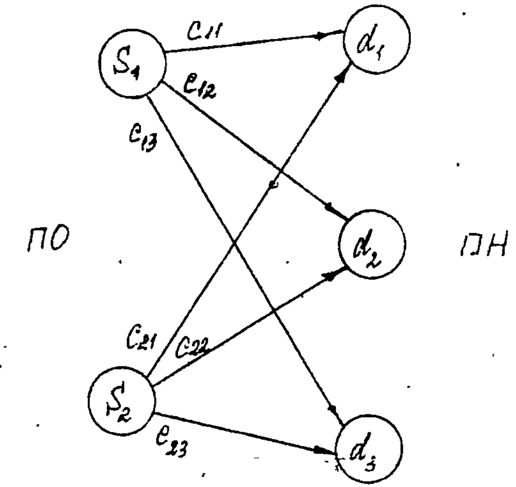


Рис. 6.I

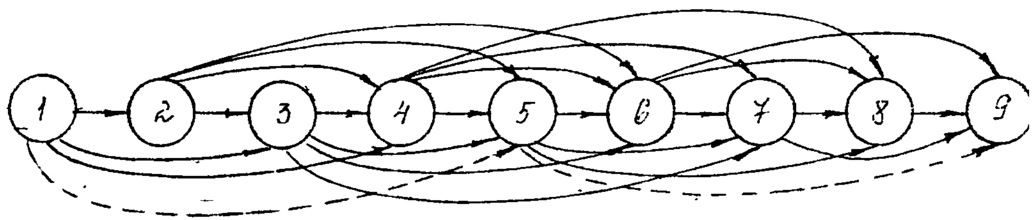


Рис. 6.2

$$V \geq 0;\ 0 \leq x_{ij} \leq b_{ij} \quad \text{для всех } i,\ j,\ k. \qquad (6.2)$$

Здесь первая сумма берется по дугам, ведущим в узел N_j, а вторая сумма – из узла N_j. Неотрицательное число V называется величиной потока. Ограничения (6.I) выражают тот факт, что в каждый узел j (кроме источника и стока) какой поток поступает, такой и уходит (условие сохранения потока). Условие (6.2) ограничивает пропускную способность дуги величиной b_{ij}.

Очевидно, что задача нахождения максимального потока в такой сети является задачей линейного программирования: максимизировать $f=\sum_{i} x_{i1}$ при ограничениях (6.I)-(6.2). Легко заметить что $f_{max} = V = \sum x_{i1}$. Но в силу специфики задачи симплекс-метод оказывается не самым рациональным.

В практических приложениях потоковых моделей часто встречаются задачи, которые могут быть сформулированы в виде задач о кратчайшей цепи, о потоке минимальной стоимости, о максимальном потоке и т.п. Например, задача о наименьшей стоимости заключается в следующем. Заданы множества дуг и узлов. Каждой дуге A_{ij} или (i,j) поставлена стоимость c_{ij} единицы потока по этой дуге. Требуется найти цепь из N_s в N_t, минимизирующую стоимость единицы потока из источника в сток. Примером подобной задачи служит задача о замене устаревшего оборудования (транспортного средства) на новое. В ней минимизируются общие затраты на покупку и обслуживание оборудования, если ликвидационная стоимость при различных сроках службы оборудования, а также эксплуатационные расходы за каждый период известны. Обычно величины задаются по годам на планируемый срок.

6.2. Задача о покупке автомобиля (самолета)

Задача заключается в определении оптимальных сроков замены старого автомобиля на новый и служит примером задачи о наикратчайшей цепи или цепи наименьшей стоимости. На рис. 6.2 показана сеть задачи при условиях предыдущего параграфа, в которой для определенности принято, что замена автомобиля должна совершаться, по крайней мере, каждые 4 года за период $T=8$ лет. Пунктиром обозначена возможная оптимальная цепь.

Четыре дуги, выходящие из узла I, соответствуют четырем возможностям по условию приобретения автомобиля на 2, 3, 4 или 5 лет. Каждому году соответствует узел. Если автомобиль куплен в i-м году, а заменен в j-м, то такому варианту соответствует дуга (i,j). Затраты на дуге (i,j) складываются

$$c_{ij} = p_i - s_j + \sum_{k=1}^{j-1} m_k , \quad j - i \le 4 ,$$

где p_i - стоимость автомобиля в i-м году;

s_j - ликвидационная его стоимость в j-м году;

m_k - эксплуатационные расходы.

Оптимальному решению поставим в соответствие цепь, минимизирующую стоимость единицы потока из узла I в узел 9, при известных стоимостях c_{ij}. Пример такой возможной цепи дан пунктиром на рис. 8.2.

Введем искомые переменные:

$$x_{ij} = \begin{cases} I, \text{ если автомобиль куплен в i-м году,} \\ \quad \text{а заменен в j-м;} \\ 0, \text{ во всех остальных случаях.} \end{cases}$$

Так для пунктирной цепи $x_{15} = I$ и $x_{59} = I$, остальные $x_{ij} = 0$.

Условия задачи запишутся

$$\sum_j x_{sj} - \sum_j x_{js} = 1 ;$$

$$\sum_j x_{ij} - \sum_j x_{ji} = 0 , \quad \text{если } i \ne s, \ i \ne t ;$$

$$\sum_{j} x_{tj} - \sum_{j} x_{jt} = -1 .$$

Требуется минимизировать

$$f = \sum_{i} \sum_{j} c_{ij} x_{ij} ;$$

$$c_{ij} \geq 0 .$$

Согласно первому равенству единица потока вытекает из источника s, а согласно третьему – втекает в сток t. Второе равенство гарантирует сохранение потока. Таким образом, получена задача линейного программирования с неизвестными x_{ij}. В качестве цепи наименьшей стоимости берется последовательность дуг (i,j) с минимальным значением f, для которых $x_{ij} = 1$.

Очевидно, что решение должно быть целочисленным. Понятно, что при дробных числах x_{ij} маршрут по сети становится неопределенным и решение не имеет смысла.

Для приведенного типа потоковых задач наименьшей стоимости имеются специальные алгоритмы, например Дейкстры [3].

6.3. Задача коммивояжёра

Предположим, что туристская фирма планирует некоторый маршрут, включающий n пунктов для посещения. Маршрут должен быть таким, чтобы затраты на путешествие были минимальными и чтобы выполнялось единственное требование: каждый пункт посещался только один раз. Полная информация о стоимости пролета самолетом из одного города (пункта) в другой имеется и может быть для примера представлена табл. 6.1 значений c_{ij}.

Таблица 6.I

№ пп	Пункты	Исходный пункт	Токио	Гонконг	Лондон	Сидней	Рим
		I	2	3	4	5	6
I	Исходный пункт	–	280	440	I70	3I0	270
2	Токио	80	–	I70	20	3I0	3I0
3	Гонконг	2I0	I40	–	360	60	I0
4	Лондон	220	I70	260	–	I90	I90
5	Сидней	I30	470	280	490	–	60
6	Рим	240	60	60	I00	80	–

Матрица стоимостей из-за льгот авиакомпаний несимметрична.

Подобные задачи имеют давнюю историю, относящуюся к задаче о Кенигсбергских мостах. В ней требовалось кратчайшим образом обойти все мосты, не пересекая ни один из них дважды. Такие задачи получили название задачи коммивояжёра. Они формулируются следующим образом. Коммивояжёр должен выехать из исходного пункта, побывать в каждом из остальных (n-I) пунктов ровно один раз и вернуться назад. Требуется определить последовательность объезда пунктов, для которой принятый критерий эффективности будет минимальным. В качестве критерия может выступать стоимость проезда, время в пути, суммарное расстояние и т.д. В задаче требуется выбрать один из $(n-1)!$ маршрутов.

Задача коммивояжёра не может быть непосредственно сформулирована и решена как задача линейного программирования. Основная особенность задачи заключается в том, что в ней требуется построение ориентированного цикла.

Эффективным алгоритмом решения является метод ветвей и границ, основанный на сетевой, топологической модели задачи. Маршрут T представляется множеством упорядоченных пар пунктов - дуг (i,j):

$$T = ((i_1, i_2); (i_2, i_3); \ldots; (i_{n-1}, i_n), (i_n, i_1)) .$$

Каждый допустимый маршрут представляет собой цикл, проходя по которому коммивояжёр посещает каждый пункт ровно один раз и возвращается в исходную точку. Стоимость $C(T)$ маршрута T равна сумме только тех c_{ij}, которые принадлежат T:

$$C(T) = \sum_{(i,j)\in T} c_{ij} .$$

Величина $C(T)$ любого допустимого маршрута является верхней границей стоимости оптимального маршрута T^*, который требуется найти. Его поиск выражается итерационной процедурой, например [3], в которой последовательно выбираются маршруты со все меньшим значением $C(T)$. Для таблицы 6.I оптимальный маршрут: исходный пункт – Лондон – Гонконг – Сидней – Рим – Токио – исходный пункт; его стоимость равна 690 единиц. Желающий, пользуясь таблицей 6.I, может сам убедиться в этом, попытавшись самостоятельно "откопать" более дешевый путь. Автор только предостерегает туристов воспользоваться приведенным решением на практике (все числа в примере вымышленные).

Может показаться, что линейные модели в задачах оптимизации должны занять господствующее положение: они выглядят универсальными и для них развит математический аппарат. Если же в условиях задачи встретится какая-либо нелинейность, то её можно будет исправить соответствующей аппроксимацией, ведь в конце концов любая модель – это лишь приближение к реальности. Однако во многих, в том числе принципиальных случаях, линейные модели не применимы.

Ярким и наглядным примером к сказанному является электрическая сеть. Возьмем простейший случай с тремя узлами и тремя дугами (рис. 6.3):

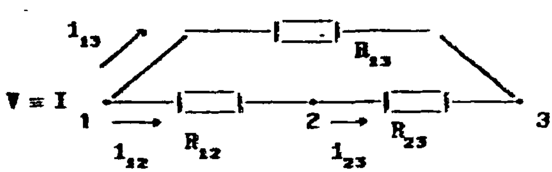


Рис. 6.3

Величиной потока V, очевидно, является ток I, вытекающий из источника-узла I. Применение законов Кирхгофа (здесь просто Ома) даёт решение для искомых токов в дугах с сопротивлениями R_{jk}.

$$i_{12} = \frac{I\,R_{13}}{R_{13} + R_{12} + R_{23}}\;;\quad i_{13} = I - i_{12},\quad i_{23} = i_{12}\;.$$

Потоковые уравнения сетевой модели (6.I) приводят к:

$$\begin{cases} I = i_{12} + i_{13}\;; \\ i_{12} - i_{23} = 0 \end{cases} \qquad (6.3)$$

с искомыми неизвестными $i_{jk} = x_{jk}$.

Примем в качестве целевой функции следующее нелинейное выражение:

$$f = i_{13}^2 R_{13} + i_{12}^2 R_{12} + i_{23}^2 R_{23}.$$

Роль стоимостных коэффициентов в ней играют сопротивления в сети, а сама функция выражает рассеиваемую мощность.

Для минимизации f нет необходимости прибегать к численным методам, задачу можно легко решить "в лоб", аналитически (подробнее о прямых методах оптимизации см. п. 9.I).

Из (6.3) получаем $\begin{cases} i_{13} = I - i_{12}; \\ i_{23} = i_{12}\;. \end{cases}$

Подставляя в f, получаем

$$f = (I - i_{12})^2 R_{13} + i_{12}^2 R_{12} + i_{12}^2 R_{23} \longrightarrow \min.$$

Необходимое условие существования экстремума функции приводит к уравнению:

$$\frac{df}{di_{12}} = -2IR_{13} + 2i_{12}(R_{13} + R_{12} + R_{23}) = 0 .$$

Окончательно, $i_{12} = \dfrac{I\,R_{13}}{R_{13} + R_{12} + R_{23}}$, что совпадает с решением по формуле Ома.

Таким образом заключаем, что согласующейся с опытом является квадратичная целевая функция, приводящая задачу к классу нелинейных. Попутно отметим другой важный вывод. Физическая задача, сформулированная без каких-либо упоминаний об оптимальности, может быть решена на базе двух принципиально различных подходов. Первом, причинном: токи обусловлены приложенным напряжением и зависят от сопротивления цепи. И втором, основанном на выполнении физического принципа, провозглашающего минимум рассеиваемой энергии. Принцип, в общем случае, может выражать некоторую гармонию, симметрию.

Приведенный пример не единственный. В аналитической механике принцип наименьшего действия позволяет решать задачи динамики не прибегая к уравнениям Ньютона (роль целевой функции там играет функция Гамильтона). Мы столкнемся ещё с вопросом о связи оптимизационных и неоптимизационных задач в п. 7.3, где речь пойдет о теореме двойственности в линейном программировании.

7. ВВЕДЕНИЕ В ИГРОВЫЕ МЕТОДЫ ПРИНЯТИЯ РЕШЕНИЙ

7.1. Матричные игры как модели конкурентных конфликтных ситуаций

В реальной деятельности при планировании различных операций необходимо учитывать возможные реакции на производимые действия. Реакции возникают в ситуациях, в которых сталкиваются интересы двух (или более) сторон, преследующих разные цели, причем выигрыш каждой стороны зависит от того, как себя поведут другие. Подобные ситуации называются конфликтными. Самый простой пример: транспортная компания – пассажир; первый хочет заработать побольше, второй – заплатить поменьше.

Теория игр представляет собой математическую теорию конфликтных ситуаций. От реального конфликта игра отличается тем, что ведется по определенным правилам. Развитие игры во времени можно представить как ряд последовательных ходов участников. Ходом называется выбор игроком (стороной) одного из предусмотренных правилами игры действий и его осуществление. Ходы бывают личные и случайные. При личном ходе игрок сознательно выбирает тот или другой вариант действий (пример – ход в шахматах). При случайном ходе выбор случаен (бросание игральной кости).

В игре каждый участник стремится максимизировать свой выигрыш. Для этого он ищет оптимальную стратегию – поведение, определяющее выбор варианта действий, из предусмотренных правилам игры при каждом личном ходе в зависимости от сложившейся ситуации. Основное предположение, исходя из которого математически находятся оптимальные стратегии, состоит в том, что противник так же "разумен", как и сам игрок, т.е. задача симметрична относительно игроков.

Игра называется игрой с нулевой суммой, если сумма выигрышей всех игроков равна нулю. Если это парная игра, то она называется еще и антагонистической. Антагонистические игры наиболее разработанные в теории игр.

Примеры антагонистических игр хорошо знакомы из практики и не только игровой. Упомянутая ранее транспортная компания, надеясь заработать побольше, подняла тарифы за проезд. Пассажиры ответили безбилетным проездом. Компания расширяет штат контролеров (несет расходы). Пассажиры меньше ездят и т.д.

Рассмотрим игру G с двумя участниками A и B, имеющими противоположные интересы: выигрыш одного равен проигрышу другого. Так как выигрыш игрока A равен взятому с обратным знаком выигрышу игрока B, мы можем интересоваться только выигрышем a игрока A. Естественно, A хочет максимизировать a, B – минимизировать a.

Пусть по правилам игры имеется m возможных стратегий $A_1, A_2, \ldots, A_m$ у игрока A и n – у B, т.е. $B_1, B_2, \ldots, B_n$ (обозначается $m \times n$). Обозначим выигрыш a_{ij} игрока A в случае, если им использовалась стратегия A_i, а противником – B_j. Если все A_i, B_j и a_{ij} известны, то они могут быть записаны в табличном виде, а сама игра G считается приведенной к матричной форме.

Рассмотрим пример игры (4×5), в которой правилами возможные стратегии транспортной компании A и пользователя B определены следующие: Стратегии A (A_1 – увеличить тариф за перевозки; A_2 – сохранить тариф; A_3 – ввести дополнительное страхование; A_4 – уменьшить тариф), Стратегии B (B_1 – уменьшить объем перевозок; B_2 – использовать льготный тариф на оптовые перевозки и т.д). Матрица игры дана в табл. 7.1.

Таблица 7.1

A_i \ B_j	B_1	B_2	B_3	B_4	B_5	α_i
A_1	3	4	5	2	3	2
A_2	1	8	4	3	4	1
A_3	10	3	1	7	6	1
A_4	4	5	3	4	8	|3|
β_j	10	8	|5|	7	8	

Величины a_{ij} определяют доход компании для пары (A_i, B_j). Соответственно эту сумму оплачивает пользователь, ведь по условию – это игра с нулевой суммой.

Из таблицы следует: для игрока A на первый взгляд предпочтительней выглядит стратегия A_3, ведь именно она может принести максимальный выигрыш a_{31} = 10. В жизни может так бы и произошло, но в теории игр считается, что и B "не промах", и на A_3 он ответит своей стратегией B_3, чем сведет предполагаемый выигрыш до единицы (a_{33}= 1). Как же быть? Исходя из основного принципа теории, заключающегося в одинаковой разумности игроков, надо выбирать ту стратегию, при которой наш минимальный выигрыш максимален. Это – так называемый принцип минимакса : для каждой из стратегий выбирается минимальный выигрыш, затем выбирается та стратегия, у которой этот минимальный выигрыш максимален. Минимальные выигрыши для стратегий $A_1 \div A_4$ записаны в столбце α_i, а максимальный из них выделен рамочкой. Ему соответствует стратегия A_4, гарантирующая при любом поведении B выигрыш не меньше, чем 3. Этот гарантируемый выигрыш называется нижней ценой игры или максимином α, в нашем случае α = 3.

Теперь станем на точку зрения противника B. Его задачей является из наихудших для него стратегий A_i выбрать свою B_j такую, чтобы проигрыш был минимальным (такое уж условие игры – "куда не кинь, всюду клин"). Например, на B_1 ответом будет A_3 и проигрыш a_{31} = 10; на B_2 ответ A_2 с проигрышем 8 (прямо как в шахматах разбор этюда). Для выбора наименьшего из вол составим строчку, в которую запишем максимумы столбцов. Минимальное значение из них также выделено рамочкой и обозначает верхнюю цену игры β или минимакс. Для оптимальной стратегии игрока B (т.е. B_3) он большую сумму, чем β = 5, не проиграет.

Таким образом, для игры, описанной таблицей 7.1, исходя из принципа одинаковой "разумности" или "осторожности" ("всегда рассчитывай на худшее"), игрок A выбирает стратегию A_4, а противник – B_3. Такие стратегии называются минимаксными. До тех пор, пока игроки придерживаются своих минимаксных стратегий, выигрыш a_{43} = 3.

Теперь допустим, что игрок A узнал о том, что его противник постоянно выбирает B_3. Тогда есть смысл для A выбрать A_1 с выигрышем $a_{13} = 5$. Но в ответ на A_1 противник очевидно выберет стратегию B_4, чтобы свести проигрыш до 2 ($a_{14}=2$) и т.д. (партнеры "заметались по стратегиям"). Мы имеем пример минимаксных стратегий, неустойчивых к информации о поведении другой стороны. Но бывают и устойчивые стратегии.

Рассмотрим пример таблицы 7.2, в котором нижняя цена игры равна верхней

$$\alpha = \beta.$$

Таблица 7.2

$A_i \backslash B_j$	B_1	B_2	B_3	B_4	α_i
A_1	2	4	7	5	2
A_2	7	|6|	8	7	|6|
A_3	5	3	4	1	1
β_j	7	|6|	8	7	

Из таблицы видно, что $\alpha = \beta = 6$. Это значит, что гарантированный выигрыш A равен наименьшему проигрышу при самых неблагоприятных для B случаев. Из этого следует, что как для A, так и для B невыгодно отступать от своих стратегий A_2 и B_2 даже в случае, когда они располагают информацией о намерениях своих противников. Любой отход игрока A от стратегии A_2 только ухудшит его положение (будет меньше α_i, ведь противник "не глуп"). Равным образом, информация, полученная B, не заставит его изменить свою стратегию B_2. Пара стратегий (A_2, B_2) обладает свойством равновесия, а выигрыш (здесь 6), достигаемый при равновесии, называется "седловой точкой". Свое название она получила из геометрии, где обозначает точку, в которой некоторая поверхность достигает минимума по одной координате и максимума по другой. Признак наличия седловой точки - это равенство нижней и верхней цены игры (теперь это просто цена $\alpha = \beta = \nu$). Стратегии, при которых достигаются седловые точки (здесь это A_2, B_2), называются оптимальными чистыми стратегиями, а их совокупность - решением игры. В таких случаях говорят, что решение принимается в условиях определенности.

Сразу отметим, что игры, решаемые (возможно только в принципе) в чистых стратегиях, - это почти всегда только "настоящие" игры (шашки, шахматы, "крестики нолики" и т.п.). Рассмотренный пример транспортной компании лишь грубое приближение: если можно представить себе транспортников, постоянно увеличивающих плату за проезд, то это трудно сделать в отношении пассажира, все чаще вынужденного ездить "зайцем". Игра предполагает неизменными условия, в которых многократно применяются правила, к тому же четко оговоренные. Тем не менее, даже идеализированные модели реальных ситуаций иногда позволяют получать практические рекомендации и, уж конечно, помогают лучше разобраться в проблеме.

Для решения игры в чистых стратегиях главное - это привести её к матричной форме (для шахмат - это невозможно пока). Дальше игра фактически сводится к единственому ходу - выбрать оптимальную стратегию. Гораздо интереснее задачи (с точки зрения практики, а не игры), в которых поведение сторон подвержено случайным воздействиям. В таких задачах выбор стратегии диктуется не только разумом игроков, но и влиянием "форс мажор". Каждая стратегия может появиться со своей вероятностью, а в сумме, говорят, это приводит к смешанным стратегиям. Смешанные стратегии игроков A и B обозначаются

$$S_A = (p_1, p_2, \dots, p_m);$$

$$S_B = (g_1, g_2, \dots, g_n),$$

где $p_1, p_2, \dots, p_m$ - вероятности применения стратегий $A_1, A_2, \dots, A_m$;

$g_1, g_2, \dots, g_n$ - вероятности стратегий $B_1, B_2, \dots, B_n$.
Вероятности p и g нормированы, т.е.

$$\sum_{i=1}^{m} p_i = 1 \; ; \qquad \sum_{j=1}^{n} g_j = 1 .$$

В частном случае, когда все *вероятности, кроме одной*, равны нулю, а эта одна – единице, смешанная стратегия превращается в чистую.

Существует основная теорема теории игр: любая конечная игра двух лиц с нулевой суммой имеет, по крайней мере, одно решение – пару оптимальных, в общем случае смешанных, стратегий (S_A^*, S_B^*) и соответствующую цену ν.

В конечных играх число стратегий ограничено. Цена игры, поскольку присутствуют вероятности, определяется как математическое ожидание выигрыша при оптимальной стратегии.

7.2. Методы решения конечных игр

Пусть имеется игра $m \times n$ без седловой точки с матрицей $\|a_{ij}\|$. Отсутствие седловой точки означает, что задача не решается в чистых стратегиях, поэтому будем искать смешанные стратегии S_A^* и S_B^*.

Допустим, что все выигрыши a_{ij} положительны (этого всегда можно добиться, прибавляя ко всем элементам матрицы положительное число M, большее по нулю любого из отрицательных её элементов; от этого цена игры увеличится на M, а решение S_A^*, S_B^* не изменится). Если все $a_{ij} > 0$, то цена игры, т.е. средний выигрыш при оптимальной стратегии, тоже положительна: $\nu > 0$.

Решение игры ищется в виде двух оптимальных смешанных стратегий

$$S_A^* (p_1, p_2, \ldots, p_m) \; ; \quad S_B^* (g_1, g_2, \ldots, g_n) \; ,$$

дающих каждой стороне гарантированный успех. Игроку А – максимальный выигрыш, игроку В – минимальный проигрыш при наихудших действиях противной стороны.

Найдем сначала S_A^*. Мы знаем, что если один из игроков (в данном случае это А) применяет свою оптимальную стратегию, то другой игрок (В) не может улучшить свое положение, отступая от своей. Если же он отступит, то выигрыш А возрастет (точнее, не уменьшится). Это условие мы и запишем в следующем виде. Пусть В пользуется чистыми стратегиями, сначала B_1, затем $B_2, \dots, B_n$ (игрок А по-прежнему держится своей стратегии S_A^*). Тогда при любой B_j выигрыш у А будет не меньше, чем ν:

$$\begin{cases} a_{11} p_1 + a_{21} p_2 + \dots + a_{m1} p_m \geq \nu, \text{ для } B_1 ; \\ a_{12} p_1 + a_{22} p_2 + \dots + a_{m2} p_m \geq \nu, \text{ для } B_2 ; \\ \dots \dots \dots \dots \\ a_{1n} p_1 + a_{2n} p_2 + \dots + a_{mn} p_m \geq \nu, \text{ для } B_n . \end{cases} \tag{7.1}$$

В левых частях неравенств стоят математические ожидания, т.е. средний выигрыш А в случае применения им оптимальной стратегии S_A^* при неизменной стратегии B_j. Разделим неравенства (7.1) на положительную величину ν и введем обозначения

$$x_1 = \frac{p_1}{\nu}, \quad x_2 = \frac{p_2}{\nu}, \quad \dots, \quad x_m = \frac{p_m}{\nu} .$$

Тогда условия (7.1) примут вид

$$\begin{cases} a_{11} x_1 + a_{21} x_2 + \dots + a_{m1} x_m \geq 1 ; \\ a_{12} x_1 + a_{22} x_2 + \dots + a_{m2} x_m \geq 1 ; \\ \dots \dots \dots \dots \\ a_{1n} x_1 + a_{2n} x_2 + \dots + a_{mn} x_m \geq 1 , \end{cases} \tag{7.2}$$

где $x_1, x_2, \dots, x_m$ – неотрицательные переменные.

В силу условия нормировки вероятностей $p_1+p_2+\dots+p_m = 1$

переменные удовлетворяют условию

$$x_1 + x_2 + \ldots + x_m = \frac{1}{\nu} . \qquad (7.3)$$

Но ν есть гарантированный выигрыш игрока A, который он стремится сделать максимальным, а значит, величину $1/\nu$ – минимальной.

Таким образом, игровая задача свелась к следующей математической. Найти неотрицательные значения переменных x_1, $x_2, \ldots, x_m$ такие, чтобы они удовлетворяли линейным ограничениям – неравенствам (7.2) и обращали в минимум линейную функцию этих переменных:

$$f(x) = x_1 + x_2 + \ldots + x_m \longrightarrow \min . \qquad (7.4)$$

В результате мы пришли к задаче линейного программирования с n ограничениями и m переменными. Определив все x^* и функцию $f(x^*) = 1/\nu$, мы без труда находим искомые вероятности p^*, т.е. оптимальную стратегию S^*_A.

Оптимальная стратегия игрока B находится аналогично, с той разницей, что B стремится минимизировать, а не максимизировать выигрыш. Он будет обращать в максимум величину $1/\nu$, а в ограничениях-неравенствах вместо знаков "$\geq$" будут стоять "$\leq$". Пара задач линейного программирования, по которой находятся оптимальные стратегии (S^*_A, S^*_B), называется парой двойственных задач линейного программирования.

В качестве примера приведем игру с матричной формой табл. 7.3.

Таблица 7.3

$A_i \diagdown B_j$	B_1	B_2	B_3	α_i
A_1	1	-3	4	$\|-3\|$
A_2	-3	4	-5	-5
A_3	4	-5	6	-5
β_j	$\|4\|$	$\|4\|$	6	

Матрица описывает игру: два игрока A и B одновременно называют по числу из трех возможных – I либо 2, либо 3. Выигрыш решает общая сумма названных чисел: если она – четное число, выигрывает A и получает её у B; если – нечетное, то наоборот. В табл. 7.3 индекс у соответствующей стратегии равен называемому числу, поэтому, например a_{13} = I+3=4 – число четное, и выигрывает A (значит a_{13} – число положительное). Поскольку верхняя и нижняя цена игры не совпадают, то игра в чистых стратегиях не решается. Решение ищется в паре двойственных задач и приводится к двум смешанным стратегиям:

$$S^*_A\ (1/4,\ 1/2,\ 1/4) \text{ и } S^*_B\ (1/4,\ 1/2,\ 1/4).$$

Непосредственной подстановкой убеждаемся, что цена игры ν, т.е. средний выигрыш для всех A_i и B_j равен нулю. Этот ответ полностью согласуется со "здравым смыслом" вследствие симметрии задачи. Например, для A_1

$$2 \cdot \frac{1}{4} - 3 \cdot \frac{1}{2} + 4 \cdot \frac{1}{4} = 0 .$$

Смысл смешанного решения заключается в том, что при многократном повторении "ходов" при неизменных условиях игры, игрок A ничего не потеряет (это лучше, чем проиграть 3), если случайным образом будет чередовать свои стратегии в соответствии с найденным p_i. Этот смысл скорее математический, чем практический. Он вытекает из тезиса "ничего не знаю и знать не желаю о намерениях противника; на все есть воля божья". Лучше, конечно, было бы получить недостающую информацию, чем бросать игральные кости для случайного выбора. Однако, если противник не конкретное лицо, а выражение некоторых случайных обстоятельств, то приготовиться к худшему бывает тоже полезно.

7.3. Теорема двойственности

Пара двойственных задач приводит к двум решениям относительно цены игры ν. В теории доказывается, что максимум линейной функции в одной из задач равен минимуму линейной функции в другой, так что ответ получается один. Этот результат связан с важной теоремой двойственности в линейном программировании.

В общем случае, любая теорема двойственности для некоторого утверждения данной теории позволяет сформулировать по определенному правилу другое утверждение таким образом, что из справедливости первого автоматически следует справедливость второго. Теоремы двойственности выражают глубинные свойства симметрии решаемых задач.

Если прямая задача заключается в минимизации функции

$$f(x) = \sum_{j=1}^{m} c_j x_j$$

при ограничениях

$$\sum_{j=1}^{m} a_{ij} x_j \ge b_i , \quad i = 1, 2, \dots, n ,$$

то двойственная задача будет следующей: Необходимо максимизировать функцию

$$\varphi(y) = \sum_{i=1}^{n} b_i y_i$$

при ограничениях $$\sum_{i=1}^{n} a_{ji} y_i \le c_j, \quad j = 1, 2, \ldots, m.$$

Из сравнения двух задач их симметричность очевидна. Так матрица из коэффициентов при переменных в прямой задаче есть транспонированная матрица обратной задачи. Действительно, обращаясь к правилу построения неравенств вида (7.I), но уже относительно игрока B, замечаем, что математическое ожидание проигрыша B при i-й стратегии игрока A записывается

$$a_{i1} g_1 + a_{i2} g_2 + \ldots + a_{in} g_n \le \nu \quad . \qquad (7.5)$$

Ведь отход A от оптимальной стратегии уменьшит его выигрыш. Сравнивая (7.I) с (7.5) мы видим, что переход от одной матрицы коэффициентов к другой получается путем транспонирования матриц. Тут же замечаем, что если в исходной задаче имеется m переменных и n ограничений, то в двойственной – n переменных и m ограничений.

Теорема двойственности гласит: пусть дана пара двойственных задач линейного программирования. Тогда, если исходная задача имеет решение, то и двойственная ей также имеет решение, при этом оптимальные значения целевых функций равны:

$$\min f(x) = \max \varphi(y) .$$

Теорема двойственности и полезна, и любопытна. Полезна тем, что в некоторых случаях двойственную задачу решать легче. А любопытна потому, что задача максимизации (или минимизации) может быть сведена к решению некоторой системы линейных неравенств без каких бы то ни было упоминаний о максимизации. Этот промежуточный вывод теоремы ничуть не облегчает поиск оптимального решения (в области неотрицательных значений), но может быть подтолкнёт вдумчивого читателя к более глубокому взгляду на роль симметрии в науке.

8. ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

8.1. Метод динамического программирования. Принцип пошаговой оптимизации

Динамическое программирование является методом оптимизации решений при планировании многошаговых или многоэтапных операций. Чаще всего он применяется к динамическим системам, в которых изменение состояния во времени может быть описано последовательностью дискретных шагов (например, через Δt). Однако он применим и к задачам, в которых время в явном виде не присутствует.

Рассмотрим операцию, состоящую из m шагов (этапов). Это может быть планирование перевозок по годам. Выберем показатель эффективности W, который будем также называть выигрышем. Предположим, что выигрыш за всю операцию складывается из выигрышей на отдельных шагах:

$$W = \sum_{i=1}^{m} w_i ,$$

где w_i – выигрыш на i-м шаге.

Если W обладает таким свойством, то его называют аддитивным критерием. Большое число практических задач характеризуется аддитивными критериями эффективности.

Для целенаправленного изменения хода операции выбирается управление. Управление понимается в широком смысле. В динамических объектах управлением могут служить отклонения рулей, но им могут быть и материальные средства, вкладываемые в производство. Можно сказать, что для управления процессом выбираются некоторые величины (угол отклонения рулей, кредиты и т.п.), влияющие на ход процесса, причем на каждом шаге они определяют выигрыш w_i. Эти величины называются шаговым управлением. Совокупность всех шаговых управлений представляет собой управление операцией в целом:

$$x = (x_1, x_2, \dots , x_m).$$

Следует иметь в виду, что все x_i в общем случае являются необязательно числами (см.далее).

Требуется найти такое управление x, при котором выигрыш W обращается в максимум:

$$W = \sum_{i=1}^{m} W_i \longrightarrow \max .$$

То управление x^*, при котором этот максимум достигается, называется оптимальным управлением. Оно состоит из совокупности оптимальных шаговых управлений:

$$x^* = (x_1^*, x_2^*, \ldots, x_m^*).$$

Тот максимальный выигрыш, который достигается при этом управлении, обозначается W^*:

$$W^* = \max_{x \in X} \{ W(x) \}.$$

Здесь отмечено, что управление x выбирается из множества возможных X в данных условиях. Если X не задано, т.е. неопределено из чего выбирать, то задача несформулирована. От выбора X, в частности, зависит решение задачи.

Для примера рассмотрим проектирование железнодорожного пути между двумя пунктами (A и B). Местность пересеченная, включает холмы, болота, реки и т.п. Требуется так спланировать путь из A в B, чтобы суммарные затраты были минимальными.

В задаче нет естественного членения на шаги; его приходится вводить искусственно. Для этого можно использовать различные приемы. Самый простой – допустить, что прокладка пути состоит из ряда шагов, и на каждом шаге можно двигаться только вдоль прямоугольной сетки, наложенной на план местности (рис. 8.I).

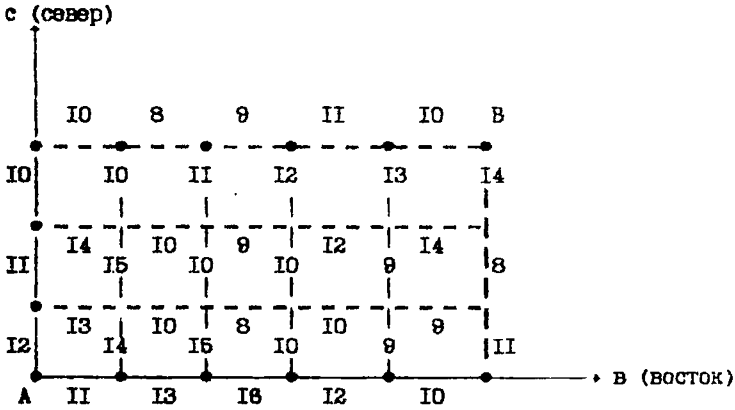


Рис. 8.I

Цифры на сетке указывают стоимость прокладки пути на соответствующем отрезке.

Будем считать, что двигаться можно либо строго на восток, либо строго на север. Тогда любой путь из A в B представляет собой ломанную линию. Управление же заключается в выборе направления и его можно записать последовательностью букв "с" (север) и "в" (восток). Например, возможно управление, переводящее из A в B:

$$x = (\text{в},\text{в},\text{в},\text{в},\text{в},\text{с},\text{с},\text{с}),$$

т.е. путь, проходящий по 2-м сторонам большого прямоугольника. Область X графически изображается прямоугольной сеткой.

Задачу можно попытаться решить перебором всевозможных вариантов маргрутов, число которых вычисляется по правилам комбинаторики. Достаточно очевидно, что как и в случае основной задачи линейного программирования такой путь неперспективен. Нужен целенаправленный поиск, и не слепой перебор.

Для решения задачи методом динамического программирования используется принцип пошаговой оптимизации. Оптимизация одного шага, как правило, проще оптимизации всего процесса: лучше много раз решать простую задачу, чем один раз сложную. Сразу отметим, что динамическое программирование не предполагает, что каждый шаг оптимизируется отдельно, независимо от других. Напротив, шаговое управление выбирается с учетом последствий выбора. Что толку в выборе управления, приносящего максимальный выигрыш на данном шаге, если в дальнейшем он обусловит разорение (заведет дорогу в болото в нашей задаче).

Значит планируя многошаговую операцию, надо выбирать управление на каждом шаге с учетом всех его будущих последствий на ещё предстоящих шагах. Иначе говоря, управление на i-м шаге выбирается так, чтобы была максимальной сумма выигрышей на данном шаге и на всех оставшихся до конца шагах.

Нетрудно догадаться, чтобы реализовать такой принцип оптимизации надо "пятиться" с конца. Ведь среди всех шагов есть один, который может планироваться без оглядки на будущее. Очевидно, это последний шаг. Рассмотрев его единственным и выбрав его оптимальным, мы можем перейти теперь к предпоследнему шагу.

Процесс динамического программирования разворачивается от конца к началу: сначала планируется последний, m-й шаг. Но ведь, чтобы *его спланировать, нужно знать, чем кончился* предпоследний (m-I)-й шаг, т.е. нужно знать условия, в которых мы подошли к последнему шагу. Их мы не знаем, поэтому по очереди перебираем все возможные состояния, в которых могли бы оказаться перед последним шагом, и для каждого из них ищем оптимальное управление (из множества допустимых). Это оптимальное управление называется условным на m-м шаге, поскольку относится только к данному состоянию, из которого можем сделать m-й шаг. Выбрав условное оптимальное управление, мы тем самым исключаем из дальнейшего рассмотрения все остальные неоптимальные управления, а значит сокращаем число вариантов перебора возможных решений.

Итак, для m-го шага мы определили все возможные начальные состояния и для каждого из них мы нашли по одному условному оптимальному управлению. Теперь можно оптимизировать предпоследний (m-1)-й шаг; причем начальные состояния m-го шага будут возможными конечными для предпоследнего (m-I)-го шага. Для каждого из них процедура поиска условного оптимального управления в точности повторяется. Снова делаются предположения о том, чем может кончиться предыдущий (m-2)-й шаг, и для каждого из этих предположений найдем такое управление на (m-1)-м шаге, при котором выигрыш за последние два шага (из которых m-й уже оптимизирован) максимален. Так мы найдем для каждого исхода (m-2)-го шага условное оптимальное управление на (m-I)-м шаге и условный оптимальный выигрыш на двух последних шагах. Далее, "пятясь назад", оптимизируем управление на (m-2)-м шаге и т.д., пока не доведем до первого. В результате для всех возможных состояний процесса (узлов сетки в нашей задаче) мы будем иметь условные оптимальные управления и условные выигрыши. Для первого шага не надо строить никаких предположений: начальное состояние S_0 задано в условии задачи. А раз задано, то для него получаем не условное, а безусловное оптимальное управление x_1^* и выигрыш на первом шаге. Осуществив это управление мы приведем процесс (систему) в новое состояние S_1^* – исходное для второго шага. Для него уже имеется найденное условное оптимальное управление x_2^*, которое к концу второго шага переведет систему в S_2^* и т.д. Теперь мы уже двигаемся от начала к концу. Что касается оптимального выигрыша W^* за всю операцию, то он нам также известен – это выигрыш на первом шаге, который складывается из первого и всех последующих и который выбран из условия его максимальности.

Таким образом, в процессе оптимизации управления методом динамического программирования многошаговый процесс "проходится" дважды: первый раз – от конца к началу, в результате чего находятся условные оптимальные управления и выигрыши за оставшийся "хвост" для всех возможных состояний; второй раз – от начала к концу, когда путь определяется безусловно, заданным начальным состоянием.

Поскольку по ходу изложения метода время явно не присутствует в выкладках, то начальное состояние можно принять за конечное и наоборот, задача совершенно симметрична по отношению к начальному и конечному состояниям.

8.2. Принцип оптимальности. Пример планирования маршрута движения (железнодорожного пути)

Метод динамического программирования достаточно универсален. В отличие от линейного программирования он "работает" и при нелинейной целевой функции – критерии W, нелинейных ограничениях, "решает" целочисленные задачи. Однако динамическое программирование не сводится к какой-либо стандартной вычислительной процедуре типа симплекс-метода. Для каждой конкретной задачи пишется свой алгоритм вычислений, но, конечно, основанный на принципе оптимальности. Какого-либо автоматического способа получения алгоритма нет, нужны опыт и интуиция исследователя.

Напомним принцип оптимальности. Каково бы ни было состояние системы S перед очередным шагом, надо выбирать управление на этом шаге так, чтобы выигрыш на данном шаге плюс выигрыш на всех последующих шагах был максимальным.

Решение задачи методом динамического программирования удобно проводить в следующей последовательности.

1. Расчленить операцию на этапы (шаги).

2. Выбрать параметры, характеризующие состояние S управляемого процесса (системы) перед каждым шагом.

3. Выяснить набор шаговых управлений x_i для каждого шага и налагаемые на них ограничения (определить область X).

4. Определить, какой выигрыш приносит на i-м шаге управление x_i, если перед этим система была в состоянии S, т.е. записать "функции выигрыша":

$$w_i = w_i\ (S,\ x_i). \tag{8.1}$$

5. Определить, как изменяется состояние S системы под влиянием управления x_i:

$$S_{i+1} = S_i\ (x_i), \tag{8.2}$$

где S_i и S_{i+1} – наборы (гаммы) возможных состояний.

6. Записать основное рекуррентное уравнение динамического программирования, выражающее условный оптимальный выигрыш $W_i(S)$ (начиная с i-го шага и до конца) через уже известную функцию $W_{i+1}(S)$:

$$W_i(S) = \max_{x_i} \{ w_i(S_i, x_i) + W_{i+1}\ [S_{i+1}]\}. \tag{8.3}$$

Этому выигрышу соответствует условное оптимальное управление на i-м шаге $x_i(S)$.

7. Произвести условную оптимизацию последнего m-го шага, задаваясь гаммой возможных предпоследних состояний. Условный оптимальный выигрыш определяется:

$$W_m(S) = \max_{x_m} \{ w_m(S, x_m)\} \quad . \tag{8.4}$$

Ему отвечает условное оптимальное управление $x_m(S)$.

8. Произвести условную оптимизацию (m-I)-го, (m-2)-го и т.д. шагов по формуле (8.3), и указать для них условное оптимальное управление $x_i(S)$. Оптимальный выигрыш за всю операцию равен выигрышу на первом шаге (поскольку остальные выигрыши в него включаются).

$$W^* = \max_{x_1} \{W_1(S_0)\} \ .$$

Соответствующее управление равно x_1^*.

9. Произвести безусловную оптимизацию управления, определяя шаговые управления в прямой последовательности с первого и до последнего и изменяя состояния системы согласно (8.2).

Продолжим рассмотрение примера из п. 8.I. В качестве множества S примем узлы координатной сетки. Членение операций на шаги определяется масштабом сетки, его выбор произволен. Однако при малом m точность проекта будет мала. При очень большом m возрастает трудоемкость расчетов. Шаговые управления, как было уже отмечено, записываются с помощью "с" или "в". Выигрыши (здесь это затраты, но их всегда можно рассматривать с обратным знаком) представлены на рис. 8.I на ребрах сетки. Далее оптимизацию удобно проводить следующим образом.

Любой путь из A в B состоит из m= 5+3= 8 отрезков. Начнем оптимизацию с последнего 8-го шага. Рассмотрим отдельно окрестность (•) B, в которой мы можем находиться после 7-го шага (рис. 8.2,а). Если мы находимся в (•) S_7^1, у нас нет выбора (управление вынужденное): надо идти на восток, и это обойдется нам в IO единиц, записанных в кружке S_7^1. Оптимальное управление показано стрелкой (означает "в"). Для (•) S_7^2 управление тоже вынужденное (север "с") с затратами I4 (тоже указаны в кружке). Таким образом, условная оптимизация последнего шага выполнена: для каждого S_7 найдены условные оптимальные "выигрыши" и управления.

Перейдем к *оптимизации предпоследнего 7-го шага*. Перед ним мы могли бы оказаться в одной из точек S_6^1, S_6^2, S_6^3, рис. 8.2,б. Найдем для каждой из них условное оптимальное управление и условный оптимальный выигрыш. Для S_6^1 управление вынужденное – "в"; обойдется нам до конца в 2I единицу (II на данном шаге, плюс IO, записанных в кружке при S_7^1). Число 2I записываем в кружке S_6^1. Для (•) S_6^2 управление уже не вынужденное: мы можем идти как на восток, так и на север. В первом случае мы затратим на данном шаге I4 единиц, плюс от S_7^2 до (•)В еще I4, всего 28. Если пойдем на север, затратим I3+IO, всего 23 единицы. Значит, условное оптимальное управление в (•) S_6^2 – идти на север (отмечаем стрелкой, а число 23 записываем в кружке у S_6^2). Для (•) S_6^3 управление снова вынужденное ("с"), обойдется это до конца в 22 единицы (стрелка – на север, 22 – в кружке).

Аналогично пятясь от предпоследнего шага назад, найдем для каждого узла координатной сетки условное оптимальное управление ("с" или "в"), которое обозначим стрелкой, и условный оптимальный выигрыш (расход до конца пути), который запишем в кружке. Конечный результат процедуры оптимизации показан на рис. 8.3.

Таким образом, условная оптимизация уже выполнена: в какой бы из узловых точек мы ни находились, мы уже знаем, куда идти (стрелка) и во что нам обойдется путь до конца (число в кружке). В (•) А записан оптимальный выигрыш на весь проект пути из А в В: $W^* = 81$.

Теперь остается построить безусловное оптимальное управление, определяющее траекторию из А в В самым дешевым способом. Для этого нужно проследить путь в направлении стрелок от начала до конца, составляющие этого пути укажут на оптимальное управление. Такой оптимальной траекторией на рис. 8.3 будут левая и верхняя стороны большого прямоугольника координатной сетки, а управлением

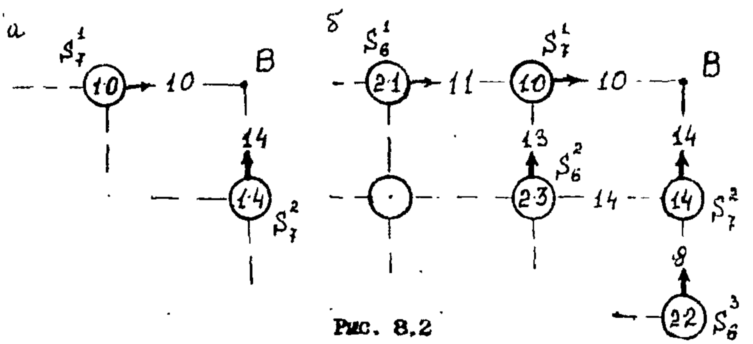


Рис. 8.2

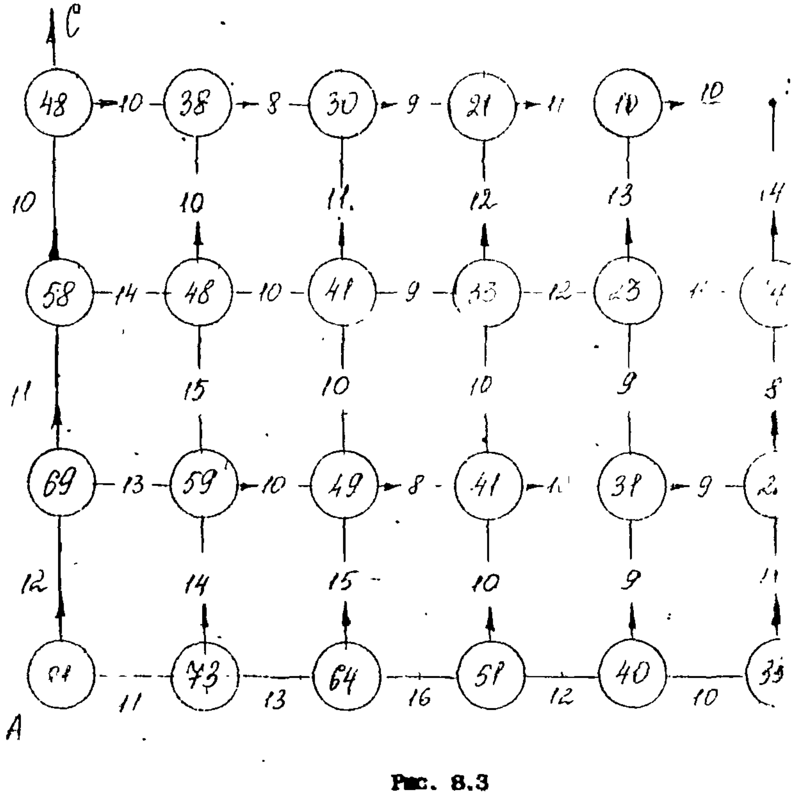


Рис. 8.3

$$x^* = (c,c,c,B,B,B,B,B).$$

Если бы мы начали строить траекторию недальновидно, на каждом шаге выбирая участок наименьшей стоимости, то получили бы управление

$$x = (B,B,C,B,C,B,C,B).$$

Здесь пятый шаг и далее давал нам свободу выбора (стоимость "с" и "в" была бы одинаковой и для $i = 5$ равнялась 10). Стоимость соответствующего пути составляет $W = 92 > W^*$.

Задачи, сходные с рассмотренной, часто встречаются на практике: например, определение наискорейшей траектории экстренного снижения самолета или наиболее экономного (в смысле расхода топлива) набора скорости и высоты полета.

8.3. Задача об оптимальной загрузке самолета методом динамического программирования

Пользуясь методом динамического программирования, можно с успехом решать многие задачи оптимизации. В частности, задачу о распределении ресурсов, некоторые задачи целочисленного программирования.

Вернёмся к первому примеру задачи целочисленного программирования. Имеется набор контейнеров $П_1$, $П_2$,..., $П_n$ (каждый в единственном экземпляре); известны их веса $g_1, g_2, \ldots, g_n$ и стоимости $v_1, v_2, \ldots, v_n$. Грузоподъемность самолета равна Q. Требуется определить набор контейнеров для погрузки в самолет, чтобы суммарная стоимость их была максимальна.

Процесс загрузки представим состоящим из n шагов; на каждом шаге мы отвечаем на вопрос: брать данный контейнер или не брать? Принимается, что управление на i-м шаге равно единице, если мы данный i-й контейнер берем, и нулю – если не берем, т.е. $x_i \in \{0,1\}$.

Состояние системы S будем характеризовать весом S, который ещё остался в нашем распоряжении до полной загрузки после того, как предыдущие шаги выполнены (какие-то контейнеры взяты). Согласно изложенным рекомендациям мы должны для каждого S найти $W_i(S)$ – суммарную максимальную стоимость контейнеров, которыми можно догрузить самолет при данном S.

Зададимся следующими численными значениями.

Таблица 8.I

Контейнер i	I		2		3		4		5		6	
Управление x_i	I	0	I	0	I	0	I	0	I	0	I	0
Вес $g(x_i)$	4	0	7	0	II	0	I2	0	I6	0	20	0
Стоимость $w(x_i)$	7	0	I0	0	I5	0	20	0	27	0	34	0

Таблица несколько загромождена нулевыми столбцами, которые напоминают, что если контейнер не брать (x_i = 0), то его вес и стоимость при расчетах принимаются равными нулю. Грузоподъемность Q = 35.

Условная оптимизация решения показана в табл. 8.2, где в каждой строке для каждого i-номера шага (равного номеру контейнера) приведены: условное оптимальное управление x_i (0 или I) и условный оптимальный выигрыш W_i (суммарная максимальная стоимость всех контейнеров при оптимальном управлении на всех шагах, оставшихся до конца). Сразу отметим, чтобы составить эту таблицу предварительно строятся вспомогательные таблицы (см. далее). Таблица заполняется слева направо, сверху вниз. Без вспомогательных таблиц заполняются только две первые колонки для x_6 и W_6 (i=6). Они получаются непосредственно из табл. 8.I. Если оставшийся свободным до заполнения Q вес S меньше 20, то управление вынужденное

x_6 = 0.

Таблица 8.2

S	i = 6		i = 5		i = 4		i = 3		i = 2		i = 1	
	x_6	W_6	x_5	W_5	x_4	W_4	x_3	W_3	x_2	W_2	x_1	W_1
0	$\boxed{0}$	0	0	0	0	0	0	0	0	0		
1	0	0	0	0	0	0	0	0	0	0		
2	0	0	0	0	0	0	0	0	0	0		
3	0	0	0	0	0	0	0	0	0	0		
4	0	0	0	0	0	0	0	0	0	0		
5	0	0	0	0	0	0	0	0	0	0		
6	0	0	0	0	0	0	0	0	0	0		
7	0	0	0	0	0	0	0	0	1	10		
8	0	0	0	0	0	0	0	0	1	10		
9	0	0	0	0	0	0	0	0	1	10		
10	0	0	0	0	0	0	0	0	1	10		
11	0	0	0	0	0	0	1	15	0	15		
12	0	0	0	0	1	20	0	20	0	20		
13	0	0	0	0	1	20	0	20	0	20		
14	0	0	0	0	1	20	0	20	0	20		
15	0	0	0	0	1	20	0	20	0	20		
16	0	0	$\boxed{1}$	27	0	27	0	27	0	27		
17	0	0	1	27	0	27	0	27	0	27		
18	0	0	1	27	0	27	0	27	0	27		
19	0	0	1	27	0	27	0	27	1	30		
20	1	34	0	34	0	34	0	34	0	34		
21	1	34	0	34	0	34	0	34	0	34		
22	1	34	0	34	0	34	0	34	0	34		
23	1	34	0	34	0	34	1	35	1	37		
24	1	34	0	34	0	34	1	35	1	37		
25	1	34	0	34	0	34	1	35	1	37		
26	1	34	0	34	0	34	1	35	1	37		
27	1	34	0	34	0	34	1	42	1	44		
28	1	34	0	34	$\boxed{1}$	47	0	47	0	47		
29	1	34	0	34	1	47	0	47	0	47		
30	1	34	0	34	1	47	0	47	0	47		
31	1	34	0	34	1	47	1	49	0	49		
32	1	34	0	34	1	54	0	54	0	54		
33	1	34	0	34	1	54	0	54	0	54		
34	1	34	0	34	1	54	0	54	0	54		
35	1	34	0	34	1	54	0	54	$\boxed{1}$	57	$\boxed{0}$	57

Действительно, вес $g_6 = 20$ не помещается в допуск, и шестой контейнер мы взять не можем. Если же $S \geq 20$, то надо брать контейнер $x_6 = 1$, ведь шаг-то последний; не оставлять же свободное место, которое можно заполнить.

Для заполнения остальных столбцов нужно строить вспомогательные таблицы 8.3 - 8.5. Например,
$i = 5$; $W_6\,[S - g(x_5)]$ берутся из колонки для $i=6$ (табл. 8.2).

Таблица 8.3

S	x_5	$g(x_5)$	$S-g(x_5)$	$w(x_5)$	$W_6[S-g(x_5)]$	$w_5+W_6=W_5$
15	|0|	0	15	0	0	|0|
	т.к. 15 < $g(x_5)$ =16, то управление вынужденное					
16	0 |1|	0 16	16 0	0 27	$W_6[16]=0$ $W_6[0]=0$	0 |27|
	.	. .	. .	. .	. . . .	.
19	0 |1|	0 16	19 3	0 27	$W_6[19]=0$ $W_6[3]=0$	0 |27|
20	|0| 1	0 16	20 4	0 27	$W_6[20]=34$ $W_6[4]=0$	|34| 27
	.	. .	. .	. .	. . . .	.
35	|0| 1	0 16	35 19	0 27	$W_6[35]=34$ $W_6[19]=0$	|34| 27

Таблица 8.4

$i = 3$

S	x_3	$g(x_3)$	$S-g(x_3)$	$w(x_3)$	$W_4[S-g(x_3)]$	$w_3+W_4=W_3$
•	•	•	•	•	•	•
30	|0| 1	0 11	30 19	0 15	$W_4[30]=47$ $W_4[19]=27$	|47| 15+27=42
31	0 |1|	0 11	31 20	0 15	$W_4[31]=47$ $W_4[20]=34$	47 15+34=|49|
32	|0| 1	0 11	32 21	0 15	$W_4[32]=54$ $W_4[21]=34$	|54| 15+34=49
•	•	•	•	•	•	•

Таблица 8.5

$i = I, \quad S = Q = 35$

Первый шаг с точно известными условиями

S	x_1	$g(x_1)$	$35-g(x_1)$	$w(x_1)$	$W_2[35-g(x_1)]$	$w_1+W_2=W_1$
35	|0| 1	0 4	35 31	0 7	$W_2[35]=57$ $W_2[31]=47$	|57| 54

Каждая строчка в табл. 8.2 заполняется числами в рамочках для соответствующих S и i из таблиц вида 8.3-8.5. Однако в самой таблице 8.2 рамочками отмечены безусловные оптимальные управления всей задачи (не путать с условными из табл. 8.3-8.5).

Для того, чтобы определить безусловное оптимальное управление, т.е. ответить на основной вопрос задачи – какие контейнеры грузить, надо теперь все шаги проследить в прямой последовательности. Для $i=I$, поскольку в нашем распоряжении весь самолет – $S=35$, оптимальное управление согласно табл. 8.2 $x_1^*=0$ (контейнер № I не берем). На втором шаге $i=2$, и мы по-прежнему располагаем полной загрузкой $S=35$; согласно табл. 8.2 $x_2^*=1$. Теперь поскольку взят $g_2=7$, в нашем распоряжении осталось $S=35-7=28$. Для шага $i=3$ и $S=28$ из табл. 8.2 следует, что $x_3^*=0$. Переходим к шагу $i=4$ с прежним значением возможностей для загрузки $S=28$; для них $x_4^*=1$. Пятый шаг $i=5$ начинается при условии $S=28-g_4=16$, для него оптимальным будет $x_5^*=1$. Для $i=6$ аналогичным образом получаем $x_6^*=0$ (это можно было и не определять, поскольку свободного места и так не оставалось). Таким образом, следует брать контейнеры № 2, 4 и 5, что дает выигрыш в 57 единиц.

В заключение добавим, что метод динамического программирования применим также и к задачам с так называемым "мультипликативным" критерием, имеющим вид произведения

$$W = \prod_{i=1}^{m} w_i \, , \quad w_i > 0 \, .$$

Эти задачи решаются также, как и с аддитивным критерием, с той единственной разницей, что в основном уравнении (8.3) вместо знака "плюс" ставится знак умножения:

$$W_i(S) = \max_{x_i} \{ w_i(S, x_i) \times W_{i+1}(S) \} \, .$$

8.4. Уравнение Беллмана

В редких случаях принцип оптимальности позволяет элегантно решать задачи в аналитическом виде, т.е. вместо вычислительного алгоритма, указывающего на дорогу решения задачи, можно получить формулу, дающую прямой ответ. Такая формула, если очень повезет, получается из решения уравнения Беллмана.

Рассмотрим управляемый объект, описываемый для примера системой дифференциальных уравнений (в общем случае нелинейных) первого порядка:

$$\begin{cases} \dot{y}_1 = f_1\ (y_1,\ y_2,\ U); \\ \dot{y}_2 = f_2\ (y_1,\ y_2,\ U), \end{cases} \tag{8.5}$$

где $y_1(t)$ и $y_2(t)$ – переменные, описывающие состояние координаты объекта;

$U(t)$ – искомое управление.

На управление $U(t)$ наложено ограничение:

$$U(t) \in U\ . \tag{8.6}$$

Требуется среди всех допустимых из (8.6) управлений найти такую функцию $U(t)$, чтобы объект, описываемый (8.5), перешёл из заданного начального состояния $S_0 = 0$ в конечное B за минимальное время:

$$W = T = \int_0^T dt\ .$$

Предположим, что существует оптимальное управление $U^*(t)$, переводящее систему в состояние B за минимальное время. Пусть этому управлению соответствует верхняя траектория с произвольной точкой A (рис. 8.4). Время, затрачиваемое на переход из A в B, тоже будет минимальным, если двигаться по той же траектории. Это вытекает из принципа оптимальности. Действительно, рассматривая процесс в виде шага OA и суммы оставшихся до конца шагов в виде AB, заключаем, что для максимального выигрыша на OB должен быть максимальный выигрыш на OA плюс максимальный выигрыш на AB. Этот же ход рассуждений будет повторен и для точек A и A'. Обозначим время из A в B через $T^*[y_1(t),\ y_2(t)]$, подчеркивая, что точка A определяется координатами y_1 и y_2. Сделаем из A шаг Δt, после которого, двигаясь по оптимальной траектории, мы окажемся в точке A' либо не по оптимальной – в точке A''.

Согласно принципу оптимальности, управление выбирается из условия максимального выигрыша на данном шаге Δt плюс оптимальный выигрыш на всех оставшихся до конца шагах. Выигрышем является время, для которого справедливо

$$T^{*}[y_1(t),\ y_2(t)] \le \Delta t + T[y_1(t+\Delta t),\ y_2(t+\Delta t)],$$

где равенство выполняется для точки A' и не выполняется для A''. Время, траектория и управление нам неизвестны. Мы их собираемся найти используя принцип оптимальности. Запишем основное рекуррентное соотношение оптимизации на шаге Δt (заменяя по условию max на min) в соответствии с (8.3):

$$\min_{u\in U}\ \{T[y_1(t),\ y_2(t)]\} = \min\ \{\Delta t + T[y_1(t+\Delta t),\ y_2(t+\Delta t)]\}.$$

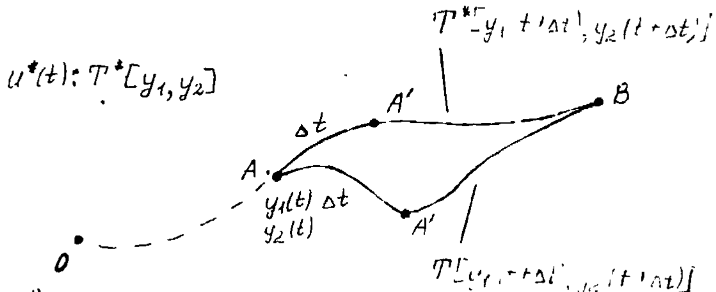


Рис. 8.4

Данное условие можно переписать в виде

$$\min_{u\in U}\left\{\frac{T[y_1(t+\Delta t),\ y_2(t+\Delta t)] - T[y_1(t),\ y_2(t)]}{\Delta t}\right\} = -1.$$

Переходя к пределу при $\Delta t \longrightarrow 0$, находим

$$\min_{u\in U}\left\{\frac{d}{dt}\Big[T[y_1(t),\ y_2(t)]\Big]\right\} = -1\ . \tag{8.7}$$

То управление, которое обеспечит движение по такой траектории, будет оптимальным $U^{*}(t)$. Соотношение (8.7) носит название уравнения Беллмана. Применительно к данному примеру, используя правила дифференцирования сложной функции, его можно представить в виде:

$$\min_{u\in U}\left\{\frac{\partial T}{\partial y_1}\cdot\frac{\partial y_1}{\partial t}+\frac{\partial T}{\partial y_2}\,\frac{\partial y_2}{\partial t}\right\}=-1 \qquad (8.8)$$

или, воспользовавшись (8.5), в знакомой форме

$$\min_{u\in U}\left\{\frac{\partial T}{\partial y_1}\,f_1\,(y_1,\,y_2,\,U)+\frac{\partial T}{\partial y_2}\,f_2\,(y_1,\,y_2,\,U)\right\}=-1.(8.9)$$

Уравнение Беллмана означает, что, если в каждый момент времени, минимальное значение производной от T, как функции от y_1 и y_2, равно – I, то движение происходит по оптимальной траектории и этому движению соответствует оптимальное управление. Соответственно, если мы решим это уравнение относительно неизвестной функции U(t), то получим управление, переводящее объект из O в B по оптимальной траектории.

Применение уравнения Беллмана на практике сталкивается с большими трудностями. Неизвестны не только искомое управление и траектория, т.е. $y_1(t)$ и $y_2(t)$, но и сама дифференцируемая функция T. Может даже оказаться, что не во всех точках функция T имеет производные.

8.5. Пример решения уравнения Беллмана для плоского движения

Предположим, что при плоском движении центра масс объекта управления, его скорость пропорциональна управляющему воздействию, т.е.

$$\left\{\begin{aligned}\frac{d\,y_1}{d\,t}&=U_1\ ;\\ \frac{d\,y_2}{d\,t}&=U_2\ .\end{aligned}\right. \qquad (8.10)$$

Требуется из любой точки пространства попасть в начало координат за наименьшее время.

Пусть, для простоты, управления таковы, что всегда

$$\sqrt{U_1^2 + U_2^2} = 1 , \qquad (8.11)$$

т.е. допустимое множество управлений U задается окружностью единичного радиуса. Практически мы имеем грубое описание полета управляемой ракеты на активном участке.

Оптимизируемым функционалом - критерием оптимальности - является время

$$W = \int_0^T dt .$$

Из физических соображений ясно, что из точки A в точку O при указанных условиях (рис. 8.5) можно попасть за наименьшее время, если двигаться по кратчайшему расстоянию - прямой OA. Поскольку в силу (8.11) движение происходит с постоянной единичной векторной скоростью, то это время равно

$$T_A = \frac{OA}{U} = \sqrt{y_{10}^2 + y_{20}^2} \,/\, 1 = \sqrt{y_{10}^2 + y_{20}^2} .$$

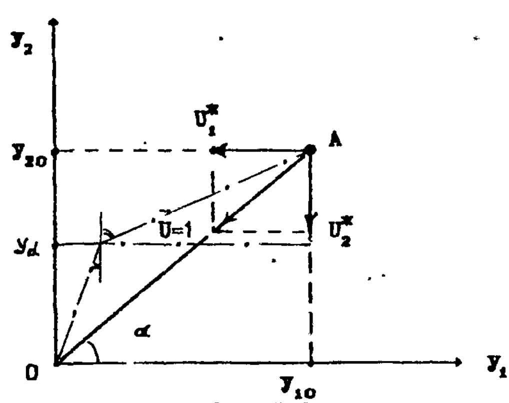


Рис. 8.5

Таким образом, для оптимального управления имеем

$$T = \sqrt{y_1^2 + y_2^2} \quad \text{и} \quad \partial T/\partial y_i = y_i \Big/ \sqrt{y_1^2 + y_2^2}\ , \ i = 1, 2 .$$

Подставляя эти выражения в уравнение Беллмана, получаем с учетом (8.10):

$$\frac{\partial T}{\partial y_1} f_1(U_1) + \frac{\partial T}{\partial y_2} f_2(U_2) =$$

$$= \frac{y_1}{\sqrt{y_1^2 + y_2^2}} U_1 + \frac{y_2}{\sqrt{y_1^2 + y_2^2}} U_2 = -1 .$$

Легко видеть, что оптимальное управление $U^* = (U_1^*, U_2^*)$ задается соотношениями, являющимися решениями алгебраического уравнения

$$U_1^* = - \frac{y_1}{\sqrt{y_1^2 + y_2^2}} = -1 \cos\alpha;$$

$$U_2^* = - \frac{y_2}{\sqrt{y_1^2 + y_2^2}} = -1 \sin\alpha .$$

Полученное управление соответствует направлению U вдоль отрезка OA, указанному на рис. 8.5. В нашей задаче оптимальная траектория получена исходя из физических соображений, а управление - путем решения уравнения Беллмана.

Теперь видоизменим ограничения (штрихпунктир на рис. 8.5). Пусть при $y > y_d$ $U=1$, а при $y \le y_d$ скорость будет меньше: $U=1/n$, $n > 1$. Нетрудно заключить, что оптимальная траектория будет ломаной прямой. Действительно, мы получили задачу на преломление луча в геометрической оптике, где $n = c/v$ - показатель преломления, равный отношению скоростей света в двух средах. Эту задачу мы формулируем как оптимизационную. Согласно принципу (опять принцип!) наименьшего времени Ферма излом луча соответсвует закону преломления Снеллиуса. Наши уравнения безинерционны, поэтому согласно полученному решению для U^*, оптимальное управление должно быть направлено вдоль оптимальной траектории.

8.8. Обобщенное уравнение Беллмана

Принцип оптимальности может быть применим не только к случаю, когда критерием оптимальности является время T перехода из состояния в состояние.

Пусть управляемая система описывается уравнениями

$$\dot{y}_i = f_i(y_1, y_2, \ldots, y_n; u_1, u_2, \ldots, u_m), \; i = 1, 2, \ldots, n, \quad (8.12)$$

а критерий качества задается функционалом

$$V = \int_{t_0}^{t_1} f_0 \; (y_1, y_2, \ldots, y_n; u_1, u_2, \ldots, y_m) \, dt \, . \quad (8.13)$$

Достаточно часто используется квадратичный функционал, который применительно к предыдущей задаче мог бы иметь вид

$$V = \int_0^T (g_1 y_1^2 (t) + g_2 y_2^2 (t) + g_3 u_1^2 (t) + g_4 u_2^2 (t) \, dt \, ,$$

где все g_i, $i = 1, 2, 3, 4$ - некоторые положительные числа, задаваемые, вообще-то, произвольно.

Выбор функционала диктуется техническими требованиями к системе и производится разработчиком из физических соображений. Обычно $f_0 > 0$ на всей области определения функции. Тогда f_0 можно формально рассматривать как некоторый масштабный множитель (переменный) при t, а подынтегральное выражение в (8.13) представить в виде

$$f_0\, dt = d\tau .$$

Здесь τ имеет смысл новой временной переменной, связанной со старым временем t соотношением

$$dt = \frac{d\tau}{f_0} . \tag{8.14}$$

Производные по времени, записанные с учетом (8.14), примут вид

$$\dot{y}_i = \frac{d\, y_i}{d\, t} = \frac{d\, y_i}{d\, \tau}\, f_0\, (y_1, y_2, \dots, y_n;\ u_1, \dots, u_m),$$

а уравнения (8.12) преобразуются

$$\frac{d\, y_i}{d\, \tau} = \frac{f_i\, (y_1, \dots, y_n;\ u_1, \dots, u_m)}{f_0\, (y_1, \dots, y_n;\ u_1, \dots, u_m)} . \tag{8.15}$$

Подставляя формулу (8.15) в уравнение (8.8) и распространяя суммирование на все n, получаем уравнение Беллмана для задачи оптимизации функционала W:

$$\min_{u \in U} \left\{ \sum_{i=1}^{n} \frac{\partial V}{\partial y_i} \, \frac{f_i\, (y_1, \dots, y_n;\ u_1, \dots, u_m)}{f_0\, (y_1, \dots, y_n;\ u_1, \dots, u_m)} \right\} = -1, \tag{8.16}$$

Здесь V – функция Беллмана, аналогичная функции T.

Решение уравнения Беллмана сопряжено с большими математическими трудностями, поэтому на практике используются вычислительные процедуры, аналогичные рассмотренным в первых параграфах главы.

9. ПРЯМЫЕ И ВАРИАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ

9.1. Простейшая оптимизационная задача нахождения экстремума целевой функции. Постановка задачи нелинейного программирования

Прямые методы оптимизации позволяют аналитически находить оптимальные решения исходя из общих представлений о нахождении экстремумов функций. Однако область их применения ограничена сравнительно простыми задачами из-за громоздкости вычислений.

Напомним общие методы решения простейших оптимизационных задач с ограничениями-равенствами. В этих методах с помощью ограничений переменные выражают друг через друга и подставляют в целевую функцию. Затем требуется найти безусловный экстремум целевой функции, который и обозначает решение задачи. Иногда используются неопределенные множители Лагранжа. Для иллюстрации метода возьмем пример.

Грузовой склад (база) берет на себя обязательство хранить груз и выдавать его потребителю в объеме r тонн ежедневно. Стоимость хранения h рублей за 1 тонну в сутки. База может получать груз только равными партиями g тонн и через равные промежутки времени T. Поступление груза производится в момент выдачи предыдущего. Расходы на загрузку не зависят от количества груза и равны p рублей.

Требуется определить оптимальный размер партии g и период его поставки T, чтобы суточные затраты на хранение были минимальными.

Стоимость хранения груза g в течение T равна hgT. Суммарные суточные затраты склада составляют

$$c = (hgT + p)/T, \tag{9.1}$$

где c – нелинейная функция двух переменных g и T.

Запишем дополнительное условие, означающее, что груза хватит точно до следующей поставки:

$$T = g/r$$

и подставим его в (9.1):

$$c = hg + pr/g .$$

Это уже функция одной переменной, для нахождения её экстремума найдем производную по g и приравняем её нулю:

$$\frac{dc}{dg} = h - \frac{pr}{g^2} = 0 .$$

Отсюда $g^2 = pr/h$ и $g^* = \sqrt{pr/h}$. Чтобы убедиться, что найден минимум, а не максимум, найдем вторую производную по g и установим её знак:

$$c'' = 2\frac{pr}{g^3} = 2\frac{h^{3/2}}{\sqrt{pr}} > 0 .$$

Мы видим, что функция принимает при $g^* = \sqrt{pr/h}$ наименьшее свое значение. При этом оптимальный период

$$T^* = \frac{g}{r} = \sqrt{\frac{p}{rh}} .$$

Суточные затраты составляют:

$$c = h\sqrt{\frac{pr}{h}} + \frac{pr\sqrt{h}}{\sqrt{pr}} = 2\sqrt{prh}$$

В более общем случае мы имеем задачу нелинейного программирования: минимизировать функцию

$$f(x) = f(x_1, x_2, \dots, x_n)$$

при ограничениях

$$\varphi_k(x) = \varphi_k(x_1, \dots, x_n) \le 0, \quad k = 1, \dots, m;$$

где $f(x)$, $\varphi_k(x)$ выпуклые функции в области действительных чисел.

Приведенная задача называется основной задачей выпуклого программирования. Легко видеть, что линейное программирование является её частным вариантом.

Решение задачи ищется с помощью функции Лагранжа

$$L(x, \lambda) = f(x) + \sum_{k=1}^{m} \lambda_k \varphi_k(x),$$

где $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, $\lambda_k \geq 0$ – неопределенные множители Лангранжа.

Для нахождения оптимального решения $x^* = (x_1^*, \ldots, x_n^*)$ составляется и решается система уравнений (и неравенств), получающаяся при дифференцировании L по x или λ. Пара (x^*, λ^*) называется седловой точкой функции Лагранжа. Однако из-за большой трудоемкости, за исключением простейших случаев, такой путь решения является малоэффективным и на практике почти не встречается.

Вернемся к задаче со складом. Для нее $f \equiv c = hg + p/T$ и ограничение $\varphi = Tr - g = 0$. Составим функцию Лагранжа

$$L = hg + p/T + \lambda_1 (Tr - g) = 0.$$

Находим производные

$$\frac{\partial L}{\partial g} = h - \lambda_1 = 0 \ ; \quad \lambda_1 = h;$$

$$\frac{\partial L}{\partial T} = - \frac{p}{T^2} + \lambda_1 r = 0 .$$

Отсюда получаем знакомые $T^* = \sqrt{p/rh}$.

9.2. Вариационное исчисление в задачах нахождения оптимальных решений

Вариационное исчисление применяется для нахождения экстремальных значений функционалов, т.е. переменных величин, зависящих от выбора одной или нескольких функций. Примером функционала может служить длина кривой (траектории полета между заданными точками). Если кривая изменяется, то изменяется, вообще говоря, и её длина. Следовательно, длина является переменной величиной, зависящей от выбора кривой, т.е. *от выбора функции*, определяющей уравнение кривой. Площадь, ограниченная замкнутой кривой, и т.д. - тоже пример функционала. Во всех указанных *случаях имеется такая зависимость, когда функции (или системе* функций) соответствует число.

Для примера рассмотрим задачу нахождения траектории экстренного снижения самолета в случае его разгерметизации или пожара на борту. Примем для простоты, что аэродинамическая подьемная сила перпендикулярна траектории, а тяга двигателей уравновешивает силу сопротивления. Требуется определить траекторию (кривую) $y = y(x)$ (рис. 9.I), соединяющую две заданные точки A и B, движение по которой займет наименьшее время. Решение задачи не изменится, если начальную скорость полета принять нулевой.

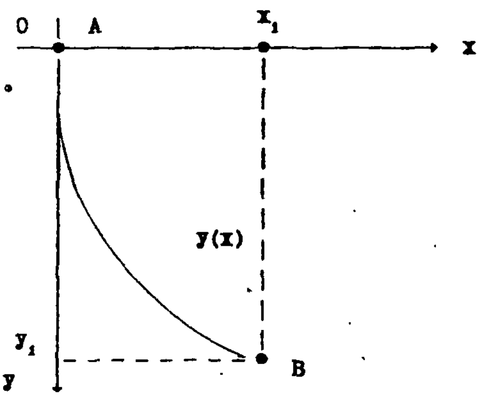


Рис. 9.I

Скорость тела, опустившегося на расстояние y, *определяется* в данном случае только силой тяжести и равна $\sqrt{2gy}$, где g - ускорение силы тяжести. Так как скорость равна производной от пути по времени, то можно записать:

$$\frac{dS}{dt} = \sqrt{2\,gy}$$

$$\text{или} \quad dt = \frac{dS}{\sqrt{2\,gy}} = \frac{\sqrt{1 + y'^2}}{\sqrt{2\,gy}}\,dx\,.$$

Здесь мы использовали известное соотношение между дифференциалом длины кривой и её производной по x. Интегрируя полученное выражение на интервале $[0,\ x_1]$, получаем

$$t = \frac{1}{\sqrt{2\,g}} \int_0^{x_1} \frac{\sqrt{1 + y'^2}}{\sqrt{y}}\,dx\,. \tag{9.2}$$

Задача свелась к нахождению кривой $y = y(x)$, проходящей через две заданные точки $A\ (0,\ 0)$ и $B\ (x_1\ y_1)$, на которой функционал

$$V = \int_0^{x_1} \frac{\sqrt{1 + y'^2}}{\sqrt{y}}\,dx$$

достигает минимума. Эта задача является частным случаем простейшей задачи вариационного исчисления и имеет историческое название задачи о брахистохроне.

Простейшая вариационная задача заключается в определении кривой $y = y(x)$, проходящей через две заданные граничные точки $(x_0,\ y_0)$ и $(x_1,\ y_1)$, на которой функционал

$$V[y(x)] = \int_{x_0}^{x_1} F[x, y(x), y'(x)]\,dx$$

достигает экстремума, т.е. наибольшего или наименьшего значения по сравнению со значениями этого функционала на всевозможных непрерывных и гладких дугах кривых, соединяющих те же две точки. В подинтегральном выражении F – есть данная функция от x, y, y' (она определена условиями конкретной задачи; например, в задаче о брахистохроне $F = \sqrt{1 + y'^2} / \sqrt{y}$), а $y=y(x)$ – искомая функция от x, от выбора которой зависит величина функционала.

Для нахождения $y(x)$ воспользуемся тем же подходом, который применялся для нахождения экстремума функции: тогда варьировалась переменная x, и если приращение функции с точностью до первого порядка относительно Δx равнялось нулю, то мы заключали, что находимся в экстремуме. В вариационном исчислении варьируется не аргумент, а сама функция.

Вводится вариация функции $\delta y(x)$, сама являющаяся функцией от x. Переход от кривой $y(x)$ к кривой сравнения $\tilde{y}(x)$ производится путем прибавления вариации δy:

$$\tilde{y}(x) = y(x) + \delta y.$$

В нашей задачи концы кривой закреплены (рис.9.2), поэтому

$$\delta y = 0 \quad \text{при } x=x_0 \text{ и } x=x_1. \tag{9.3}$$

Идея заключается в следующем. Мы как бы покачиваем с помощью δy некоторую кривую $y(x)$ и смотрим на поведение функционала. Если относительно величин первого порядка малости значение функционала неизменно (вариация самого функционала равна нулю $\delta V=0$), то кривая $y(x)$ является искомой, т.е. доставляющая экстремум V.

Для получения необходимых условий экстремума вычисляют

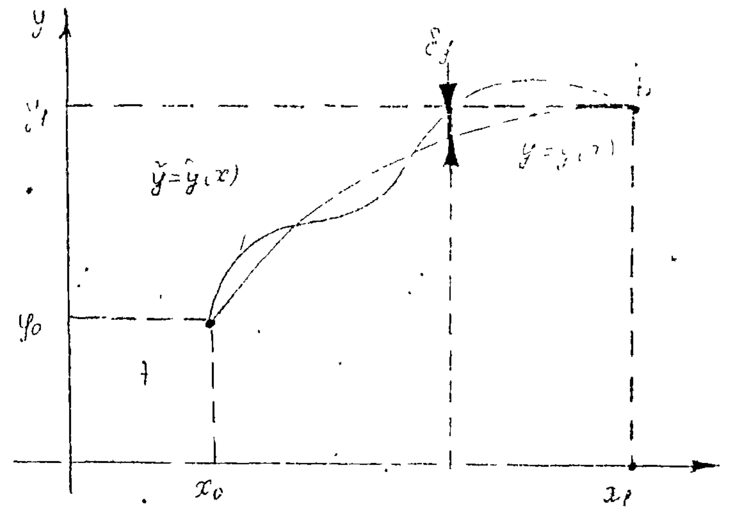


Рис. 9.2

приращение функционала V при переходе от $y(x)$ к $\tilde{y}(x) = y(x) + \delta y$:

$$\Delta V = \int_{x_0}^{x_1} F\,(x,\, \tilde{y},\, \tilde{y}')\, dx - \int_{x_0}^{x_1} F\,(x,\, y,\, y')\, dx$$

и преобразуют его по теореме о конечном приращении (либо то же самое получится при использовании разложения в ряд Тэйлора):

$$\Delta V = \int_{x_0}^{x_1} \left[\widehat{\left(\frac{\partial F}{\partial y} \right)} \delta y + \widehat{\left(\frac{\partial F}{\partial y'} \right)} \delta y' \right] dx \,,$$

где знак ^ указывает, что производные вычислены для промежуточных значений y и y'. Отсюда пользуясь непрерывностью производных $\partial F/\partial y'$ и пренебрегая членами второго порядка малости, получаем вариацию функционала

$$\delta V = \int\limits_{x_0}^{x_1} \left[\frac{\partial F}{\partial y} \delta y + \frac{\partial F}{\partial y'} \delta y' \right] dx \ .$$

В точке экстремума (например, минимума) вариация $\delta V = 0$. В противном случае появилась бы возможность еще уменьшить V: если $\delta V < 0$ при $\delta y > 0$, то непосредственно; если $\delta V > 0$ при вариации $\delta y > 0$, то в силу того, что для величин первого порядка малости δV пропорциональна δy, т.е. сделав $\delta y < 0$, мы бы вариацию δV сделали бы отрицательной, а значит уменьшили V, а это противоречит определению минимума. Таким образом,

$$\int\limits_{x_0}^{x_1} \left[\frac{\partial F}{\partial y} \delta y + \frac{\partial F}{\partial y'} \delta y' \right] dx = 0 \ . \qquad (9.4)$$

Если бы нам удалось во втором члене подинтегрального выражения каким-либо образом заменить $\delta y'$ на δy, тогда наша задача бы упростилась. Ведь при произвольном виде вариации δy единственной возможностью сделать интеграл нулевым было бы приравнять множитель при δy нулю. Для достижения такой цели проинтегрируем по частям второе слагаемое подинтегральной функции:

$$\int\limits_{x_0}^{x_1} \left[\frac{\partial F}{\partial y'} \delta y' \right] dx = \frac{\partial F}{\partial y'} \delta y \Bigg|_{x_0}^{x_1} - \int\limits_{x_0}^{x_1} \frac{d}{dx} \left[\frac{\partial F}{\partial y'} \right] \delta y dx \ .$$

В силу (9.3) первый член в правой части равенства равен нулю. В результате (9.4) преобразуется к виду:

$$\int\limits_{x_0}^{x_1} \left[\frac{\partial F}{\partial y} - \frac{d}{dx} \left[\frac{\partial F}{\partial y'} \right] \right] \delta y dx = 0 \ .$$

Последний интеграл должен равняться нулю при любом выборе δy.

а это возможно лишь в случае тождественного обращения в нуль первого множителя:

$$\frac{\partial F}{\partial y} - \frac{d}{dx}\left[\frac{\partial F}{\partial y'}\right] = 0 . \qquad (9.5)$$

Итак, функция $y(x)$, для которой достигается экстремум функционала V, должна удовлетворять уравнению Эйлера (9.5); или в развернутом виде:

$$\frac{\partial F}{\partial y} - \frac{\partial^2 F}{\partial x \partial y'} - \frac{\partial^2 F}{\partial y \partial y'} y' - \frac{\partial^2 F}{\partial y'^2} y'' = 0 . \qquad (9.6)$$

Интегральные кривые этого дифференциального уравнения 2-го порядка называются экстремалями.

Следует отметить, что вариационные задачи аналитически решаются лишь в исключительно редких случаях. Тем не менее, прикладное значение вариационного исчисления велико. Многие основные принципы физики, механики формулируются в виде вариационных принципов. Принцип оптимальности в динамическом программировании своими корнями обращается к вариационным задачам.

Если подынтегральная функция $F(x,y,y')$ не содержит одного из аргументов, то уравнение Эйлера легко интегрируется. Если F не содержит явно y, то

$$\frac{\partial^2 F}{\partial y \, \partial y'} = 0 \text{ , из чего следует, что } \frac{\partial F}{\partial y'} = c .$$

Если же F не содержит явно x, то из (9.6)

$$F - y' \frac{\partial F}{\partial y'} = c . \qquad (9.7)$$

Вернемся к нашей исходной задаче (9.2), для решения которой воспользуемся уравнением вида (9.7). Подставим в него функцию

$$F = \frac{\sqrt{1 + y'^2}}{\sqrt{y}} ,$$

для которой
$$\frac{\partial F}{\partial y'} = \frac{y'}{\sqrt{y}} \bullet \frac{1}{\sqrt{1+y'^2}}$$

Производя подстановки, после несложных преобразований приходим к дифференциальному уравнению

$$\left(\frac{dy}{dx} \right)^2 + 1 = \frac{c}{y} ,$$

решением которого является кривая, называемая циклоидой, с точкой возврата в точке A. В параметрической форме при единичных параметрах её уравнения записываются:

$x = t - \sin t$; $y = 1 - \cos t$.

Непосредственной подстановкой легко убедиться в справедливости нашего утверждения (мы для краткости только не учли граничные условия, влияющие на параметры циклоиды).

10. УПРАВЛЕНИЕ ТРАНСПОРТНЫМИ ПОТОКАМИ С ПОЗИЦИИ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ

10.1. Задачи теории массового обслуживания. Основные положения и классификация систем массового обслуживания

Работа АСУ, связанных с перевозками, движением транспортных средств, обслуживанием пассажиров, перемещением и складированием грузов и т.п. сталкивается с необходимостью решения задач, описываемых теорией систем массового обслуживания (СМО). Предмет теории массового обслуживания – построение и исследование математических моделей, связывающих условия работы СМО (характер производственных связей, производительность, правила работы, характер потока заявок на обслуживание) с показателями эффективности работы СМО. В качестве таких показателей часто используются: среднее число обслуживаемых заявок, среднее число каналов СМО, среднее время ожидания в очереди, вероятность отказа в обслуживании и т.д.. Приведенный перечень явно говорит о том, что теория СМО опирается на математический аппарат теории вероятностей.

Каждая СМО состоит из какого-то числа обслуживающих единиц, называемых каналами. Это могут быть: транспортные коммуникации, линии связи, билетные кассы, самолеты, бригады обслуживания и др. Всякая СМО предназначена для обслуживания какого-то потока заявок, поступающих в случайные моменты времени. Обслуживание заявки каналом продолжается случайно. Существует время T, в течение которого он не может принять другой заявки, но по истечению которого он освобождается и готов к приему следующих заявок. Случайный характер потока заявок и времени обслуживания приводит к тому, что в какие-то периоды на входе СМО скапливается большое число заявок (они либо становятся в очередь, либо получают отказ и покидают СМО); в другие же периоды СМО может простаивать.

Работа СМО представляет собой случайный процесс с дискретными состояниями и непрерывным временем: состояние СМО меняется скачком (приход заявки, освобождение канала), причем в любой, не фиксированный заранее, момент.

Пример такого процесса - две бригады обслуживают прилетающие самолеты. Для анализа удобно воспользоваться графом состояний (рис. 10.1).

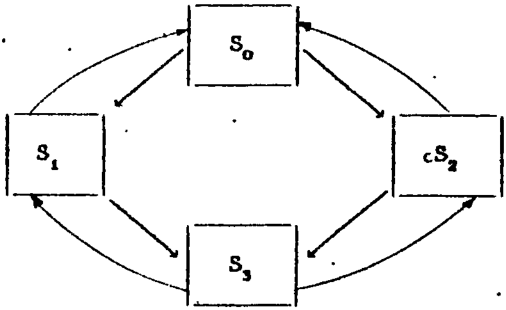


Рис. 10.1

На рис. возможными состояниями являются:

S_0 - обе бригады свободны;

S_1 - первая занята обслуживанием, а вторая нет;

S_2 - наоборот, первая свободна, а вторая занята;

S_3 - обе бригады заняты.

Стрелки на графе означают возможные переходы между состояниями, причем совпадение двух (и более) событий считается невероятным (одновременный прилет двух ВС, одновременное окончание обслуживания).

Системы массового обслуживания делятся на типы. Различают СМО с отказами и СМО с очередью. В СМО с отказами заявка, поступившая в момент, когда все каналы заняты, получает отказ и покидает СМО. Примером такой СМО служит телефонная сеть: когда телефон занят - получаешь отказ, сигнализируемый короткими гудками. В СМО с очередью заявка, пришедшая в момент, когда все каналы заняты, становится в очередь.

Кроме этого СМО делятся на открытые и замкнутые. В открытой СМО характеристики потока заявок не зависят от того, сколько каналов занято. В замкнутой - зависят.

10.2. Марковские процессы и простейший поток событий

Для описания систем массового обслуживания применяются математические потоковые модели, использующие графовые построения. С графами состояний мы уже встречались при изучении сетевых задач линейного программирования, которые рассматривались тогда с точки зрения детерминированных процессов; сейчас нас будут интересовать вероятностные характеристики.

Пусть имеется система массового обслуживания со случайным потоком заявок, который мы также будем называть потоком событий, на входе СМО. Обслуживание заявок производится в продолжение случайного времени. Для наиболее простого описания такой системы (но не обязательно наиболее точного) весьма удобным является предположение о том, что случайный процесс, характеризующий работу системы, является марковским. А именно, для любого момента времени вероятностные характеристики процесса в будущем зависят только от его состояния в данный момент и не зависят от того, когда и как система пришла в это состояние. Например, среднее время обслуживания самолета не зависит от того, сколько времени он ждал своей очереди. Это приводит к тому, что вероятность вовремя обслужить данный самолет не зависит от того, как обслуживались до этого другие самолеты. Такое предположение может не очень согласуется с практикой, зато позволяет достаточно просто получить ответ. (Какая польза от более точных моделей, но с которыми нам не справиться).

Чтобы процесс был марковским, необходимо соблюсти ряд требований математического плана. Если все потоки событий, переводящие систему из состояния в состояние - простейшие, то процесс, протекающий в системе, будет марковским. Простейшим (или пуассоновским) потоком называется поток, если он обладает сразу тремя свойствами: стационарен, ординарен и не имеет последствия. В стационарном потоке все вероятностные характеристики не зависят от времени. В ординарном потоке события появляются поодиночке, совпадение двух событий исключается. Поток без последствия характеризуется независимостью событий друг от друга. Не давая математической формулировки этих свойств, сразу приведем формулу плотности вероятности интервала времени t между соседними событиями в простейшем потоке:

$$f(t) = \lambda e^{-\lambda t}, \quad t > 0 . \qquad (10.1)$$

Плотность вероятности имеет так называемое показательное распределение с параметром λ, являющимся интенсивностью потока. Интенсивность λ – это среднее число событий в единицу времени.

Для нас полезным следствием закона (10.1) является выражение вероятности того, что за Δt произойдет одно событие. Для этого мы попросту должны в (10.1) принять, что $t=\Delta t$, и от плотности перейти к вероятности согласно известной формуле: $P(t) \approx f(t)\Delta t$.

Имеем

$$P(\Delta t) = \lambda e^{-\lambda\Delta t}\,\Delta t .$$

Теперь разложим экспоненту в ряд Тейлора и ограничимся линейными членами по Δt:

$$P(\Delta t) = \lambda e^{-\lambda\Delta t}\Delta t \cong \lambda(1-\lambda\Delta t)\Delta t = \lambda\Delta t - \lambda^2\Delta t^2 \approx \lambda\Delta t.$$

$$\text{Итак} \quad P(\Delta t) \approx \lambda\Delta t . \qquad (10.2)$$

Вероятность события за Δt называется элементом вероятности. Из (10.2) следует, что элемент вероятности совершенно не зависит от того, сколько событий и когда появились ранее.

10.3. Уравнения для вероятностей состояний марковских процессов

Рассмотрим СМО, характеризуемую марковскими процессами. Потоки приходящих и обслуженных заявок являются простейшими, описываемые функцией плотности вероятности вида (10.1). Причем интенсивность входного потока будем по-прежнему обозначать λ, а интенсивность потока обслуживания заявок μ, где μ равна единице, деленной на среднее время обслуживания заявки.

Вернемся к примеру обслуживания самолетов двумя бригадами из п.10.1. Повторим граф состояний, на котором для удобства у стрелок укажем интенсивности соответствующих потоков, переводящих систему из S_i состояния в S_j $(i,j = 0,1,2,3)$.

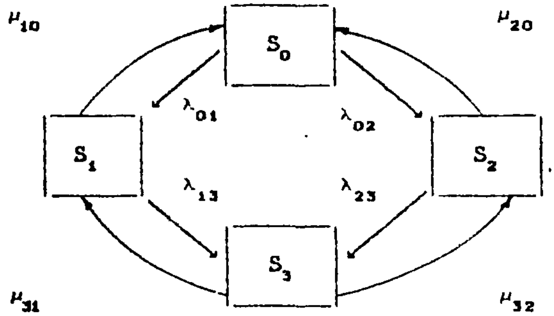


Рис. 10.2

Данный граф относится только к данной задаче из п. 10.1; для других задач граф, очевидно, будет другим. Тем не менее, на частном примере будет видно общее правило описания марковских СМО.

Назовем вероятностью i-го состояния вероятность $p_i(t)$ того, что в момент t система будет находиться в состоянии S_i. Для состояния S_2 – это вероятность $p_2(t)$. Придадим t малое приращение Δt и найдем $p_1(t+\Delta t)$ – вероятность того, что в момент $t+\Delta t$ система будет в состоянии S_1. Это может произойти (рис. 10.2) тремя способами:

1 – в момент t система уже была в состоянии S_1, а за время Δt не вышла из него;

2 – в момент t система была в состоянии S_0, а за Δt перешла в S_1;

3 – аналогичный переход из S_3 в S_1 за Δt.

Найдем вероятность первого способа. Она равна вероятности того, что система находится в S_1 и не будет переходов и в S_0, и в S_3. В силу независимости событий – это произведение вероятностей, для которых вероятности неперехода равны обратным вероятностям перехода в S_0 и S_3, т.е.

$$p_1^1(t+\Delta t) = p_1(t)(1-\mu_{10}\Delta t)(1-\lambda_{13}\Delta t) =$$

$$p_1(t)(1-\mu_{10}\Delta t-\lambda_{13}\Delta t+\mu_{10}\lambda_{13}\Delta t^2)\ .$$

Пренебрегая малыми величинами второго порядка получаем

$$p_1^1(t+\Delta t) = p_1(t)\left[1-\left(\mu_{10}+\lambda_{13}\right)\Delta t\right]\ .$$

Для второго способа вероятность равна произведению вероятности нахождения в S_0 и вероятности перехода в S_1, т.е.

$$p_1^2(t+\Delta t) = p_0(t)\,\lambda_{01}\,\Delta t\ .$$

Для третьего способа аналогично имеем

$$p_1^3(t+\Delta t) = p_3(t)\,\mu_{31}\Delta t\ .$$

Как легко догадаться, данные выражения получены с использованием свойства (10.2).

Складывая вероятности всех способов (по правилу сложения вероятностей событий – хорошо, что они независимы), получаем:

$$p_1(t+\Delta t) = p_1^1(t+\Delta t) + p_1^2(t+\Delta t) + p_1^3(t+\Delta t) =$$

$$= p_1(t)\left[1-\left(\mu_{10}+\lambda_{13}\right)\Delta t\right] + p_0(t)\,\lambda_{01}\Delta t + p_3(t)\mu_{31}\Delta t\ .$$

Раскрываем скобки, группируем члены и делим обе части на Δt:

$$\frac{p_1(t+\Delta t)-p_1(t)}{\Delta t} = \lambda_{01}p_0(t)+\mu_{31}p_3(t)-(\mu_{10}+\lambda_{13})\,p_1(t)\ .$$

Переходя к пределу $\Delta t \longrightarrow 0$ приходим к линейному дифференциальному уравнению

$$\frac{d\,p_1(t)}{d\,t} = \lambda_{01}p_0(t) + \mu_{31}p_3(t) - (\mu_{10} + \lambda_{13})\,p_1(t)\ .$$

Рассуждая аналогично для всех остальных состояний можно записать ещё три дифференциальных уравнения. Однако из-за условия нормировки вероятностей $\Sigma\, p_i = 1$, одно из них будет лишним. Поэтому, произвольно отбрасывая уравнение для S_0, приходим к следующей системе

$$\begin{cases} \dfrac{d\,p_1}{d\,t} = \lambda_{01}p_0 + \mu_{31}p_3 - (\lambda_{13} + \mu_{10})\,p_1 \quad ; \\[2ex] \dfrac{d\,p_2}{d\,t} = \lambda_{02}p_0 + \mu_{32}p_3 - (\lambda_{23} + \mu_{20})\,p_2 \quad ; \\[2ex] \dfrac{d\,p_3}{d\,t} = \lambda_{13}p_1 + \lambda_{23}p_2 - (\mu_{31} + \mu_{32})\,p_3 \quad ; \\[2ex] p_0 + p_1 + p_2 + p_3 = 1 \quad . \end{cases} \tag{10.3}$$

Уравнения (10.3) называются уравнениями Колмогорова. Задавшись начальными условиями, например приняв $p_0(0) = 1$ (остальные вероятности, очевидно, нули), уравнения решаются классическими способами. Эрудированный читатель, возможно, разглядит в них некоторую аналогию с матричными уравнениями Гейзенберга: и те и другие применяются к марковским процессам.

10.4. Работа СМО в стационарном режиме. Финальные или терминальные вероятности

Уравнения вида (10.3) позволяют рассчитывать вероятности $p_i(t)$ как функции времени. Но при одних начальных условиях это одни функции, при других – другие, причем функции суть вероятности. А пользоваться вероятностями, которые меняются во времени, не очень удобно. На практике обычно исследуют стационарные режимы работы СМО, когда вероятности состояний, по истечению некоторого времени после начала работы, приходят к своим установившимся значениям, к тому же, вне зависимости от начальных условий. В этом случае, в пределе $t \longrightarrow \infty$, функции $p_i(t)$ становятся просто некоторыми числами p_i, а иметь дело с числами гораздо проще. Мы оставляем в стороне математические аспекты возможности существования стационарных решений, а также оценки времени t, при котором процесс может считаться стационарным.

Числа p_i являются пределами (если они существуют) функций p_i при $t \longrightarrow \infty$ и называются финальными или терминальными вероятностями состояний. Для них, конечно, справедливо условие нормировки

$$\sum_{i=1}^{n} p_i = 1 \quad , \quad i \in (1; n) \quad .$$

Как же вычислить финальные вероятности? Очень просто. Если вероятности p_1, p_2, ... постоянны (поскольку являются пределами), то их производные равны нулю. Значит, чтобы найти финальные вероятности, нужно все левые части в уравнениях вида (10.3) положить равными нулю и решить систему уже не дифференциальных, а линейных алгебраических уравнений. Или лучше прямо, по графу состояний написать систему алгебраических уравнений. По виду (10.3) легко заметить следующее правило для составления таких уравнений. Если перенести отрицательный член каждого уравнения из правой части в левую, то получим сразу систему, где слева стоит финальная вероятность данного состояния p_i, умноженная на суммарную интенсивность всех потоков, ведущих из данного состояния, а справа - сумма произведений интенсивностей всех потоков, входящих в i-е состояние, на вероятности тех состояний, из которых потоки исходят.

Пользуясь этим правилом, напишем уравнения для финальных вероятностей СМО, граф состояния которой дан на рис. 10.2:

$$\begin{cases} (\lambda_{01} + \lambda_{02})\, p_0 = \mu_{10} p_1 + \mu_{20}\, p_2\ ; \\ (\lambda_{13} + \mu_{10})\, p_1 = \lambda_{01} p_0 + \mu_{31}\, p_3\ ; \\ (\lambda_{23} + \mu_{20})\, p_2 = \lambda_{02} p_0 + \mu_{32}\, p_3\ ; \\ (\mu_{31} + \mu_{32})\, p_3 = \lambda_{13} p_1 + \lambda_{23}\, p_2\ . \end{cases} \qquad (10.4)$$

Система (10.4) линейных однородных уравнений с четырьмя неизвестными p_0, p_1, p_2, p_3 определяет неизвестные с точностью до произвольного множителя (и то если определитель системы равен нулю). Этот множитель находится из нормировочного условия (неоднородного уравнения):

$$p_0 + p_1 + p_2 + p_3 = 1.$$

При этом одно (любое) из уравнений (10.4) можно отбросить (оно вытекает как следствие из остальных).

Как понимать эти финальные вероятности? При $t \longrightarrow \infty$ в системе S устанавливается предельный стационарный режим, в ходе которого система случайным образом меняет свои состояния, но их вероятности уже не зависят от времени, т.е. постоянны. Финальную вероятность состояния S_i можно истолковать как среднее относительное время пребывания системы в этом состоянии. Например, если система S имеет 4 состояния S_0, S_1, S_2, S_3 и их финальные вероятности равны 0,05; 0,2; 0,3; 0,45, то это значит, что в стационарном, предельном режиме система в среднем проводит 0,05 общего времени в состоянии S_0; 0,2 времени – в состоянии S_1 и т.д.

Финальные вероятности характеризуют эксплуатационные и потребительские показатели СМО. Вместе с тем они позволяют давать оценку эффективности её работы. Пусть система S в состоянии S_0 (бригады простаивают) приносит доход в минус 8 условных единиц (т.е. убыток), в состоянии S_1 – доход 4, в состоянии S_2 – доход 5 (бригады неэквивалентные), в состоянии S_3 –доход 10. Тогда в стационарном режиме средний доход в единицу времени будет:

$$W = -8 \cdot 0{,}05 + 4 \cdot 0{,}2 + 5 \cdot 0{,}3 + 10 \cdot 0{,}45 = 6{,}4.$$

Эта оценка может служить отправной точкой для последующей оптимизации системы.

10.5. Схема гибели и размножения

Свой термин схема получила от биологических задач, описывающих изменение численности популяции. В теории массового обслуживания эта схема получила большое распространение ввиду своей универсальности.

Граф состояний для схемы гибели и размножения имеет вид, показанный на рис. 10.3.

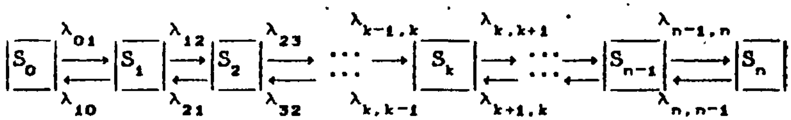


Рис. 10.3

Особенность этого графа в том, что все состояния системы можно вытянуть в одну цепочку, в которой каждое среднее состояние ($S_1, S_2, \ldots, S_{n-1}$) связано прямой и обратной связью с каждым из соседних, а крайние состояния (S_0, S_n) – только с одним соседним состоянием. Не следует путать связь состояний системы с соединением в ней каналов. Граф описывает логику переключения состояний, а не физику соединения блоков в системе.

Предположим, что все потоки событий в системе – простейшие. Пользуясь правилом из предыдущего параграфа, составим систему уравнений для финальных вероятностей. Для S_0 имеем

$$\lambda_{01} p_0 = \lambda_{10} p_1 . \qquad (10.5)$$

Для состояния S_1: $\quad (\lambda_{12} + \lambda_{10})\, p_1 = \lambda_{01} p_0 + \lambda_{21} p_2 .$

С учетом (IO.5) получаем $\lambda_{12}p_1 = \lambda_{21}p_2$ и далее, совершенно аналогично $\lambda_{23}p_2 = \lambda_{32}p_3$.

В итоге получаем систему:

$$\begin{cases} \lambda_{01}\, p_0 = \lambda_{10}\, p_1 \; ; \\ \lambda_{12}\, p_1 = \lambda_{21}\, p_2 \; ; \\ \dots\dots\dots\dots ; \\ \lambda_{k-1,k} p_{k-1} = \lambda_{k,k-1} p_k \; ; \\ \dots\dots\dots\dots ; \\ \lambda_{n-1,n} p_{n-1} = \lambda_{n,n-1} p_n \; . \end{cases} \qquad (IO.6)$$

Система решается последовательной подстановкой. Из первого уравнения (IO.6) имеем

$$p_1 = \frac{\lambda_{01}}{\lambda_{10}}\, p_0 \; . \qquad (IO.7)$$

Из второго, с учетом (IO.7), получаем

$$p_2 = \frac{\lambda_{12}}{\lambda_{21}}\, p_1 = \frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}}\, p_0$$

и вообще, для любого k (от I до n):

$$p_k = \frac{\lambda_{k-1,k} \, \dots \, \lambda_{12}\lambda_{01}}{\lambda_{k,k-1} \, \dots \, \lambda_{21}\lambda_{10}} \; . \qquad (IO.8)$$

Формула легко запоминается. В числителе стоит произведение всех интенсивностей, стоящих у стрелок, ведущих слева направо (с начала и до данного состояния S_k), а в знаменателе – справа налево.

Таким образом, все вероятности $p_1, p_2, \ldots, p_n$ выражены через p_0. Подставим их в нормировочное условие и, вынося за скобку p_0, получим:

$$p_0 \left[1 + \frac{\lambda_{01}}{\lambda_{10}} + \frac{\lambda_{12}\,\lambda_{01}}{\lambda_{21}\,\lambda_{10}} + \ldots + \frac{\lambda_{n-1,n} \ldots \lambda_{12}\,\lambda_{01}}{\lambda_{n,n-1} \ldots \lambda_{21}\,\lambda_{10}} \right] = 1 ,$$

откуда окончательно имеем

$$p_0 = \left[1 + \frac{\lambda_{01}}{\lambda_{10}} + \frac{\lambda_{12}\,\lambda_{01}}{\lambda_{21}\,\lambda_{10}} + \ldots + \frac{\lambda_{n-1,n} \ldots \lambda_{12}\,\lambda_{01}}{\lambda_{n,n-1} \ldots \lambda_{21}\,\lambda_{10}} \right]^{-1} . \qquad (10.9)$$

Итак, если известны все интенсивности λ_{ij}, то по формулам (10.8, 10.9) рассчитываются вероятности всех состояний СМО.

10.6. Формула Литтла

Формула связывает (для стационарного режима) среднее число заявок N_s, находящихся в СМО, и среднее время пребывания заявки τ_s в системе.

Сейчас рассмотрим любую СМО (одноканальную, многоканальную, марковскую, немарковскую, с неограниченной или ограниченной очередью) и связанные с ней два потока событий: поток заявок, приходящих в СМО, и поток заявок, покидающих СМО. Если в системе установился стационарный режим, то среднее число заявок, прибывающих в СМО за единицу времени, равно среднему числу заявок, покидающих её: оба потока имеют одну и ту же интенсивность.

Обозначим: $X(t)$ число заявок, прибывших в СМО до момента t; $Y(t)$ число заявок, покинувших СМО до момента t. Обе функции являются случайными и меняются скачком (увеличиваются на единицу $X(t)$ в моменты приходов заявок и $Y(t)$ – в моменты уходов). Вид функций $X(t)$ и $Y(t)$ показан на рис. 10.4. На рисунке заштрихованными прямоугольниками обозначено пребывание каждой заявки в системе. Каждый прямоугольник имеет единичную

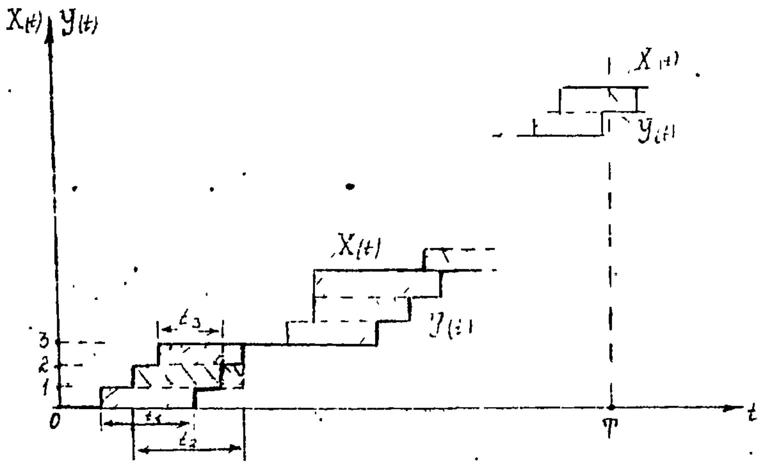


Рис. 10.4

высоту и основание, равное времени t_i нахождения i-й заявки в системе. Из рисунка видно, что X(t) повторяет передние фронты прямоугольников, а Y(t) – не обязательно проходит по задним (сплошная линия). Так, из-за того, что третья заявка обслужилась раньше второй, ступенька Y(t) пересекла прямоугольник второй заявки. Однако, как следует из рисунка, ввиду равновеликости двух маленьких прямоугольников, образованных ломаной линией Y(t), у второй и третьей заявки, замечаем следующее. Для трех первых заявок площадь трех соответствующих прямоугольников равна площади, ограниченной двумя ломаными линиями X(t) и Y(t). Обобщая вывод на все заявки, поступившие за время T, приходим к выводу: площадь, заключенная между двумя ломаными линиями X(t) и Y(t), равна сумме времен пребывания всех заявок за время T. Выразим площадь через интеграл, тогда имеем

$$\int_0^T \left[X(t) - Y(t) \right] dt = \sum_i t_i \, . \qquad (10.10)$$

где сумма распространяется на все заявки, пришедшие за время T.

Теперь рассмотрим подынтегральное выражение. Разность $Z(t) = X(t) - Y(t)$ есть не что иное, как число заявок, находящихся в СМО. Если их нет, то линии $X(t)$ и $Y(t)$ сливаются.

Рассмотрим достаточно большой промежуток времени T и вычислим для него среднее число заявок, находящихся в СМО. Оно равно интегралу от $Z(t)$, делённому на T:

$$N_{\text{з}} = \frac{1}{T}\int_0^T Z(t)dt .$$

С учетом (10.10) можно записать

$$N_{\text{з}} = \frac{1}{T}\sum_i t_i . \tag{10.11}$$

Разделим и умножим правую часть (10.11) на интенсивность λ:

$$N_{\text{з}} = \left[\frac{1}{T\lambda}\sum_i t_i\right]\lambda .$$

Но величина $T\lambda$ есть среднее число заявок, пришедших за время T. Если разделить сумму всех t_i на среднее число заявок, то получим среднее время пребывания заявки в системе $\tau_{\text{з}}$:

$$N_{\text{з}} = \lambda\,\tau_{\text{з}} ,$$

откуда

$$\tau_{\text{з}} = \frac{1}{\lambda}N_{\text{з}} . \tag{10.12}$$

Это и есть формула Литтла: для любой СМО, при любом характере потока заявок, при любом распределении времени обслуживания, при любой дисциплине обслуживания среднее время пребывания заявки в системе равно среднему числу заявок в системе, деленному на интенсивность потока заявок.

Таким же образом выводится вторая формула Литтла, связывающая среднее время пребывания заявки в очереди τ_q и среднее число заявок в очереди N_q:

$$\tau_q = \frac{1}{\lambda} T_q \quad . \tag{10.13}$$

Для вывода достаточно вместо нижней линии на рис. 10.4 взять функцию $U(t)$ – количество заявок, ушедших до момента t не из системы, а из очереди, причем формально считается, что все заявки проходят через очередь (если заявка, пришедшая в систему, не становится в очередь, а сразу обслуживается, можно все же считать, что она становится в очередь, но находится в ней нулевое время).

10.7. СМО n-канальная с отказами

Задача возникла из практических нужд телефонии и была решена датским математиком Эрлангом, поэтому носит его имя. Задача ставится так.

Имеется n параллельных каналов (линий обслуживания), на которые поступает поток заявок с интенсивностью λ. Поток обслуживаний каждым каналом имеет интенсивность μ (величина, обратная среднему времени обслуживания заявки). Найти финальные вероятности состояний СМО, а также характеристики её эффективности:

- абсолютную пропускную способность A, т.е. среднее число заявок, обслуживаемых в единицу времени;

- относительную пропускную способность Q, т.е. среднюю долю пришедших заявок, обслуживаемых системой;

- вероятность отказа $P_{отк}$, т.е. того, что заявка покинет СМО необслуженной;

- среднее число занятых каналов k.

Пронумеруем состояния СМО по числу заявок, находящихся в системе (в данном случае они совпадают с числом занятых каналов):

S_0 – в СМО нет ни одной заявки;

S_1 – в СМО находится одна заявка (один канал занят, остальные свободны);

. . . .

S_k – в СМО находится k заявок (любые k каналов заняты, остальные свободны);

. . . .

S_n – в СМО находится n заявок (все n каналов заняты).

Граф состояний СМО соответствует схеме гибели и размножения (рис. 10.5).

$$\boxed{S_0} \underset{\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_1} \underset{2\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_2} \underset{3\mu}{\overset{\lambda}{\rightleftarrows}} \cdots \underset{k\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_k} \underset{(k+1)\mu}{\overset{\lambda}{\rightleftarrows}} \cdots \underset{n\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_n}$$

Рис. 10.5

Если СМО находится в k-ом состоянии, то в $(k+1)$-е состояние, очевидно, переводит её тот же входной поток с интенсивностью λ (ему все равно, сколько каналов системы уже занято).

С потоком обслуживания иначе. Пусть система находится в S_1, т.е. работает один канал. Он производит μ обслуживаний в единицу времени. Если система в S_2, т.е. работают два канала, каждый с потоком обслуживаний μ, то суммарная интенсивность их потоков обслуживаний равна 2μ. Соответственно $\mu_k = k\,\mu$.

Теперь можно воспользоваться готовыми формулами (10.8), (10.9) для расчета финальных вероятностей.

$$p_0 = \left(1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2\mu^2} + \frac{\lambda^3}{2\cdot 3\mu^3} + \ldots + \frac{\lambda^k}{k!\mu^k} + \ldots + \frac{\lambda^n}{n!\mu^n}\right)^{-1}.$$

Члены разложения $\frac{\lambda}{\mu}$; $\frac{\lambda^2}{2\mu^2}$; ... представляют собой коэффициенты в выражениях для $p_1, p_2, \ldots, p_n$:

$$p_1 = \frac{\lambda}{\mu} p_0 \;;\; p_2 = \frac{\lambda^2}{2\mu^2} p_0 \;;\ldots;\; p_n = \frac{\lambda^n}{n!\mu^n} p_0 .$$

Обозначим $\lambda/\mu = \rho$ и будем называть ρ приведенной интенсивностью потока заявок. Её смысл – среднее число заявок, приходящее за среднее время обслуживания одной заявки. В новых обозначениях получим

$$p_0 = \left[1 + \rho + \frac{\rho^2}{2!} + \ldots + \frac{\rho^n}{n!}\right]^{-1} ; \qquad (10.14)$$

$$p_1 = \rho p_0 ; \quad p_2 = \frac{\rho^2}{2!} p_0 ; \ldots ; p_n = \frac{\rho^n}{n!} p_0 . \qquad (10.15)$$

Таким образом, финальные вероятности найдены, по ним вычисляются характеристики эффективности СМО.

Вероятность отказа $P_{отк}$, как легко догадаться, равна вероятности того, что все n каналов заняты, значит:

$$P_{отк} = p_n = \frac{\rho^n}{n!} p_0 . \qquad (10.16)$$

Отсюда находится относительная пропускная способность – вероятность того, что заявка будет обслужена:

$$Q = 1 - P_{отк} = 1 - \frac{\rho^n}{n!} p_0 . \qquad (10.17)$$

Абсолютная пропускная способность находится из выражения

$$A = \lambda Q = \lambda \left(1 - \frac{\rho^n}{n!} p_0 \right) . \qquad (10.18)$$

По сути это интенсивность потока обслуженных системой заявок. Каждая заявка в среднем обслуживается время, равное $1/\mu$. Значит среднее число k занятых каналов равно:

$$k = A/\mu = \rho \left(1 - \frac{\rho^n}{n!} p_0\right) . \qquad (10.19)$$

При вычислении факториалов весьма полезной может оказаться формула Стирлинга

$$\ln n! = n \ln n - n + 0{,}5 \ln (2\pi n) .$$

10.8. Одноканальная СМО с неограниченной очередью

На практике довольно часто встречаются одноканальные СМО с очередью (билетная касса, взлетно-посадочная полоса).

Пусть имеется одноканальная СМО с очередью, не имеющей ограничений ни по длине, ни по времени ожидания. На вход поступает поток с интенсивностью λ, поток обслуживаний имеет интенсивность μ. Требуется найти финальные вероятности и характеристики эффективности системы: N_s, τ_s, N_q, τ_q.

Что касается абсолютной пропускной способности A и относительной Q, то вычислять их нет надобности: в силу того, что очередь неограничена, каждая заявка рано или поздно будет обслужена, поэтому $A=\lambda$; по той же причине $Q=1$. Неограниченность очереди не означает, что она может бесконечно увеличиваться, ведь тогда не будет установившегося предельного режима.

Для решения состояния системы, как и раньше, будем нумеровать по числу заявок, находящихся в СМО:

S_0 – канал свободен;

S_1 – канал занят, очереди нет;

S_2 – канал занят, одна заявка стоит в очереди;

. . . .

S_k – канал занят, $k-1$ заявок стоят в очереди;

. . . .

Теоретически число состояний бесконечно. Граф состояний имеет вид, показанный на рис. 10.8.

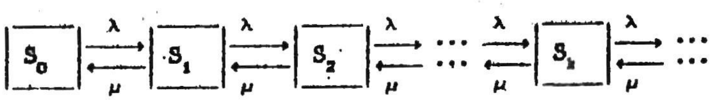


Рис. 10.8

Это схема гибели и размножения, но с бесконечным числом состояний. Имеются ли для такой схемы финальные решения? Ведь в принципе при $t \longrightarrow \infty$ очередь может неограниченно возрастать! Финальные вероятности для такой СМО существуют не всегда, а только когда $\mu > \lambda$ $(\rho < 1)$; т.е., когда обработка заявок происходит в среднем быстрее, чем они поступают, но из-за случайностей иногда образуется очередь.

Финальные вероятности будем искать с использованием формул (10.8), (10.9).

$$p_0 = \left[1 + \lambda/\mu + \left(\lambda/\mu\right)^2 + \ldots + \left(\lambda/\mu\right)^k + \ldots\right]^{-1} =$$

$$= \left[1 + \rho + \rho^2 + \ldots + \rho^k + \ldots\right]^{-1} .$$

В квадратных скобках стоит геометрическая прогрессия со знаменателем ρ . При $\rho<1$ ряд сходится, что косвенно подтверждает условие существования финальных вероятностей. Сумма ряда равна $1/(1-\rho)$. Следовательно

$$p_0 = 1 - \rho \quad . \tag{10.20}$$

Остальные вероятности находятся по формулам:

$$p_1 = \rho p_0 \ ; \quad p_2 = \rho^2 p_0 \ , \ldots , \ p_k = \rho^k p_0 \ ,$$

откуда с учетом (10.20) имеем окончательно:

$$p_1 = \rho(1-\rho); \quad p_2 = \rho^2(1-\rho); \ \ldots \ ; \ p_k = \rho^k(1-\rho); \ \ldots \tag{10.21}$$

Как видно, вероятности $p_0, p_1, \ldots$ образуют геометрическую прогрессию со знаменателем ρ $(\rho<1)$. Максимальная из них p_0 – вероятность того, что канал будет вообще свободен (ведь $\mu>\lambda$).

Найдем среднее число заявок в СМО N_s. Случайная величина Z – число заявок в системе – имеет возможные значения 0,1,2,... с вероятностями $p_0, p_1, p_2, \ldots$ Её математическое ожидание:

$$N_s = 0 \cdot p_0 + 1 \cdot p_1 + 2 \cdot p_2 + \ldots = \sum_{k=1}^{\infty} kp_k ,$$

где суммирование начинается с единицы, поскольку нулевой член равен нулю.

Подставим в эту формулу выражения (10.21) для p_k:

$$N_s = \sum_{k=1}^{\infty} k\rho^k (1-\rho) = \rho(1-\rho) \sum_{k=1}^{\infty} k\rho^{k-1} .$$

Замечаем, что выражение под знаком суммы есть производная по ρ от ρ^k; значит,

$$N_s = \rho(1-\rho) \sum_{k=1}^{\infty} \frac{d}{d\rho} \rho^k .$$

Поскольку производная от суммы функций равна сумме производных, меняем местами операции суммирования и дифференцирования:

$$N_s = \rho(1-\rho) \frac{d}{d\rho} \sum_{k=1}^{\infty} \rho^k .$$

но данная сумма нечто иное, как сумма бесконечно убывающей геометрической прогрессии с первым членом ρ и знаменателем ρ. Эта сумма равна $\rho/(1-\rho)$, а её производная $1/(1-\rho)^2$. Делая соответствующую подстановку, получаем

$$N_s = \frac{\rho}{1-\rho} . \qquad (10.22)$$

Среднее время пребывания заявки в системе получаем по формуле Литтла

$$\tau_в = \frac{\rho}{\lambda\,(1-\rho)} \,. \tag{10.23}$$

Найдем среднее число заявок в очереди N_q. Будем рассуждать так: число заявок в очереди равно числу заявок в системе минус число заявок, находящихся под обслуживанием. То же самое справедливо и для их средних значений. Число заявок под обслуживанием может быть либо нулем (канал свободен), либо единицей (если он занят). Её среднее значение (а, точнее математическое ожидание) по определению :

$$N_{об} = 0 \cdot p_0 + 1(1-p_0) = 1 - p_0 = \rho \,.$$

Следовательно, среднее число заявок под обслуживанием равно

$$N_{об} = \rho \,, \tag{10.24}$$

отсюда $$N_q = N_в - \rho = \frac{\rho}{1-\rho} - \rho$$

и окончательно

$$N_q = \frac{\rho^2}{1-\rho} = \rho N_в \,. \tag{10.25}$$

По формуле Литтла найдем среднее время пребывания заявки в очереди:

$$\tau_q = \frac{\rho^2}{\lambda\,(1-\rho)} \,. \tag{10.26}$$

10.9. СМО n-канальная с неограниченной очередью

Схема СМО с n каналами и очередью очень часто подходит под практические случаи: несколько бригад (равноценных) обслуживают самолеты, терминалы, билетные кассы и т.п. Задача решается аналогично предыдущей. Каналы предполагаются совершенно идентичными.

Обозначим:

S_0 – в СМО все каналы свободны (заявок нет);

S_1 – занят один канал, остальные свободны;
S_2 – занято два (любых) канала, остальные свободны;
. . . .
S_k – занято k каналов;
. . . .
S_{n+1} – заняты все n каналов, одна заявка стоит в очереди;
. . . .
S_{n+r} – заняты все n каналов, r заявок стоят в очереди.
Граф состояний показан на рис. 10.7.

$$\boxed{S_0} \underset{\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_1} \underset{2\mu}{\overset{\lambda}{\rightleftarrows}} \cdots \underset{k\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_k} \underset{(k+1)\mu}{\overset{\lambda}{\rightleftarrows}} \cdots \underset{n\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_n} \underset{n\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_{n+1}} \underset{n\mu}{\overset{\lambda}{\rightleftarrows}} \cdots \underset{n\mu}{\overset{\lambda}{\rightleftarrows}} \boxed{S_{n+r}} \underset{n\mu}{\overset{\lambda}{\rightleftarrows}} \cdots$$

Рис. 10.7

До n-го состояния мы имеем схему n-канальной СМО с отказами. Далее, когда возникает очерёдь, поток обслуживания определяется суммой интенсивностей μ всех n каналов (поток максимально возможный).

Применяя формулы из схемы гибели и размножения, находим финальные вероятности:

$$p_0=\left\{1+\frac{\lambda}{\mu}+\ldots+\frac{\lambda^n}{n!\mu^n}+\frac{\lambda^{n+1}}{n!n\mu^{n+1}}+\frac{\lambda^{n+1}}{n!n^2\mu^{n+2}}+\ldots+\frac{\lambda^{n+r}}{n!n^r\mu^{n+r}}+\ldots\right\}^{-1}=$$

$$=\left[1+\rho+\ldots+\frac{\rho^n}{n!}+\frac{\rho^n}{nn!}\left(\rho+\frac{\rho^2}{n}+\ldots+\frac{\rho^r}{n^{r-1}}+\ldots\right)\right]^{-1}.$$

В круглых скобках стоит геометрическая прогрессия с первым членом ρ и знаменателем ρ/n, её сумма при $\rho/n<1$ (это значит, что $n\mu>\lambda$) равна

$$\frac{\rho}{1-\dfrac{\rho}{n}}=\frac{n\rho}{n-\rho}.$$

В результате получаем:

$$\begin{cases} p_0 + \left[1 + \dfrac{\rho}{1!} + \dfrac{\rho^2}{2!} + \ldots + \dfrac{\rho^n}{n!} + \dfrac{\rho^{n+1}}{n!\,(n-\rho)}\right]^{-1}; \\ p_1 = \dfrac{\rho}{1!}\,p_0\,,\ p_2 = \dfrac{\rho^2}{2!}\,p_0,\ \ldots\ ,\ p_n = \dfrac{\rho^n}{n!}\,p_0\,; \\ p_{n+1} = \dfrac{\rho^{n+1}}{n \cdot n!}\,p_0,\ \ldots\ ,\ p_{n+r} = \dfrac{\rho^{n+r}}{n^r \cdot n!}\,p_0\,,\ \ldots\ , \end{cases} \tag{10.27}$$

Условие $n\mu>\lambda$ при нормальной организации работы системы обычно выполняется. Если один канал не справляется, то ставится второй, ... , n-й. Конечно, могут возникнуть технические или иные ограничения. Поэтому разрабатываются специальные конструкции, технические стандарты, как в случае сотовой телефонной связи. Их изучение – удел других дисциплин.

Среднее число занятых каналов k, как и в п.10.7, равно A/μ. А поскольку, следуя рассуждениям из п. 10.8, $A = \lambda$, то в итоге

$$k = \lambda/\mu = \rho\,, \tag{10.28}$$

что вообще справедливо для любой СМО с неограниченной очередью.

Среднее число заявок в очереди:

$$N_q = \sum_{r=1}^{\infty} r p_{n+r}\,.$$

Подставляя в это выражение формулы (10.27) и выполняя преобразования по образцу задачи из предыдущего параграфа (с дифференцированием ряда), получаем

$$N_q = \frac{\rho^{n+1}\,p_0}{n \cdot n!\,(1-\rho/n)^2} \tag{10.29}$$

При $n=1$ формула (10.29) приводится к виду (10.25).

Прибавляя к N_q число заявок под обслуживанием, равное среднему числу занятых каналов $k = \rho$, получаем

$$N_s = N_q + \rho \quad . \tag{10.30}$$

Среднее время пребывания заявки в очереди и в системе, согласно формулам Литтла, равно:

$$\tau_q = \frac{N_q}{\lambda} \; ; \quad \tau_s = \frac{N_s}{\lambda} \; . \tag{10.31}$$

Все рассмотренные нами модели СМО строились из предположения марковости систем и простейшего потока заявок. Такое допущение зачастую можно принимать лишь как некоторое приближение, позволяющее относительно просто, но подчас грубо, рассчитывать характеристики реальных систем. В специальных руководствах читатель может найти несравненно больше различных моделей СМО, из которых можно выбрать ту, которая лучше всего описывает реальную ситуацию.

ЛИТЕРАТУРА

1. Глушков В.М. Введение в АСУ. Киев. Техника,1972.
2. Вентцель Е.С. Исследование операций. Задачи, принципы, Методология. М.; Наука, 1980.
3. ГрешиловА.А. Прикладные задачи математического программирования. М.; МГТУ, 1990.
4. Сборник задач по высшей математике. Ч.3. Специальные курсы. Под ред. Ефимова А.В. М.; Наука, 1984.
5. Кейн В.М., Красов А.И.,Федоров С.М. Теория управления в гражданской авиации. Ч. 2. Л.; ОЛАГА, 1978.
6. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ. М.; Мир, 1991.
7. Визо Манс. Microsoft Access 2.0. Под ред. Каратыгина [illegible]

ОГЛАВЛЕНИЕ

Редактор и технический редактор Т.Н.Заграничная
Подписано к печати 7.04.99г.Формат бумаги 60x90 1/16.Тираж 300.
Заказ 406.Усл.печ.л. 9,5. Уч.-изд.л. 9,5. С 24.Из плана 1998г.
Тип.Академии ГА.196210, С.-Петербург,ул.Пилотов,дом 38.