

А.А. Грешилов

Математические методы принятия решений

*Допущено Учебно-методическим объединением вузов
по университетскому политехническому образованию
в качестве учебного пособия для студентов высших
учебных заведений, обучающихся
по машиностроительным специальностям*

2-е издание, исправленное и дополненное

 **ИЗДАТЕЛЬСТВО**
МГТУ им. Н.Э. БАУМАНА

Москва
2014

УДК 519.24+519.81

ББК 22.18

Г81

Рецензент

д-р физ.-мат. наук, профессор *А.Ф. Кушнир*

Грепилов А. А.

Г81 Математические методы принятия решений : учеб. пособие (с расчетными программами на оптическом диске) / А. А. Грепилов. — 2-е изд., испр. и доп. — М. : Изд-во МГТУ им. Н. Э. Баумана, 2014.— 647, [1] с. : ил.

ISBN 978-5-7038-3910-2

Изложены методы решений задач математического программирования и статистических задач принятия решений (задачи распознавания образов). Рассмотрены алгоритмы, позволяющие учитывать влияние погрешностей всех случайных величин, фигурирующих в задаче (конфлюэнтный анализ).

Рассматриваются реальные примеры, например, идентификации землетрясений и слабых взрывов по результатам сейсмических наблюдений, идентификации летательных аппаратов, задачи о назначениях, о максимизации выпуска продукции и т. п.

К пособию прилагается оптический диск с обучающими программными продуктами.

Учебное пособие создано на основе лекций и практических занятий для студентов МГТУ им. Н.Э. Баумана.

Для студентов технических вузов, специалистов, занимающихся задачами принятия решений, а также слушателей курсов системы дополнительного профессионального образования, изучающих подобные задачи.

УДК 519.24+519.81

ББК 22.18

© А.А. Грепилов, 2014

© Оформление. Издательство

МГТУ им. Н. Э. Баумана, 2014

ISBN 978-5-7038-3910-2

Предисловие

Математика как наука с самого зарождения является инструментом в процессе поиска истины, и потому можно считать, что любые математические операции, даже самые простые, являются математическими методами принятия решений. Однако в приложениях термин «принятие решений» имеет более конкретный смысл. Так, в разделе математической статистики, называемом «принятие решений», изучают способы принятия или не принятия некоторой основной гипотезы при наличии конкурирующей гипотезы с учетом функции потерь. Теория принятия решений развивает методы математической статистики — методы проверки гипотез. Различные величины потерь при выборе разных гипотез приводят к результатам, отличным от тех, которые получены методами статистической проверки гипотез. Выбор менее вероятной гипотезы может оказаться более предпочтительным, если потери в случае ошибочности такого выбора окажутся меньше потерь, вызванных ошибочностью выбора более вероятной конкурирующей гипотезы. Такие задачи называют статистическими задачами принятия решений. Для решения этих задач необходимо найти минимальное значение функции риска на множестве возможных исходов, т. е. решить задачу отыскания условного экстремума. Как правило, для этих задач можно выделить цель и указать условия, т. е. ограничения, при которых они должны быть решены. Подобными задачами занимаются в разделе математики «математическое программирование», который, в свою очередь, является частью раздела «исследование операций».

В настоящее время под принятием решений понимается особый процесс человеческой деятельности, направленный на выбор наилучшего варианта действий*.

В данной книге термином «принятие решений» объединены методы решений задач математического программирования и статистических задач принятия решений.

В задачах раздела «принятие решений» необходимо найти оптимум некоторого функционала в детерминированной или стохас-

* *Ларичев О. И.* Теория и методы принятия решений, а также Хроника событий в Волшебных странах: Учебник. — 2-е изд., перераб. и доп. — М.: Логос, 2003. — 392 с.

тической форме. Следует отметить две особенности. Во-первых, математические методы принятия решений для задач, связанных с различными направлениями деятельности человека, начинают взаимное проникновение друг в друга, например, оптимизационные задачи управления при переходе от непрерывных переменных к дискретным становятся задачами математического (линейного) программирования, оценка разделяющей функции в статистических методах принятия решений может проводиться с помощью процедур линейного или квадратичного программирования и т. д. Во-вторых, исходные числовые данные как результат измерений или наблюдений в задачах принятия решений для реальных ситуаций не являются детерминированными, а чаще являются случайными величинами с известными или неизвестными законами распределения, поэтому последующая обработка данных требует применения методов математической статистики, теории нечетких множеств или теории возможностей.

Несмотря на то, что по методам принятия решений существует достаточно много литературы, книги чаще ориентированы на математиков либо на студентов экономических вузов или радиоинженеров, изучавших дифференциальное и интегральное исчисления. В большинстве книг не учитывается случайный характер участвующих в расчетах величин. Математические методы принятия решений часто излагаются по «отраслевому» принципу: статистические методы принятия решений — в радиолокации, линейное программирование — в экономике и т. п. Пока нет книги, которая была бы доступна инженеру, владеющему математикой в объеме программы технического вуза, и в которой были бы рассмотрены математические методы принятия решений для задач, относящихся к различным разделам математики. Восполнить этот пробел предназначено настоящее учебное пособие. В нем рассмотрены алгоритмы, позволяющие учитывать погрешности всех случайных величин, фигурирующих в задаче (конфлюэнтный анализ). Данное пособие рассчитано на самый широкий круг пользователей персональных компьютеров. Для понимания всех глав книги достаточно знаний математики в объеме первых двух курсов технических вузов, а для понимания алгоритмов решения задач и этого не требуется. По ходу изложения теоретических основ решения задач даны определения, необходимые для понимания излагаемых алгоритмов: квадратичных форм, скалярного произведения векторов, частных производных, градиента, производной по направлению, линейной зависимости векторов и т. п.

За последние годы произошел грандиозный скачок в развитии персональных компьютеров и их программного обеспечения. Создано много программных продуктов и целых систем, реализующих методы принятия решений. Развитие вычислительной техники и широкое распространение персональных компьютеров дает возможность проверять принимаемые решения, предварительно построив модель явления. В свою очередь, такой подход требует разработки алгоритмов и методов решения задач, в частности производственных, доступных самому широкому кругу пользователей.

Свободный доступ к программному обеспечению позволяет решать конкретные задачи в интерактивном режиме. При применении интерактивных процедур мы можем проводить исследования множества допустимых альтернатив, изменяя как условия-ограничения задач, так и параметры целевых функций для выбора оптимального решения. Интерактивные процедуры характеризуются поочередной сменой этапов вычислений, анализа и принятия решений. На каждой итерации лицо, принимающее решение (ЛПР), для дальнейшего исследования может генерировать новые условия задачи.

Обратная связь между человеком и моделью позволит ЛПР получить новые сведения о стоящей перед ним проблеме и более полно оценить диапазон возможностей, задаваемый множеством допустимых решений, а также взаимозаменяемость критериев. Все это должно позволить ЛПР лучше понять, как искать более удачные решения и как распознать окончательное решение.

Сущность интерактивного подхода состоит в том, чтобы облегчить человеку участие в процессе поиска решения на основе корректировки этого процесса. Интерактивные процедуры дают возможность для эффективного разделения труда: компьютер выполняет то, что он делает лучше всего (обрабатывает данные), ЛПР, получив новую информацию, разрабатывает методы для получения лучшего решения.

Тем не менее главная роль остается за человеком, ибо правильная постановка задачи, грамотная формулировка модели являются залогом успешного результата, который невозможно получить без понимания построения алгоритмов решения задач математического программирования и знания статистических методов принятия решений и их особенностей.

В задачах принятия решений важное значение имеет возможность исследования поведения решения задачи при изменении параметров целевой (целевых) функций и условий-ограничений. Ча-

сто в подобных случаях решают серию задач с близкими значениями параметров. Можно учесть неопределенность исходных данных с помощью параметрического и многокритериального программирования. В настоящей книге наряду с классическими задачами математического программирования рассмотрены многокритериальные задачи линейного программирования и задачи параметрического программирования. Для задач линейного программирования получены условия устойчивости оптимального решения при изменении исходных данных.

В части I книги «Математическое программирование» кроме классических задач математического программирования рассмотрены сетевые задачи, а также динамическое программирование как один из способов решения задач на условный экстремум и теория игр с нулевой суммой, использующая линейное программирование для получения решения.

Статистические задачи принятия решений в настоящей книге (часть II) рассматриваются на примере решения задач распознавания образов. Алгоритмы распознавания образов по совокупности признаков различаются этапностью принятия решений, степенью и характером учета статистики признаков, помех, сигналов.

К решению задач распознавания образов привлекают также ряд математических теорий и методов, развитие которых связано с появлением экспертных систем: теорию нечетких множеств и теорию возможностей, теорию игр и другие математические методы.

В задачах распознавания образов и принятия решений важно, как учтены (насколько строго) неопределенности исходных данных. Использование усредненных величин ведет к смещенным оценкам основных показателей, определяющих решение и, как следствие, к неверным практическим выводам. Поэтому здесь особое внимание уделено строгому учету погрешностей измерений вектора признаков объекта.

В отличие от традиционных статистических методов распознавания образов в настоящей книге разработаны и используются методы, которые позволяют учесть погрешности наблюдаемых значений координат вектора признаков объекта, получить несмещенные точечные и интервальные оценки функций условных плотностей распределения вероятностей признаков и разделяющих функций, оценить «истинные» координаты вектора признаков, по которому ведется идентификация объектов, и включить эти данные в процедуру принятия решений. Это приводит к принципиально новым решениям. Так, интуитивно ясно, что не всегда коли-

чество информации достаточно для принятия надежного решения. Тем не менее в задачах принятия решений при фиксированном объеме выборки решение принимается всегда, независимо от объема выборки. В книге это противоречие предлагается разрешить путем нахождения интервальной оценки разделяющей функции, определяющей область невозможности принятия решения (нулевую зону).

Изложение теоретического материала доступно лицам, владеющим математикой в объеме программы технического вуза, и сопровождается рассмотрением реальных примеров, как, например, идентификация землетрясений и слабых взрывов по результатам сейсмических наблюдений, идентификация летательных аппаратов, задачи о назначениях, о максимизации выпуска продукции и т. п.

Книга состоит из двух частей. В части I (главы 1–5) изложены задачи математического программирования и родственные им.

В главе 1 изложены история становления математического программирования, общая постановка задач математического программирования, основные особенности задач на условный экстремум, решаемых в этом разделе математики. Приведены графические методы решения задач математического программирования, методы безусловной оптимизации.

В главе 2 рассмотрен наиболее хорошо изученный раздел математического программирования — линейное программирование. Здесь же рассмотрены и частные случаи задач линейного программирования — транспортные задачи и задачи целочисленного линейного программирования. Рассмотрены дробно-линейное программирование и анализ устойчивости оптимального решения задачи линейного программирования при неопределенности параметров задачи.

В главе 3 описаны сетевые (поточковые) задачи (задачи о кратчайших путях, о максимальных потоках, о многопродуктовых потоках, задача коммивояжера), показаны их особенности и преимущества по сравнению с задачами линейного программирования.

В главе 4 рассмотрены основы динамического программирования и элементы теории игр с нулевой суммой.

В главе 5 представлены некоторые направления развития методов решения задач математического программирования: параметрическое программирование; решение многопродуктовых сетевых задач; один из эвристических методов решения сетевых задач; метод штрафных (барьерных) функций; метод проекции

градиента; целевое программирование; параметрическое программирование; многокритериальное линейное программирование.

В части II (главы 6–10) изложены статистические методы принятия решений.

В главе 6 приведен анализ традиционных методов статистических задач решения; рассмотрены как стохастический, так и детерминистский подходы, а также задачи принятия решения с фиксированным объемом выборки и последовательная решающая модель. Сформулирована математическая постановка задачи учета неопределенностей всех исходных данных для принятия решений.

В главе 7 изложены методы регрессионного и конфлюэнтного анализа, служащие основой статистических методов принятия решений. Отмечено, что нестрогий учет погрешностей исходных данных приводит к регрессионному парадоксу, когда по одним и тем же исходным данным можно получить разные решения, поменяв местами зависимые и независимые переменные. Для различных моделей рассмотрены методы получения точечных и интервальных оценок, учитывающие одновременно погрешности всех исходных данных.

В главе 8 описаны способы учета погрешностей вектора признаков в статистических задачах решения с фиксированным объемом выборки, когда вид функций обобщенных условных плотностей распределения вероятностей известен априори, но не известны оценки их параметров; приводятся алгоритмы получения несмещенных точечной и интервальных оценок разделяющей функции, а также оценок «истинных» значений измеренного вектора признаков, по которым принимается решение. Рассмотрены методы получения оценок в некорректных задачах и показаны различия в решениях, полученных традиционным и предлагаемым методами.

В главе 9 рассмотрена задача распознавания образов, когда обобщенные условные плотности распределения вероятностей не заданы, но предполагается известным вид разделяющих функций, параметры которых подлежат определению с учетом погрешности наблюдаемых значений признаков.

В главе 10 рассмотрены различные методы построения прогнозов: разностные методы, экспоненциальное сглаживание и сглаживание с помощью скользящей средней, байесовские прогнозы, прогнозирование сезонных явлений, методы диагностической проверки моделей, оценки и коррекции ошибки прогнозов.

Первое издание книги было в 2006 г. Второе издание дополнено программными продуктами, инструкциями пользователю и математическим словарем.

К книге прилагается оптический диск, содержащий программные продукты и их описания (инструкции пользователю). Описания (инструкции пользователю) содержатся и в приложении книги.

Программный продукт «Регрессия» предназначен для построения линий регрессии Y на X , X на Y и ортогональной регрессии, а также решение задач методом наименьших квадратов. В программном продукте предусмотрен вывод всей процедуры (всех этапов) решения задачи, что помогает студенту разобраться в решении задачи. С помощью программных продуктов, функционирующих в интерактивном режиме, можно решать задачи, входящие в понятие математического программирования: симплекс-метод, задача о выпуске продукции, транспортная задача, задачи целочисленного линейного программирования, динамического программирования (задача о рюкзаке), сетевые задачи (задача о максимальном потоке). В случае ошибки студенту высвечивается информация о неверно выбранном значении параметра, и процесс решения останавливается до выбора правильного шага.

Программные продукты позволяют получить решение задачи за одну итерацию (без участия студента в управлении решением) и поэтапно, выполняя каждую итерацию последовательно. Последний вариант позволяет студенту вникнуть в алгоритм решения задачи, а первый вселяет в студента уверенность, поскольку задача решается. В программном продукте предусмотрен вывод всей процедуры (всех этапов) решения задачи, что помогает студенту разобраться в решении задачи.

Симплекс-метод содержится в двух программных продуктах: `LinProg.exe` и в `MatProg.exe`. В `MatProg` кроме симплекс-метода содержатся программы для решения транспортной задачи, задачи о максимальном потоке и минимальном разрезе, сетевых задач (сеть наименьшей стоимости, многополюсная кратчайшая сеть), поиска на графах И-ИЛИ (решатель) и матричные антагонистические игры.

Задачи динамического программирования решаются с помощью программного продукта `dionprog.exe` и пакета `Matlab`, вызывая функцию `dinprog.m`. В качестве примера рассмотрена задача «о рюкзаке». С помощью пакета `Matlab`, вызывая функцию `celprog.m`, решаются задачи целочисленного линейного программирования. В качестве примера решается та же задача «о рюкзаке».

Для сетевых задач и матричных игр инструкция в подменю «справка».

В книгу включен краткий математический словарь, облегчающий читателю усвоить прочитанное.

Таким образом, излагаемый материал представляет интерес для различных областей знаний и практических приложений. В задачах математического программирования всегда рассматривают функции многих переменных. Поэтому решение задачи об экстремуме имеет несколько координат, а рассчитываемые величины являются векторами. Чтобы познакомить читателя с разнообразными формами записи решения в линейном программировании, в процессе изложения материала приведены различные формы симплекс-таблиц.

В написании программ и инструкций пользователю принимали участие к.т.н., доцент кафедры ИУ-7 МГТУ им. Н.Э. Баумана З.Н. Русакова (программный продукт MatProg.exe), к.т.н. А.Л. Лебедев (программный продукт LinProg.exe) и выпускник кафедры ФН-1 МГТУ им. Н.Э. Баумана А.А. Созинов (программный продукт celprog.exe).

Настоящая книга предназначена студентам технических и экономических специальностей вузов, а также специалистам, занимающимся задачами принятия решений. Она будет полезна слушателям курсов дополнительного профессионального образования, изучающим подобные задачи.

Автор благодарит всех коллег, способствовавших выходу этой книги, в частности президента МГТУ им. Н. Э. Баумана И. Б. Федорова, руководителя НУК ИБМ МГТУ им. Н.Э. Баумана доктора техн. наук, доктора экон. наук, профессора И.Н. Омельченко за оказанную помощь в подготовке 2-го издания книги и многих других, а также рецензентов, высказавших полезные замечания.

Часть I

Математическое программирование

Нужно много учиться, чтобы
немного знать.

Шарль Монтескье

ВВЕДЕНИЕ В МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

§ 1.1. Общие положения математического программирования

Деятельность отдельных людей и коллективов, как правило, связана с выбором таких решений, которые позволили бы получить некие оптимальные результаты: затратить минимум средств на питание семьи, достичь максимальную прибыль предприятия, добиться наилучших показателей в спорте и т. д. Однако в каждой конкретной ситуации необходимо учитывать реальные условия, налагаемые на решение данной задачи. Например, при расчете затрат на питание следует учитывать стоимость тех продуктов и такое их количество, чтобы организм получил необходимые ему жиры, белки, углеводы и т. п.; достигнуть максимальной прибыли предприятия не удастся, если не учитывать реальные запасы сырья, его стоимость и целый ряд других факторов; для достижения наилучших показателей в спорте необходимо квалифицированно организовать тренировку спортсменов, оптимально использовать имеющиеся технические средства и площадки, правильно сформировать команду. Чтобы что-то рассчитать, необходимо формализовать задачу, т. е. составить математическую модель изучаемого явления, поскольку математические методы можно применять лишь к математическим моделям. Результаты исследований математических моделей представляют практический интерес только тогда, когда эти модели адекватно отображают реальные ситуации и достаточно совершенны.

Приведенные примеры позволяют выделить в описывающих их моделях цель и сформулировать *целевую функцию (оптимизируемый критерий)*: минимум затрат, максимум прибыли, наилучшие спортивные достижения, — и *условия ограничения*: необходимое количество жиров, белков и углеводов; запасы сырья и его стоимость; возможности спортивных площадок и различные варианты состава команд.

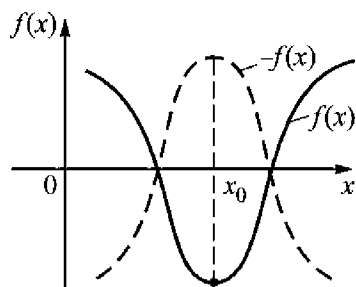


Рис. 1.1. Положения максимума и минимума функций $-f(x)$ и $f(x)$

Задача нахождения минимума некоторой функции $f(x)$ эквивалентна задаче отыскания максимума той же функции, взятой со знаком минус (рис. 1.1), и наоборот. Поскольку принципиальных отличий в нахождении минимума и максимума не существует, будем говорить об оптимальных (лат. *optimum* — наилучшее), или экстремальных значениях целевых функций.

Итак, сформулированные в реальных задачах требования могут быть выражены количественными критериями и записаны в виде математических выражений, т. е. можно записать условия задачи математически и получить так называемую математическую постановку задачи. Процесс составления математической задачи и последующего ее решения достаточно сложен. Этот процесс можно представить в виде следующих этапов.

1. *Изучение объекта.* Анализ особенностей функционирования объекта; определение факторов, оказывающих на это влияние (их количество и степени влияния); изучение характеристик объекта при различных условиях; выбор оптимизируемого критерия (целевой функции).

2. *Описательное моделирование.* Установление и фиксация основных связей и зависимостей между характеристиками процесса или явления согласно оптимизируемому критерию.

3. *Математическое моделирование.* Запись описательной модели математическими формулами. Все условия записывают в виде соответствующей системы равенств и неравенств, а критерий оптимизации — в виде функции. После того как задача записана в математической форме, ее конкретная постановка перестает нас интересовать до проведения содержательного анализа полученного решения. Дело в том, что различные по своему физическому смыслу задачи часто можно свести к одной и той же формальной математической записи.

4. *Выбор или создание метода решения.* Исходя из полученной математической записи задачи выбирают либо известный метод решения, либо некую модификацию известного метода, либо разрабатывают новый метод решения. *Допустимым решением* называют такой набор значений искомых величин (переменных), который удовлетворяет поставленным условиям-ограничениям задачи.

Решением задачи будет то решение из множества допустимых решений, при котором целевая функция достигает своего наибольшего (наименьшего) значения.

5. *Выбор или написание программы для решения задачи на ЭВМ.* Задачи, содержащие целевую функцию и условия-ограничения и описывающие поведение реальных объектов, как правило, имеют много переменных и много зависимостей (*уравнений связи*) между ними. Поэтому в разумные сроки они могут быть решены только с помощью ЭВМ. Программа для ЭВМ реализует выбранный метод решения задачи.

6. *Решение задачи на ЭВМ.* Необходимую информацию для решения задачи вводят в память ЭВМ вместе с программой. В соответствии с программой компьютер обрабатывает введенную числовую информацию, получает решение и выдает его пользователю в заданной им форме.

7. *Анализ полученного решения.* Анализ решения бывает формальным и содержательным. При *формальном* (математическом) анализе проверяют соответствие полученного решения построенной математической модели, т. е. проверяют, правильно ли введены исходные данные, правильно ли функционируют программа, компьютер и т. д. При *содержательном* анализе проверяют соответствие полученного решения тому реальному объекту, который моделировали. В результате содержательного анализа в модель (словесную и математическую) могут быть внесены изменения, затем весь рассмотренный процесс повторяют.

8. *Анализ устойчивости решения.* Аналитически или с помощью численных методов исследуют поведение решения при небольших (в пределах возможных погрешностей или неопределенностей) изменениях исходных данных.

Только после полного завершения анализа модель можно использовать для расчета. Чтобы подчеркнуть важность содержательного анализа, приведем следующий пример. Когда впервые решали *задачу о питании*, то в качестве фактора оптимизации брали минимум затрат, а в условие-ограничение включили только требование по калорийности пищи. Решение задачи было таковым: питаться следует *уксусом*, который входит в состав всевозможных продуктов питания, тогда будет и калорийность обеспечена, и стоимость минимальна. Построим математическую модель задачи о питании.

Предположим, что в рацион семьи входят три различных питательных вещества и требуется их соответственно не менее b_1 , b_2 , b_3 единиц. В магазине продается пять различных продуктов по цене c_1, c_2, \dots, c_5 . Единица продукта i -го вида содержит a_{ij} единиц j -го питательного вещества, т. е., например, a_{23} показывает, что в единице второго продукта третьего питательного вещества будет a_{23} единиц. Какое количество продуктов x_1, x_2, x_3, x_4, x_5 каждого вида следует купить, чтобы стоимость продуктов была минимальна и рацион семьи содержал все необходимые питательные вещества в нужном количестве?

Целевая функция этой задачи — минимизировать по x_1, x_2, x_3, x_4, x_5 стоимость продуктов:

$$c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 + c_5x_5 = \sum_{i=1}^5 c_i x_i \rightarrow \min_{x_1, \dots, x_5}.$$

Условия-ограничения задачи следующие: количество первого питательного вещества должно быть не менее b_1 , т. е.

$$a_{11}x_1 + a_{21}x_2 + a_{31}x_3 + a_{41}x_4 + a_{51}x_5 = \sum_{i=1}^5 a_{i1}x_i \geq b_1.$$

Аналогично для других питательных веществ получим неравенства

$$a_{12}x_1 + a_{22}x_2 + a_{32}x_3 + a_{42}x_4 + a_{52}x_5 = \sum_{i=1}^5 a_{i2}x_i \geq b_2,$$

$$a_{13}x_1 + a_{23}x_2 + a_{33}x_3 + a_{43}x_4 + a_{53}x_5 = \sum_{i=1}^5 a_{i3}x_i \geq b_3.$$

Очевидно, что количество продуктов — величина неотрицательная:

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0.$$

Разработкой методов решения задач, содержащих целевую функцию и условия-ограничения (задач на условный экстремум), занимаются в разделе математики, называемом *математическое программирование*.

Математическое программирование — это математическая дисциплина, в которой изучают теорию и методы решения задач о нахождении экстремумов функций на множествах, определяемых линейными и нелинейными ограничениями в виде равенств и неравенств.

Название «математическое программирование» связано с тем, что целью решения задач является выбор программы действий.

Отдельные задачи математического программирования и методы их решения известны давно. Так, в числе старинных русских задач по математике есть следующая.

Сколько надо взять бабе на базар для продажи живых гусей, уток и кур, чтобы выручить как можно больше денег, если она может нести не более P кг, причем известно, что a_1 — вес одной курицы, a_2 — вес одной утки, a_3 — вес одного гуся, c_1 — стоимость одной курицы, c_2 — стоимость одной утки, c_3 — стоимость одного гуся.

Пусть x_1 , x_2 , x_3 — число соответственно кур, уток и гусей, взятых бабой для продажи. Целевая функция этой задачи — максимальная выручка от продажи птицы:

$$c_1x_1 + c_2x_2 + c_3x_3 = \sum_{i=1}^3 c_i x_i \rightarrow \max_{x_1, x_2, x_3}.$$

Условие-ограничение определяется весом товара, который может нести баба:

$$a_1x_1 + a_2x_2 + a_3x_3 = \sum_{i=1}^3 a_i x_i \leq P.$$

Среди старинных задач по математике встречается *задача о Кёнигсбергских мостах*, сформулированная и решенная в 1736 г. известным математиком Л. Эйлером (1707–1783): можно ли поочередно обойти все семь мостов города Кёнигсберга, соединяющих районы этого города с островом на реке Прегель, пройдя по каждому мосту только один раз. Как мы увидим далее, это тоже задача математического программирования.

Принципиальные результаты теории оптимизации, явившейся основой математического программирования, были получены еще в период становления математического анализа. В связи с этим следует отметить теорему французского математика П. Ферма (1601–1665) о необходимом условии локального экстремума в *безусловной задаче оптимизации* и исследования другого французского математика Ж. Лагранжа (1736–1815) в теории *условных экстремумов*, указывающие необходимые условия экстремума в задаче оптимизации при наличии ограничений в виде равенств. Ж. Лагранж предложил метод решения задач на условный экстремум (1797 г.), который заключается в сведении этих

задач к задачам на безусловный экстремум вспомогательной функции, названной впоследствии *функцией Лагранжа*. Сам метод получил название *метод (неопределенных) множителей Лагранжа*. Функцию Лагранжа применяют как при исследовании задач вариационного исчисления, так и задач математического программирования.

Естественное развитие методов математического анализа, используемых для нахождения точек экстремумов функций, привело к появлению таких математических дисциплин, как вариационное исчисление и математическое программирование. *Вариационное исчисление* — раздел математики, в котором изучают методы отыскания экстремальных (наибольших и наименьших) значений *функционалов* — переменных величин, зависящих от выбора одной или нескольких функций. Одной из первых задач вариационного исчисления была знаменитая *задача о брахистохроне*, сформулированная И. Бернулли (1696 г.): определить форму кривой, лежащей в вертикальной плоскости, по которой тяжелая материальная точка, двигаясь под действием одной только силы тяжести и не имея начальной скорости, перейдет из верхнего положения в нижнее за минимальное время. Эта задача сводится к отысканию функции $y(x)$, доставляющей минимум функционалу

$$T(y) = \int_a^b \frac{\sqrt{1 + (dy/dx)^2}}{\sqrt{20y(x)}} dx,$$

где a и b — абсциссы верхней и нижней точек.

Несмотря на столь ранние истоки математического программирования его развитие относится к концу 30-х годов XX столетия. Математическое программирование развивалось как дисциплина, посвященная теории и методам решения задач управления и планирования, а далее — как раздел возникшей в 50-е годы науки об *исследовании операций* при создании совокупности методологических средств, называемых *системным анализом*.

Наиболее разработанным разделом математического программирования является *линейное программирование*, содержащее теорию и методы решения условных экстремальных задач, в которых критерии оптимальности *линейно* зависят от неизвестных, а ограничения — *линейные равенства и неравенства*. Развитие линейного программирования тесно связано с задачами управления и планирования. Первые публикации по линейному программированию

принадлежат советскому ученому Л.В. Канторовичу*, удостоенному в 1975 г. совместно с американским ученым Т. Купмансом Нобелевской премии за внесенный ими вклад в теорию оптимизации распределения ресурсов.

Математическое программирование стало бурно развиваться наряду с совершенствованием вычислительной техники и применением ЭВМ в научных исследованиях. Появление быстродействующих вычислительных машин создало мощные предпосылки для автоматизации решений многочисленных задач управления и стимулировало разработку специальных новых математических методов, позволяющих сводить решение задач управления и планирования к последовательности автоматически выполняемых операций в соответствии с исходной информацией, при использовании математического, в основном линейного, программирования.

Методы математического программирования применялись и одновременно развивались во время Второй мировой войны для планирования военных операций. Еще до начала Второй мировой войны методы анализа военных систем с использованием математического программирования стали применяться военными специалистами в Великобритании, а затем и в других странах. В США и Канаде были созданы специальные подразделения, занимавшиеся анализом военных операций. В 1938 г. в США был введен термин «исследование операций» для характеристики рода деятельности необычной исследовательской группы, созданной по инициативе организации Air Ministry Research Station и выполнявшей работы по анализу военных систем, в частности решавшей задачи оптимального использования радиолокационных установок в общей системе обороны страны. Этот анализ являлся основой для принятия командованием соответствующих решений. Впоследствии исследование операций сформировалось в научное направление.

Исследование операций — построение, разработка и использование математических моделей для принятия оптимальных решений. Описание всякой задачи исследования операций включает задание компонентов (факторов) решения, налагаемые на них ограничения и систему целей. Каждой из целей соответствует целевая функция, заданная на множестве допустимых решений, значения которой выражают меру осуществления цели.

* Канторович Л.В. Математические методы в организации и планировании производства. — Л.: Изд-во ЛГУ, 1939.

Среди задач исследования операций выделяются те, в которых имеется *одна* целевая функция, принимающая численные значения. Это и есть задачи математического оптимального программирования, т. е. математическое программирование — раздел науки об исследовании операций. Задачи с *несколькими* целевыми функциями или с одной целевой функцией, но принимающей векторные значения или значения еще более сложной природы, называют *многокритериальными*. Их решают путем сведения к задачам с единственной целевой функцией либо на основе использования *теории игр*.

Задачи исследования операций классифицируют и по их *теоретико-информационным свойствам*. Если субъект в ходе принятия решения сохраняет свое информационное состояние, т. е. никакой информации не приобретает и не утрачивает, то принятие решения можно рассматривать как мгновенный акт. Соответствующие задачи называются *статическими*. Напротив, если субъект в ходе принятия решения изменяет свое информационное состояние, получая или теряя информацию, то в такой *динамической задаче* обычно целесообразно принимать решение *поэтапно* (многошаговое решение). Значительная часть динамических задач исследования операций входит в раздел математики *динамическое программирование*.

С конца 40-х годов XX в. сфера приложения методов исследования операций стала охватывать разнообразные стороны человеческой деятельности. Исследование операций используют для решения как чисто технических (особенно технологических), так и технико-экономических задач, а также задач управления на различных уровнях.

Лишь отдельные задачи исследования операций поддаются аналитическому решению и сравнительно немногие — численному решению вручную. Поэтому рост возможностей использования методов исследования операций тесно связан с прогрессом вычислительной техники.

В настоящее время издается много научных журналов по исследованию операций, первый из которых вышел в свет в 1950 г. В 1957 г. в Лондоне был созван первый конгресс Международной Федерации обществ исследования операций (International Federation of Operations Research Societies — IFORS); эти конгрессы проводят каждые три года.

С 50-х годов XX в. для обоснования решений сложных проблем в системах политического, социального, военного, экономи-

ческого, научного и технического характера стали применять совокупность методологических средств, получившую название *системный анализ*. Системный анализ используют для решения таких задач, как распределение производств и мощностей для выпуска различных видов изделий, определение будущей потребности в новом оборудовании и в рабочей силе той или иной квалификации, прогнозирование спроса на различные виды продукции, а также при решении проблем, связанных с развитием и техническим оснащением вооруженных сил, освоением космоса и т. д.

Системный анализ опирается на ряд прикладных математических дисциплин и разделов, в частности на исследования операций. Когда в задаче системного анализа имеется *одна* четко выраженная цель, степень достижения которой можно оценить на основе *одного* критерия, используют методы математического программирования. Первое применение системного анализа в военном деле относят к истории древних веков, когда правитель Сиракуз обратился к Архимеду с просьбой помочь осажденному городу прорвать осаду римлян.

Отдельные исследования по системному анализу проводились в конце XIX в. и начале XX в., а также в Первую мировую войну. Так, в 1886 г. военное командование прибегло к использованию системного анализа, чтобы принять решение относительно производства 12-дюймовых орудий (1 дюйм — 2,54 см), заряжающихся с казенной части и предназначенных для использования в береговой артиллерии. Необходимо было сделать выбор между орудием, выпускаемым фирмой Krupp, и орудием нового образца американского производства. Во время Первой мировой войны с помощью системного анализа разрабатывались, например, стратегические планы борьбы с подводными лодками. Однако эти работы не имели практического применения и были неизвестны. Поэтому во время Второй мировой войны работы пришлось начинать заново. Системный анализ (и его часть — исследование операций) во время Второй мировой войны применяли в основном при тактическом планировании операций, например для того, чтобы решить, что в первую очередь использовать в качестве радиолокационных помех — пассивные или активные помехи; как определить наиболее эффективные цели для бомбометания; какие из способов обнаружения подводных лодок являются наилучшими и т. п.

После войны область исследований переместилась от решения тактических задач к решению задач планирования, так как возникла необходимость создания новых видов оружия.

Примерами задач, решаемых с помощью методов исследования операций при использовании математического программирования, могут быть следующие:

- 1) разработка методов управления техникой, обеспечивающих достаточно высокий уровень эффективности деятельности людей;
- 2) разработка методов использования имеющейся в распоряжении людей техники, обеспечивающей выполнение поставленной задачи с минимальными затратами или с максимальным эффектом;
- 3) разработка техники и материалов, которые необходимо создать или приобрести в рамках общей стратегии деятельности людей.

Для решения первых двух задач наиболее полно используют применение формальных методов, которые являются максимально продуктивными. Методы, применяемые для принятия решений и распределения ресурсов в различных областях науки и техники (экономике, торговле и промышленности — управление запасами, назначение персонала, составление маршрутов и т. п.) интенсивно развиваются в последние десятилетия.

Важным классом задач математического программирования являются так называемые *сетевые (поточковые)* задачи, в терминах которых могут быть сформулированы задачи линейного программирования.

Рассмотрим в качестве примера так называемую *транспортную задачу*, являющуюся одной из первых потоковых задач (см. гл. 3), решенную Ф. Л. Хитчкоком в 1941 г.

Пусть имеются два завода и три склада. Заводы производят соответственно s_1 и s_2 единиц продукции, возможности складов — d_1 , d_2 , d_3 единиц, $s_1 + s_2 = d_1 + d_2 + d_3$. Задача состоит в том, чтобы минимизировать затраты на перевозку продукции с заводов на склады.

Пусть x_{ij} — объем продукции, который необходимо перевезти с i -го завода на j -й склад, и c_{ij} — стоимость перевозки единицы продукции с i -го завода на j -й склад. Тогда целевая функция задачи состоит в минимизации стоимости перевозки:

$$c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{23} = \sum_{i=1}^2 \sum_{j=1}^3 c_{ij}x_{ij} \rightarrow \min.$$

Условия того, что вся продукция будет вывезена с каждого завода, запишем в следующем виде:

$$x_{11} + x_{12} + x_{13} = \sum_{j=1}^3 x_{1j} = s_1,$$

$$x_{21} + x_{22} + x_{23} = \sum_{j=1}^3 x_{2j} = s_2.$$

Эти два равенства можно записать кратко:

$$\sum_{j=1}^3 x_{ij} = s_i, \quad i=1,2.$$

Условия заполнения складов имеют вид

$$\sum_{i=1}^2 x_{ij} = d_j, \quad j=1,2,3,$$

причем $x_{ij} \geq 0$, $i=1,2$, $j=1,2,3$.

Эта модель может быть описана с помощью сети (см. далее гл. 3), если предположить, что узлами сети являются заводы и склады, а дугами — имеющиеся для перевозки груза дороги (рис. 1.2). Сформулированная транспортная задача является частным случаем задачи поиска потока минимальной стоимости на сети. Сетевые задачи применяют при проектировании и совершенствовании больших и сложных систем, а также при поиске путей их наиболее рационального использования. В первую очередь, это связано с тем, что с помощью сетей можно довольно просто построить модель системы. Кроме того, расширение области использования сетей связано с тем, что методы сетевого анализа позволяют:

- 1) строить модель сложной системы как совокупность простых систем;
- 2) составлять формальные процедуры для определения качественных характеристик системы;
- 3) указывать механизм взаимодействия компонентов управляющей системы с целью описания последней с помощью ее основных характеристик;
- 4) определять, какие данные необходимы для исследования системы;

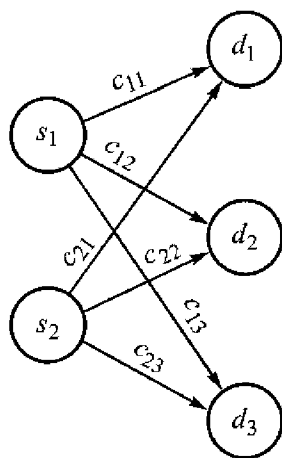


Рис. 1.2. Сеть для транспортной задачи

5) проводить исследования управляющей системы и составлять предварительное расписание работы ее компонентов.

Основное достоинство сетевого подхода заключается в том, что он может быть успешно применен к решению практически любых задач, когда исследователь может точно построить *сетевую модель*. Преимущества использования сетевых моделей можно сформулировать следующим образом:

1) сетевые модели могут точно описывать многие реально существующие системы;

2) для людей, не занимающихся научной работой, сетевые модели являются, вероятно, более понятными, чем любые другие модели, используемые в исследовании операций (пользователю легче понять сетевую диаграмму, чем абстрактные формулы);

3) сетевые алгоритмы позволяют находить наиболее эффективные решения при изучении некоторых больших систем;

4) по сравнению с другими сетевые алгоритмы нередко позволяют решать задачи со значительно бóльшим числом переменных и ограничений; это становится возможным в силу того, что часто удается ограничиться изучением лишь части рассматриваемой системы.

Сетевой анализ берет начало с упоминавшейся задачи Эйлера о мостах Кёнигсберга. Спустя более века Дж. К. Максвелл и Г. Р. Кирхгофф, исследуя электрические цепи, сформулировали некоторые основные принципы сетевого анализа. В начале XX в. европейскими и американскими инженерами были разработаны методы расчета наибольшей пропускной способности телефонных линий и коммутаторов, позволяющие обеспечить гарантированное обслуживание определенного числа абонентов. В связи с развитием вычислительной техники стали проводиться более глубокие исследования построенных моделей и поиск путей реализации алгоритмов на ЭВМ.

В настоящее время трудно назвать область практической и научной деятельности, где бы не применялись методы математического программирования. Области их использования: планирование производства; управление использованием запасов полезных ископаемых и трудовыми ресурсами; планирование и размещение объектов; техническое обслуживание оборудования; планирование работ над проектами и календарное планирование; построение систем: вычислительных, информационных, городской сферы обслуживания, здравоохранения, электроэнергетических, военных, транспортных; организация туризма, спорта и развлечений и т. д.

Поэтому наряду с термином «решение» в математическом программировании используются в этом же смысле термины «план», «стратегия», «управление», «поведение».

§ 1.2. Общая запись задачи математического программирования и ее виды

Из рассмотренных здесь примеров видно, что задача математического программирования должна содержать некую целевую функцию, оптимум которой следует определить, и систему равенств и неравенств, описывающих условия-ограничения задачи. Общая задача математического программирования состоит в определении вектора x^* с координатами $x_1^*, x_2^*, \dots, x_n^*$, который является решением следующей задачи: оптимизировать функцию

$$f(x_1, x_2, \dots, x_n) \quad (1.1)$$

при ограничениях

$$\begin{aligned} g_1(x_1, x_2, \dots, x_n) &\geq 0, \\ g_2(x_1, x_2, \dots, x_n) &\geq 0, \\ &\dots\dots\dots \end{aligned} \quad (1.2)$$

$$\begin{aligned} g_m(x_1, x_2, \dots, x_n) &\geq 0; \\ h_1(x_1, x_2, \dots, x_n) &= 0, \\ h_2(x_1, x_2, \dots, x_n) &= 0, \\ &\dots\dots\dots \end{aligned} \quad (1.3)$$

$$h_p(x_1, x_2, \dots, x_n) = 0.$$

Используем понятие вектора как упорядоченную совокупность действительных чисел $x = (x_1, x_1, \dots, x_n)$ (в отличие от свободного вектора, известного в геометрии, — направленного отрезка, который можно переносить в пространстве параллельно его первоначальному положению). Тогда выражения (1.1)–(1.3) можно записать в более компактной форме: оптимизировать функцию $f(x)$ при ограничениях

$$\begin{aligned} g_i(x) &\geq 0, & i &= 1, 2, \dots, m; \\ h_j(x) &= 0, & j &= 1, 2, \dots, p. \end{aligned}$$

Текущие индексы i и j принимают все целочисленные значения от 1 до соответственно m и p .

Координаты вектора x часто необходимо записывать не в виде строки, а в виде столбца. Для этого используется операция транспонирования — элементы строки становятся соответствующими элементами столбца, и наоборот. Обозначается эта операция верхним индексом «Т»:

$$(x_1, x_2, \dots, x_n)^T \equiv \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}.$$

Общая задача математического программирования разбивается на задачи, названия которых определяются видом функций, которые необходимо оптимизировать и которые входят в условия-ограничения, типом переменных в задаче, алгоритмом решения. Если функции $f(x)$, $g_i(x)$, $h_j(x)$ в выражениях (1.1)–(1.3) линейны, то полученную задачу называют задачей *линейного программирования* (например, рассмотренные ранее задача о питании и транспортная задача).

Если хотя бы одна из функций $f(x)$, $g_i(x)$, $h_j(x)$ нелинейна, то (1.1)–(1.3) называют задачей *нелинейного программирования*. В свою очередь, множество задач нелинейного программирования разбивается на подмножества, имеющие собственные названия. Так, если $f(x)$ является квадратичной функцией, а ограничения линейны, то получаем задачу *квадратичного программирования* (более точно, $f(x)$ должна быть квазиопределенной квадратичной формой*).

В *сепарабельном программировании* целевая функция $f(x)$ представляет собой сумму функций, различных для каждой переменной. Условия-ограничения здесь могут быть как линейными, так и нелинейными, но все недиагональные элементы матрицы, состоящей из вторых частных производных любой функции задачи, равны нулю.

Если координаты искомого вектора являются только целыми числами, то получаем задачу *целочисленного программирования* (линейного или нелинейного).

* Некоторые определения приведены в § 1.3.

§ 1.3. Некоторые сведения об экстремуме функции, частных производных, градиенте и производной по направлению

Из курса математического анализа читателю известны простейшие задачи нахождения точек максимума или минимума функции одной переменной. Функция $y = f(x)$, определенная в точке x_0 , достигает *максимума (минимума)* в окрестности точки x_0 , если для всех точек этой окрестности выполнено неравенство $f(x) \leq f(x_0)$ ($f(x) \geq f(x_0)$).

Максимум и минимум функции объединяют одним названием *экстремум*. Как правило, точка x_0 — внутренняя точка естественной области определения функции $f(x)$, и экстремум называют *внутренним*. Если существует производная $f'(x)$ в точке x_0 , то функция $f(x)$ может иметь в точке x_0 внутренний экстремум лишь в том случае, когда при $x = x_0$ производная $f'(x_0)$ равна нулю (*необходимое условие экстремума*). Экстремум может быть и в тех точках x_0 , где производная $f'(x_0)$ не существует. Однако выполнение необходимого условия еще не означает, что в точке x_0 будет экстремум. Если производная $f'(x)$ в окрестности точки x_0 при переходе через эту точку меняет свой знак с плюса на минус, то x_0 — точка максимума, а если с минуса на плюс, то x_0 — точка минимума. Известен и другой признак: если в точке x_0 первая производная $f'(x_0)$ равна нулю и существует вторая производная $f''(x_0) \neq 0$, то в точке x_0 будет максимум при $f''(x_0) < 0$ и минимум при $f''(x_0) > 0$.

В общем случае, если существуют производные от $f(x)$ до n -го порядка включительно и если $f'(x_0) = \dots = f^{(n-1)}(x_0) = 0$ и $f^{(n)}(x_0) \neq 0$, то функция $f(x)$ имеет в точке x_0 максимум при четном n и $f^{(n)}(x_0) < 0$ и минимум при четном n и $f^{(n)}(x_0) > 0$. Если n нечетно, то функция $f(x)$ в точке x_0 не имеет ни минимума, ни максимума, а имеет точку *перегиба*.

Сформулируем несколько определений, которые потребуются далее.

Действительная функция $f(x)$, определенная в точке x_0 , имеет в этой точке (*локальный*) *максимум* или (*локальный*) *минимум* $f(x_0)$, если существует такое положительное число δ , что при всех приращениях Δx независимого переменного x , равных $x - x_0$, для которых выполнено неравенство $0 < |\Delta x| < \delta$ и существует значение $f(x_0 + \Delta x)$, приращение данной функции удовлетворяет соответственно неравенству

$$\Delta f \equiv f(x_0 + \Delta x) - f(x_0) < 0$$

или

$$\Delta f \equiv f(x_0 + \Delta x) - f(x_0) > 0.$$

Если в каждом из этих случаев выполнены нестрогие неравенства, то говорят, что функция $f(x_0)$ имеет в точке x_0 *нестрогий максимум (минимум)*.

Локальный максимум (минимум) называют *внутренним максимумом (минимумом)* или *граничным максимумом (минимумом)*, если точка x_0 является соответственно внутренней или граничной точкой области определения функции $f(x)$.

В формулировке задачи должна быть точно указана область определения функции $f(x)$. Например, функция $f_1(x) = x$ при $-\infty < x < +\infty$ не имеет максимума, а функция $f_2(x) = x$ при $x \leq 1$ имеет в точке $x = 1$ граничный максимум.

Если неравенство $f(x) < f(x_0)$ ($f(x) > f(x_0)$) выполнено для любой точки x , принадлежащей области определения функции $f(x)$, то говорят о *глобальном максимуме (минимуме)* функции $f(x)$ в точке x_0 . Аналогичные определения справедливы для функции многих переменных.

Функцию $f(x)$, имеющую в данной точке x_0 производную, называют *дифференцируемой в точке x_0* ; функцию, имеющую производную во всех точках некоторого промежутка (a, b) , называют *дифференцируемой в промежутке (a, b)* .

Функцию многих переменных, имеющую полный дифференциал (в данной точке, области), называют *дифференцируемой (в этой точке, области)*. *Необходимое условие* дифференцируемости функции многих переменных — наличие частных производных первого порядка (в точке, области). *Достаточные условия* дифференцируемости функции многих переменных — существование и непрерывность всех частных производных первого порядка (в точке, области).

Числовую функцию $f(x)$ одного векторного аргумента $x = (x_1, x_2, \dots, x_n)$ вида

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j \equiv \sum_{i=1}^n a_{ii} x_i^2 + 2 \sum_{i=1}^n \sum_{\substack{j=2 \\ j>i}}^n a_{ij} x_i x_j,$$

где a_{ij} — элементы симметрической матрицы $A = \|a_{ij}\|$ порядка $n \times n$, $a_{ij} = a_{ji}$ (квадратной таблицы чисел), называют *квадратичной формой* переменных.

Квадратичную форму $f(x)$ называют *положительно (отрицательно) определенной*, если для любого ненулевого вектора x выполняется неравенство

$$f(x) > 0 \quad (f(x) < 0).$$

Такие формы объединяют общим названием — *знакоопределенные*. Если существует ненулевой вектор x , для которого $f(x) = 0$, форму называют *квазизнакоопределенной*.

Квадратичную форму называют *знакопеременной*, если существуют такие векторы x^1 и x^2 , что $f(x^1) > 0$, $f(x^2) < 0$.

Для наглядного представления поведения функции $y = f(x)$ строят *график функции*. Если независимую переменную x (аргумент) и зависимую переменную y рассматривать как декартовы координаты на плоскости, то действительная функция $y = f(x)$ действительной переменной изобразится кривой — графиком функции y от x .

Для функции многих переменных $y = f(x_1, x_2, \dots, x_n)$ упорядоченному множеству значений независимых переменных x_1, x_2, \dots, x_n ставят в соответствие значения переменной y . Множество значений x_1, x_2, \dots, x_n , для которых определено соотношение $y = f(x_1, x_2, \dots, x_n)$, называется *областью определения функции* $f(x_1, x_2, \dots, x_n)$.

Графиком функции многих переменных является поверхность для функций двух переменных и *гиперповерхность* для функций большего числа переменных. Чтобы представить функцию n переменных, вводятся понятия *линий* и *поверхностей уровня*. Это геометрическое место точек, в которых функция принимает одно и то же значение. Уравнение поверхности уровня имеет вид $f(x_1, x_2, \dots, x_n) = C$. Для различных значений константы C получаем семейство поверхностей уровня, определяющих поведение функции. Линии уровня вводятся для функции двух переменных: $f(x_1, x_2) = C$. Семейство линий уровня дает возможность представить функцию двух переменных $y = f(x_1, x_2) = C$ на плоскости. Например, семейство линий уровня на географических картах дает представление и о морских глубинах, и о высоте горных хребтов.

Для характеристики скорости изменения функции многих переменных относительно одной из переменных, например x_i , при фиксированных значениях остальных независимых переменных

вводится понятие *частных производных* $\frac{\partial f}{\partial x_i}$. Частные производ-

ные $\frac{\partial f}{\partial x_i}$, $i = 1, 2, \dots, n$, определяются дифференцированием функции $f(x_1, x_2, \dots, x_n)$ по x_i , если остальные $n - 1$ независимых переменных рассматривать как постоянные параметры.

Направление, в котором скорость возрастания функции многих переменных наибольшая, определяется вектором, называемым *градиентом*. Противоположное направление называют *антиградиентом*. Градиент скалярной функции $f(x_1, x_2, \dots, x_n)$ есть *векторная функция* точки и определяется следующим образом:

$$\text{grad } f \equiv \nabla f(x_1, x_2, \dots, x_n) = \frac{\partial f}{\partial x_1} \mathbf{i} + \frac{\partial f}{\partial x_2} \mathbf{j} + \dots + \frac{\partial f}{\partial x_n} \mathbf{k},$$

где ∇ — знак градиента, $\mathbf{i}, \mathbf{j}, \dots, \mathbf{k}$ — единичные векторы (орты), направленные по координатным осям:

$$\mathbf{i} = (1, 0, \dots, 0), \quad \mathbf{j} = (0, 1, \dots, 0), \quad \dots, \quad \mathbf{k} = (0, 0, \dots, 1).$$

Иногда применяется обозначение градиента в виде $\nabla_x f$; индекс x показывает переменные, по которым определяется градиент. Другими словами, градиент скалярной функции — это вектор, координатами которого являются частные производные заданной функции.

Скорость изменения скалярной функции $f(x_1, x_2, \dots, x_n)$ в произвольном направлении, задаваемом единичным вектором $\mathbf{u} = \cos \alpha \mathbf{i} + \cos \beta \mathbf{j} + \dots + \cos \gamma \mathbf{k}$ с направляющими косинусами $\cos \alpha, \cos \beta, \dots, \cos \gamma$, определяется *производной по направлению* (действительное число)

$$\frac{\partial f}{\partial \mathbf{u}} = \frac{\partial f}{\partial x_1} \cos \alpha + \frac{\partial f}{\partial x_2} \cos \beta + \dots + \frac{\partial f}{\partial x_n} \cos \gamma.$$

Производная по направлению связана с градиентом скалярной функции ∇f скалярным произведением $df/du = (\nabla f, \mathbf{u})$.

Скалярным произведением двух векторов $\mathbf{a} = (a_1, a_2, \dots, a_n)$ и $\mathbf{b} = (b_1, b_2, \dots, b_n)$ называют действительное число, равное сумме произведений соответствующих координат векторов:

$$(\mathbf{a}, \mathbf{b}) = a_1 b_1 + a_2 b_2 + \dots + a_n b_n,$$

или произведению длин этих векторов на косинус угла между ними:

$$(a, b) = |a| \cdot |b| \cos(\widehat{a, b}).$$

Градиент ∇f всегда ортогонален поверхности (линии) уровня функции $f(x_1, x_2, \dots, x_n)$. Действительно,

$$\frac{\partial f}{\partial u} = |\nabla f| \cdot |u| \cos(\widehat{\nabla f, u}).$$

Производная по направлению касательной к поверхности (линии) уровня df/du равна нулю, $|\nabla f| \neq 0$, $|u| \neq 0$. Поэтому

$$\cos(\widehat{\nabla f, u}) = 0, \quad (\widehat{\nabla f, u}) = \frac{\pi}{2}, \quad \nabla f \perp u.$$

Нам понадобятся понятия *линейной зависимости* и *независимости векторов*. Векторы a_1, a_2, \dots, a_m называют линейно зависимыми, если найдутся такие действительные числа $\alpha_1, \alpha_2, \dots, \alpha_m$, не все равные нулю, что линейная комбинация векторов a_1, a_2, \dots, a_m равна нулю: $\alpha_1 a_1 + \alpha_2 a_2 + \dots + \alpha_m a_m = 0$. Если это равенство выполнено только тогда, когда все числа $\alpha_1, \alpha_2, \dots, \alpha_m$ равны нулю, то векторы a_1, a_2, \dots, a_m называют линейно независимыми.

Из определения линейной зависимости векторов следует, что если векторы линейно зависимы, то один из них может быть представлен в виде линейной комбинации остальных, и, наоборот, если один из векторов есть линейная комбинация остальных, то векторы линейно зависимы.

§ 1.4. Особенности нахождения оптимальных решений в задачах математического программирования

В задачах математического программирования требуется найти условный экстремум (максимум или минимум) функции при наличии ограничений. Рассмотрим задачу математического программирования, в которой есть ограничения только в виде равенств. Пусть целевая функция задачи является функцией двух переменных: $z = f(x) = f(x_1, x_2)$. Ее аргументы связаны уравнением $\varphi(x_1, x_2) = 0$ (ограничения в виде неравенств отсутствуют). Если функции $z = f(x_1, x_2)$ поставить в соответствие некоторую поверхность, то в данной задаче необходимо найти следующие точки:

1) принадлежащие линии пересечения поверхности $z = f(x_1, x_2)$ и цилиндра с образующей, параллельной оси z , и с направляющей $\varphi(x_1, x_2) = 0$;

2) являющиеся экстремальными для функции $z = f(x_1, x_2)$ (рис. 1.3).

Как видно из рис. 1.3, точки условного экстремума A и B не совпадают с наибольшим или наименьшим значением функции $z = f(x_1, x_2)$ — с безусловным экстремумом функции $f(x_1, x_2)$.

Если из уравнения связи $\varphi(x_1, x_2) = 0$ можно выразить в явном виде одну переменную через другую, например $x_2 = \psi(x_1)$, то $z = f(x_1, x_2) = f(x_1, \psi(x_1))$ становится функцией одной переменной x_1 и ее безусловный экстремум определяется традиционными методами (приравниваем первой производной функции $f(x_1, \psi(x_1))$ по x_1 нулю). Безусловный экстремум функции $f(x_1, \psi(x_1))$ является условным экстремумом для функции $f(x_1, x_2)$ при ограничении $\varphi(x_1, x_2) = 0$.

Однако выразить в явном виде из условий-ограничений необходимую часть переменных, как правило, не удастся.

Лагранж предложил оригинальный метод нахождения условного экстремума функции. Метод носит его имя. Пусть требуется решить следующую задачу: минимизировать функцию $f(x_1, x_2, \dots, x_n)$ при ограничениях $h_j(x_1, x_2, \dots, x_n) = 0$, $j = 1, 2, \dots, p$. По условию задачи составляется *функция Лагранжа*

$$F(x_1, x_2, \dots, x_n) = f(x_1, x_2, \dots, x_n) + \sum_{j=1}^p \lambda_j h_j(x_1, x_2, \dots, x_n).$$

Здесь λ_j — неизвестные постоянные множители, подлежащие определению (*множители Лагранжа*), т. е. требуется найти n неизвестных x_1, x_2, \dots, x_n и p множителей Лагранжа $\lambda_1, \lambda_2, \dots, \lambda_p$.

Для рассматриваемого в начале параграфа примера, когда минимизируется функция $f(x_1, x_2)$ при условии $\varphi(x_1, x_2) = 0$, функция Лагранжа имеет вид

$$F(x_1, x_2) = f(x_1, x_2) + \lambda \varphi(x_1, x_2).$$

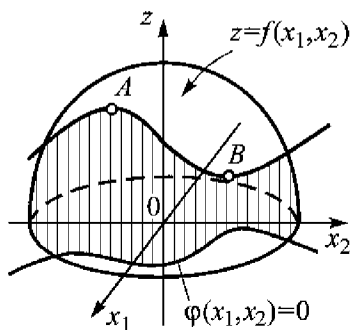


Рис. 1.3. Геометрическая интерпретация метода Лагранжа

Точки, в которых возможен экстремум, находятся как решение системы алгебраических уравнений, полученной приравняв нулю частных производных функции Лагранжа по искомым переменным (n уравнений) и включением в эту систему p ограничений-равенств. Метод Лагранжа сводит задачу нахождения *условного экстремума* функции $f(x)$ к задаче нахождения *безусловного экстремума* функции $F(x, \lambda)$.

Ограничения-неравенства еще более усложняют задачу. Они задают область допустимых значений переменных. Например, пусть требуется оптимизировать некоторую функцию $f(x)$ при ограничениях

$$g_1(x) = x_1 - x_2^2 \geq 0,$$

$$g_2(x) = 1 - x_1^2 - x_2^2 \geq 0.$$

Область допустимых значений переменных x_1 и x_2 в этой задаче есть пересечение области, лежащей «внутри» параболы $x_1 = x_2^2$, с кругом единичного радиуса, уравнение окружности которого имеет вид $x_1^2 + x_2^2 = 1$ (рис. 1.4).

Пересечение цилиндра, направляющей которого является граница полученной области D , с поверхностью $z = f(x)$ может привести к самым разнообразным вариантам положения точки экстремума. На рис. 1.5, *а–в* показаны поверхности, полученные в результате пересечения цилиндра, направляющей которого служит граница области допустимых значений переменных x_1 и x_2 , и поверхности, соответствующей целевой функции $z = f(x_1, x_2)$.

На рис. 1.5, *а* точка M безусловного экстремума функции $z = f(x_1, x_2)$ является и точкой условного экстремума задачи. На рис. 1.5, *б* точка M является уже граничной и в ней целевая функция достигает своего наибольшего значения. На рис. 1.5, *в* точка M не принадлежит области допустимых значений переменных, а целевая функция имеет равные наибольшие значения по линиям ALB и AKB (т. е. не ясно, что же брать за решение задачи). Эти неоднозначные результаты получены даже в случае, когда поверхность целевой функции $z = f(x)$ достаточно проста и обладает единственным (глобальным) максимумом.

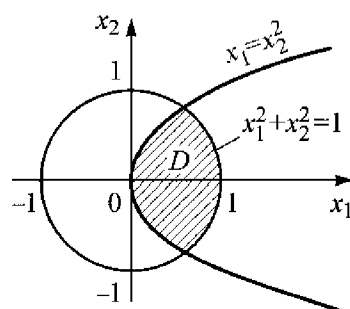
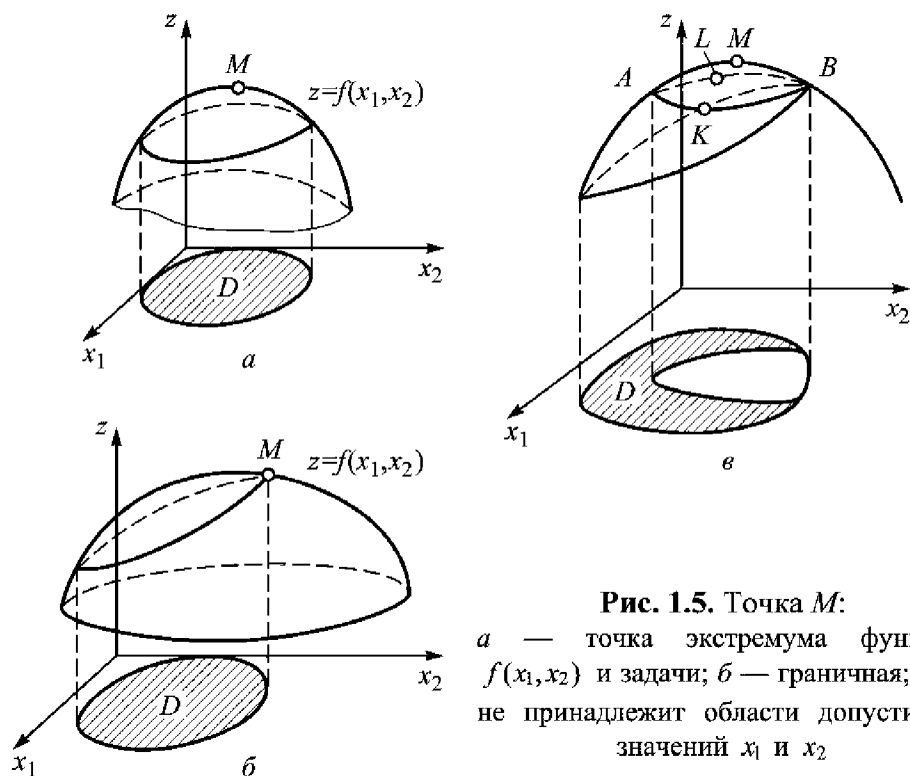
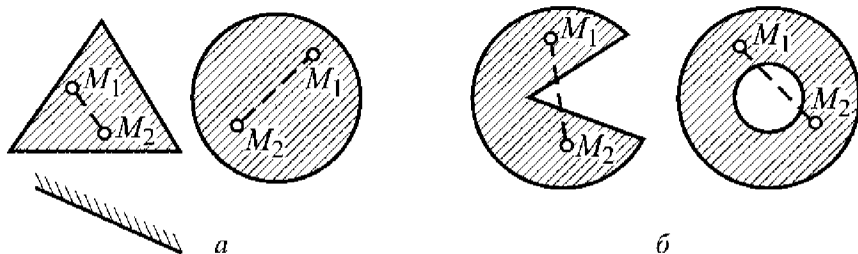


Рис. 1.4. Область допустимых значений x_1 и x_2

Рис. 1.5. Точка M :

a — точка экстремума функции $f(x_1, x_2)$ и задачи; *б* — граничная; *в* — не принадлежит области допустимых значений x_1 и x_2

Наиболее полные результаты в задачах математического программирования получены для *выпуклых* целевых функций, когда область допустимых значений является *выпуклым множеством*. Множество точек D называют *выпуклым*, если для любых точек M_1 и M_2 , принадлежащих области D , отрезок M_1M_2 принадлежит множеству (области) D (рис. 1.6, *a*). Другими словами, любая точка $\lambda M_1 + (1-\lambda)M_2$ принадлежит области D для любого λ , $0 \leq \lambda \leq 1$, и для любых точек M_1 и M_2 , принадлежащих области D . Пересечение конечного числа выпуклых множеств выпукло. На рис. 1.6, *б* показаны *невыпуклые множества*.

Рис. 1.6. Выпуклые (*a*) и невыпуклые (*б*) области (множества)

Функцию $f(M)$ называют *выпуклой* на непустом выпуклом множестве D , если для любых двух точек M_1 и M_2 , принадлежащих области D , и любого числа λ , $0 \leq \lambda \leq 1$, справедливо неравенство

$$f(\lambda M_1 + (1-\lambda)M_2) \leq \lambda f(M_1) + (1-\lambda)f(M_2).$$

Функцию $f(M)$ называют *строго выпуклой*, если для $0 < \lambda < 1$ и $M_1 \neq M_2$ выполнено строгое неравенство

$$f(\lambda M_1 + (1-\lambda)M_2) < \lambda f(M_1) + (1-\lambda)f(M_2).$$

Геометрически выпуклая функция лежит над своими касательными. Примером выпуклой функции является парабола.

Сумма выпуклых на множестве D функций есть также выпуклая на D функция.

Функцию $f(x)$ называют *вогнутой* на выпуклом множестве D , если функция $-f(x)$ выпукла на D .

Ограничения $g_i(x) \geq 0$, $i=1,2,\dots,t$, образуют *выпуклое множество D* (выпуклую область D), если все функции $g_i(x)$ *вогнуты*.

В математическом программировании выделяют важный класс задач — *задачи выпуклого программирования*: минимизировать функцию $f(x)$ при ограничениях $g_i(x) \geq 0$, $i=1,2,\dots,t$, где $f(x)$ — выпуклая функция, а все функции $g_i(x)$ вогнуты, т. е. рассматривают выпуклые функции на выпуклых множествах.

Задачи выпуклого программирования обладают важным положительным свойством: *локальные минимумы* целевых функций являются одновременно *глобальными* (единственными).

Очевидно, что решить подобную задачу проще (но не просто), чем в случае, когда целевая функция $f(x)$ и область D будут общего вида.

§ 1.5. Необходимые и достаточные условия экстремума в задачах математического программирования

В общем случае в задачах математического программирования необходимо найти локальный минимум (максимум) целевой функции, т. е. такое значение x^* , что для x , принадлежащих некоторой окрестности точки x^* , выполнены неравенства $f(x^*) \geq f(x)$ для строгого минимума/максимума и $f(x^*) \leq f(x)$

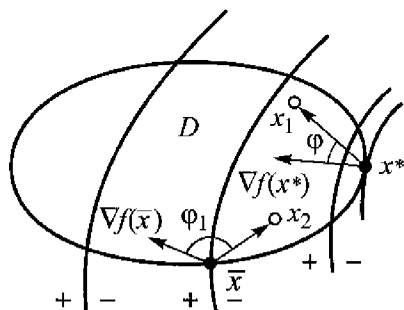


Рис. 1.7. Положение градиента $\nabla f(x^*)$ в точке решения x^* и в точке \bar{x} , не являющейся решением задачи математического программирования (знаками «+» и «-» указано направление возрастания значений линий уровня)

найдется такая точка x_2 , что угол ϕ_1 будет больше $\pi/2$.

В тех случаях, когда решение x^* находится внутри области D , градиент $\nabla f(x^*)$ равен нулю.

Сформулированное условие является *необходимым условием* в задаче минимизации дифференцируемой функции на выпуклом множестве (для выпуклой задачи оно является и *достаточным условием глобальной оптимальности*).

Для области D , имеющей вид параллелепипеда, когда $a_i \leq x_i \leq b_i$, $-\infty < a_i < b_i < +\infty$, $i=1, 2, \dots, m$, данное (необходимое) условие записывается следующим образом:

$$\frac{\partial f(x^*)}{\partial x_i} = 0, \quad \text{если } a_i < x_i^* < b_i,$$

$$\frac{\partial f(x^*)}{\partial x_i} > 0, \quad \text{если } x_i^* = a_i \neq -\infty,$$

$$\frac{\partial f(x^*)}{\partial x_i} < 0, \quad \text{если } x_i^* = b_i \neq +\infty.$$

Здесь градиент направлен внутрь области D .

Чтобы определить координаты возможной оптимальной точки, составим из необходимых условий соответствующие системы уравнений и решим их.

для нестрогого минимума/максимума. Как и для функций одной переменной, в задачах математического программирования требуется сформулировать необходимые и достаточные условия существования оптимума. Если в задаче математического программирования множество D выпукло, а функция $f(x)$ дифференцируема в точке $x^* \in D$, то градиент $\nabla f(x^*)$, если он отличен от нуля, составляет нетупой угол ϕ с вектором, направленным из x^* в любую точку $x_1 \in D$. Другими словами, скалярное произведение $(\nabla f(x^*), x_1 - x^*) \geq 0$ (рис. 1.7). Для точки \bar{x} , не являющейся решением задачи, всегда

В общем случае для задачи вида

$$f(x) \rightarrow \min, \quad (1.4)$$

$$g_i(x) \geq 0, \quad i = 1, 2, \dots, m, \quad (1.5)$$

$$h_j(x) = 0, \quad j = 1, 2, \dots, p, \quad (1.6)$$

вводится функция Лагранжа

$$L(x, u, \lambda) = f(x) - \sum_{i=1}^m u_i g_i(x) + \sum_{j=1}^p \lambda_j h_j(x), \quad (1.7)$$

где u_i , $i = 1, 2, \dots, m$, λ_j , $j = 1, 2, \dots, p$, — множители Лагранжа, подлежащие определению наряду с координатами вектора x . Множители Лагранжа λ_j для ограничений-равенств могут иметь любой знак, множители Лагранжа u_i для ограничений-неравенств — неотрицательны. Если в задаче математического программирования (1.4)–(1.6) множество P , $x \in P \subset \mathbb{R}^n$, выпукло, функции $f(x)$, $g_i(x)$, $i = 1, 2, \dots, m$, выпуклы на P и дифференцируемы в точке $x^* \in D$,

$$D = \{x \in P \mid g_i(x) \geq 0, \quad i = 1, 2, \dots, m, \quad h_j = 0, \quad j = 1, 2, \dots, p\},$$

функции $h_j(x)$, $j = 1, 2, \dots, p$, линейны и при некоторых u^* , λ^* выполнены условия $(\nabla_x L(x^*, u^*, \lambda^*), x - x^*) \geq 0$ для всех $x \in P$ и $u_i^* g_i(x^*) = 0$, $i = 1, 2, \dots, m$, то x^* — (глобальное) решение этой задачи.

Соотношения

$$(\nabla_x L(x^*, u^*, \lambda^*), x - x^*) \geq 0 \quad (1.8)$$

при всех $x \in P$,

$$u_i^* g_i(x^*) = 0, \quad i = 1, 2, \dots, m, \quad (1.9)$$

для задачи выпуклого программирования являются не только необходимыми, но и достаточными условиями существования решения (условия Куна — Таккера):

а) если x^* является внутренней точкой области P , $x^* \in \text{int } P$, то условие (1.8) эквивалентно равенству $\nabla_x L(x^*, u^*, \lambda^*) = 0$;

б) если область P имеет вид параллелепипеда, т. е.

$$P = \{x \in \mathbb{R}^n \mid a_k \leq x_k \leq b_k, \quad k = 1, 2, \dots, n\},$$

где $-\infty < a_k < b_k < +\infty$, то соотношение (1.8) эквивалентно следующему условию: для любого $k = 1, 2, \dots, n$

$$\frac{\partial L}{\partial x_k}(x^*, u^*, \lambda^*) = 0, \quad \text{если } a_k < x_k^* < b_k,$$

$$\frac{\partial L}{\partial x_k}(x^*, u^*, \lambda^*) \geq 0, \quad \text{если } x_k^* = a_k \neq -\infty,$$

$$\frac{\partial L}{\partial x_k}(x^*, u^*, \lambda^*) \leq 0, \quad \text{если } x_k^* = b_k \neq +\infty;$$

в) если в области P выполнено условие неотрицательности только s переменных и P имеет вид

$$P = \{x \in \mathbb{R}^n \mid x_k \geq 0, k = 1, 2, \dots, s\}, \quad 0 \leq s \leq n,$$

то условие (1.8) эквивалентно совокупности условий

$$\begin{aligned} \frac{\partial L}{\partial x_k}(x^*, u^*, \lambda^*) &\geq 0, \\ x_k^* \frac{\partial L}{\partial x_k}(x^*, u^*, \lambda^*) &= 0, \quad k = 1, 2, \dots, s, \\ \frac{\partial L}{\partial x_k}(x^*, u^*, \lambda^*) &= 0, \quad k = s + 1, \dots, n. \end{aligned} \quad (1.10)$$

В задачах, имеющих ограничения-неравенства, появляется дополнительное условие (1.9), которое называют *условием дополняющей нежесткости*. Это условие разделяет ограничения-неравенства на активные, которые в точке оптимума обращаются в нуль ($g_i(x^*) = 0, i = 1, 2, \dots, l_1; l_1 \leq m$), и пассивные ($g_i(x^*) \neq 0, i = l_1 + 1, \dots, m$). Для пассивных ограничений коэффициенты Лагранжа u_i^* должны быть равны нулю, при этом пассивные ограничения не оказывают влияния на выбор решения x^* .

Рассмотрим случай, когда условиями задачи являются только ограничения-неравенства:

$$f(x) \rightarrow \min, \quad g_i(x) \leq 0, \quad i = 1, 2, \dots, m.$$

Тогда в точке минимума имеем

$$\nabla_x f(x^*) + \sum_{i=1}^m u_i^* \nabla_x g_i(x^*) = 0,$$

т. е. *антиградиент* целевой функции является неотрицательной линейной комбинацией градиентов функций, образующих активные ограничения в точке x^* (рис. 1.8). На рис. 1.8 показано множество, образованное неравенствами $g_1(x) \leq 0, g_2(x) \leq 0$,

$g_3(x) \leq 0$. Здесь же в точках x^* и \bar{x} указаны направления градиентов активных ограничений и антиградиента целевой функции. Отсюда следует, что точкой экстремума не может быть точка \bar{x} , так как в ней не выполнено условие того, что антиградиент функции $f(x)$ есть положительная линейная комбинация градиентов активных ограничений. Решением является точка x^* , где данное условие выполнено.

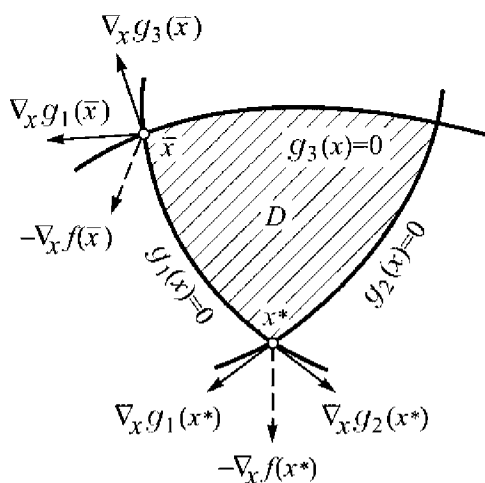


Рис. 1.8. Связь направлений градиентов активных ограничений и антиградиента целевой функции в точке решения x^* и в точке \bar{x} , не являющейся точкой решения

Мы рассмотрели некоторые условия существования решения, используя только производные первого порядка. Однако сформулированы и доказаны условия оптимальности второго порядка, в которых применяют вторые частные производные функции Лагранжа. Здесь мы их рассматривать не будем.

В общем случае задачу математического программирования можно было бы решать по следующей схеме.

1. Записать задачу в каноническом виде (1.4)–(1.6) и составить функцию Лагранжа (1.7).

2. Выписать систему условий, которые характеризуют решение (определяют точки, где возможно существование оптимального решения, — стационарные точки): в развернутой форме записать условия (1.8), (1.9), а также условия, налагаемые задачей на допустимые значения и на множители Лагранжа. Например, для условия (1.10) полная система для определения стационарных точек имеет вид

$$x_k \geq 0, \quad \frac{\partial L}{\partial x_k}(x, u, \lambda) \geq 0, \quad x_k \frac{\partial L}{\partial x_k}(x, u, \lambda) = 0, \quad k = 1, 2, \dots, s,$$

$$\frac{\partial L}{\partial x_k}(x, u, \lambda) = 0, \quad k = s + 1, \dots, n,$$

$$u_i \geq 0, \quad g_i(x) \geq 0, \quad u_i g_i(x) = 0, \quad i = 1, 2, \dots, l,$$

$$g_i(x) = 0, \quad i = l + 1, \dots, m.$$

3. Решить полученную систему необходимых условий. Сделать это в аналитическом виде возможно лишь в редких случаях.

4. Если удалось найти решение системы необходимых условий — получить стационарные точки, необходимо провести исследование этих точек для отбора среди них решений. Это тоже сделать непросто. Иногда проще провести непосредственное исследование поведения целевой функции в стационарной точке.

На последних двух этапах полезно привлечение физических и геометрических соображений о возможном решении задачи математического программирования.

§ 1.6. Теория двойственности и недифференциальные условия оптимальности в задаче выпуклого программирования

В задачах математического программирования (1.4)–(1.6) можно указать условия оптимальности, не прибегая к понятиям производных и градиентов, с помощью так называемой теории *двойственности*. Особенно эффективен этот подход к решению задач выпуклого программирования.

Рассмотрим функцию Лагранжа (1.7). Обозначим через f^* точную нижнюю грань целевой функции задачи (1.4)–(1.6), определенной на допустимом множестве D :

$$f^* = \inf_{x \in D} f(x).$$

Точка $x^* \in D$ является решением задачи (1.4)–(1.6) в том и только в том случае, если $f^* = f(x^*)$.

Введем вектор y с координатами u_i , $i=1,2,\dots,t$, и λ_j , $j=1,2,\dots,p$. Вектор y^* называется *вектором Куна — Таккера* задачи (1.4)–(1.6), если при всех $x \in D$

$$f^* \leq f(x) - \sum_{i=1}^m u_i^* g_i(x) + \sum_{j=1}^p \lambda_j^* h_j(x) = L(x, y^*).$$

Любой задаче математического программирования можно поставить в соответствие так называемую *двойственную задачу* оптимизации. Между прямой и двойственной задачами существуют связи, позволяющие в некоторых случаях упростить решение.

Двойственной к задаче (1.4)–(1.6) называют задачу

$$\varphi(y) \rightarrow \max, \quad y \in Y,$$

где

$$\varphi(y) = \inf_{x \in D} L(x, u, \lambda) = \inf_{x \in D} \left(f(x) - \sum_{i=1}^m u_i g_i(x) + \sum_{j=1}^p \lambda_j h_j(x) \right),$$

$$Y = \{y \in Q \mid \varphi(y) > -\infty\}, \quad Q = \{y \in \mathbb{R}^n \mid y_i \geq 0, i = 1, 2, \dots, k\}.$$

Рассмотрим подробнее задачу линейного программирования

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

при

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

$$x_j \geq 0, \quad j = 1, 2, \dots, s.$$

Переменные $x_j = s+1, \dots, n$ могут иметь любые значения. Функция Лагранжа этой задачи имеет вид

$$L(x, y) = \sum_{j=1}^n c_j x_j + \sum_{i=1}^m y_i \left(b_i - \sum_{j=1}^n a_{ij} x_j \right) = \sum_{j=1}^n \left(c_j - \sum_{i=1}^m a_{ij} y_i \right) x_j + \sum_{i=1}^m b_i y_i,$$

где

$$x \in P, \quad P = \left\{ x \in \mathbb{R}^n \mid \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, 2, \dots, k, \right. \\ \left. \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = k+1, \dots, m \right\},$$

$$y \in Q, \quad Q = \{y \in \mathbb{R}^n \mid y_i \geq 0, i = 1, 2, \dots, k\}.$$

Тогда

$$\inf_{x \in P} L(x, y) = \begin{cases} \sum_{i=1}^m b_i y_i, & \text{если } y \in Y, \\ -\infty, & \text{если } y \in Q \setminus Y, \end{cases}$$

где

$$Y = \left\{ y \in Q \mid \sum_{i=1}^m a_{ij} y_i < c_j, \quad j = 1, 2, \dots, s, \quad \sum_{i=1}^m a_{ij} y_i = c_j, \quad j = s+1, \dots, n \right\}.$$

Согласно определению двойственности, двойственной задачей к исходной задаче является следующая:

$$\sum_{i=1}^m b_i y_i \rightarrow \max$$

при

$$\sum_{i=1}^m a_{ij} y_i \leq c_j, \quad j = 1, 2, \dots, s,$$

$$\sum_{i=1}^m a_{ij} y_i = c_j, \quad j = s+1, \dots, n,$$

$$y_i \geq 0, \quad i = 1, 2, \dots, k.$$

При этом в точке оптимума (x^*, y^*) выполняются соотношения

$$x_j^* \left(\sum_{i=1}^m a_{ij} y_i^* - c_j \right) = 0, \quad j = 1, 2, \dots, s,$$

$$y_i^* \left(\sum_{j=1}^n a_{ij} x_j^* - b_i \right) = 0, \quad i = 1, 2, \dots, k.$$

Исходную задачу называют *прямой*. Если целевую функцию в двойственной задаче $\varphi(y) \rightarrow \max$ заменить на $-\varphi(y) \rightarrow \min$, то можно утверждать, что задача, двойственная к произвольной задаче математического программирования, всегда выпукла. Если в задаче математического программирования допустимое множество значений x замкнуто и выпукло, функции $f(x)$, $g_i(x)$, $i = 1, 2, \dots, m$, непрерывны и выпуклы на D , функции $h_j(x)$, $j = 1, 2, \dots, p$, линейны или отсутствуют и решение прямой задачи конечно ($f^* > -\infty$), в частности она имеет решение, то множество решений двойственной задачи непусто и совпадает с множеством векторов Куна — Таккера прямой задачи. При этом справедливо соотношение двойственности $f^* = \varphi^*$, т. е. минимум целевой функции прямой задачи совпадает с максимумом целевой функции двойственной задачи. Поскольку число переменных в двойственной задаче равно числу условий-ограничений в прямой задаче, в ряде случаев двойственную задачу решить проще.

Получим необходимые и достаточные условия оптимальности решения задачи выпуклого программирования на основе теории двойственности. В этом случае изменится только форма необходимых и достаточных условий (в них не будут включены производные), но предпосылки для формулирования условий оптимальности в обоих случаях одинаковы.

Для примера рассмотрим задачу *квадратичного программирования*:

$$f(x) = \frac{1}{2}(Cx, x) + (d, x) \rightarrow \min$$

при $(a_i, x) \leq b_i$, $i = 1, 2, \dots, k$, и $(a_i, x) = b_i$, $i = k + 1, \dots, m$, где C — положительно определенная симметрическая матрица размерности $n \times n$; d, a_1, \dots, a_m — заданные векторы из \mathbb{R}^n ; b_1, \dots, b_m — заданные числа.

Функция Лагранжа задачи имеет вид

$$\begin{aligned} L(x, y) &= \frac{1}{2}(Cx, x) + (d, x) + \sum_{i=1}^m y_i((a_i, x) - b_i) = \\ &= \frac{1}{2}(Cx, x) + (d, x) + (y, Ax - b) = \frac{1}{2}(Cx, x) + (d + yA, x) - (y, b). \end{aligned}$$

Задачей, двойственной к исходной задаче квадратичного программирования, будет следующая:

$$\varphi(y) \rightarrow \max, \quad y \in Y,$$

где

$$\varphi(y) = \inf_{x \in \mathbb{R}^n} L(x, y) = \inf_{x \in \mathbb{R}^n} \left(\frac{1}{2}(Cx, x) + (d + yA, x) - (y, b) \right),$$

$$Y = \{y \in Q \mid \varphi(y) > -\infty\}.$$

Для положительно определенной матрицы C производная функции Лагранжа $L'_x(x, y) = Cx + d + yA$ равна нулю в точке $x(y) = -C^{-1}(d + yA)$, и двойственная задача записывается в виде

$$\varphi(y) \rightarrow \max, \quad y \in Q,$$

где

$$\begin{aligned} \varphi(y) &= L(x(y), y) = -\frac{1}{2}(C^{-1}(d + yA), d + yA) - (y, b) = \\ &= -\frac{1}{2}([AC^{-1}A^T]y, y) - (AC^{-1}d + b, y) - \frac{1}{2}(C^{-1}d, d). \end{aligned}$$

Полученная функция квадратична, в задаче учтены условия неотрицательности первых k переменных. Решить двойственную задачу гораздо проще. Если в исходной задаче нет ограничений-неравенств, то двойственная задача приводит к безусловной оптимизации квадратичной функции.

Пара $(x^*, y^*) \in P \times Q$ называется *седловой точкой* функции $L(x, y)$ на $P \times Q$, если

$$L(x^*, y^*) = \min_{x \in P} L(x, y^*),$$

$$L(x^*, y^*) = \max_{y \in Q} L(x^*, y),$$

т. е. $L(x, y^*) \geq L(x^*, y^*) \geq L(x^*, y)$ при всех $x \in P$, $y \in Q$.

Тогда точка $x^* \in P$ является решением прямой задачи в том и только в том случае, если существует вектор $y^* \in Q$ такой, что пара (x^*, y^*) является седловой точкой функции Лагранжа $L(x, y)$ на $P \times Q$. Таким образом, если одновременно решать прямую и двойственную задачи, то к точке минимума (к решению) мы можем приближаться двумя способами.

Вектор Куна — Таккера имеет различные экономические интерпретации. Рассмотрим две из них.

1. Пусть предприятие выпускает некую продукцию и стремится получить максимальный доход. Условия-ограничения в виде неравенств характеризуют затраты ресурсов при выпуске продукции. Очевидно, что в процессе выпуска продукции один или несколько ресурсов будут исчерпаны полностью (активные ограничения). Другая часть ресурсов будет не использована (пассивные ограничения). В двойственной задаче (согласно условию дополняющей нежесткости) для активных ограничений $u_i^* \neq 0$, $i = 1, 2, \dots, k$, а для пассивных ограничений $u_i^* = 0$, $i = k + 1, \dots, m$ (эти ресурсы недефицитны), т. е. предприятию следует закупать в первую очередь те ресурсы, для которых u_i^* , $i = 1, 2, \dots, k$, имеют наибольшие значения. Если есть возможность увеличивать количество всех ресурсов одновременно, то их желательно приобретать в пропорциях, описываемых вектором Куна — Таккера.

2. Предприятию необходимо продать «ненужную» часть сырья и установить за него такую цену, чтобы максимизировать общий доход. Пусть c — заданный вектор, элементами которого являются цены на сырье. Предприятие стремится продать сырье по таким ценам, чтобы получить такую же прибыль, как и в случае, если бы из проданного сырья была изготовлена продукция, а для этого координаты вектора c должны быть равны координатам вектора Куна — Таккера y^* .

§ 1.7. Графическое решение задач математического программирования

Самыми наглядными методами решения задач математического программирования являются графические. Но они приемлемы только для функций двух и иногда трех переменных.

Рассмотрим два примера.

Пример 1. Минимизировать функцию

$$f(x) = |x_1 - 2| + |x_2 - 2|$$

при ограничениях

$$g(x) = x_1 - x_2^2 \geq 0, \quad h(x) = x_1^2 + x_2^2 - 1 = 0.$$

Получили задачу *нелинейного* математического программирования. Прежде всего построим по условиям-ограничениям допустимую область D — множество точек (x_1, x_2) , удовлетворяющих ограничениям задачи. Ограничение $g(x) \geq 0$ определяет область «внутри» параболы $x_1 = x_2^2$,

ограничение $h(x) = 0$ — окружность единичного радиуса с центром в начале координат (рис. 1.9). Допустимая область D этой задачи — дуга ABC окружности. Чтобы найти точку, в которой функция $f(x)$ принимает минимальное значение на допустимой области D , построим линии уровня $f(x)$, обозначенные штриховыми линиями. В точке $(2, 2)$ $f(x) = 0$; при $f(x) = 1$ и $f(x) = 2$ линии уровня образуют квадраты. Градиент функции $\nabla f(x)$ направлен в сторону дуги ABC , и функция $f(x)$ будет иметь минимальное значение в точке касания линии уровня к дуге ABC . Поскольку линии уровня отсекают на осях x_1 и x_2 равные отрезки, то координаты точки касания равны $1 \cdot \cos(\pi/4) = 1 \cdot \sin(\pi/4) = \sqrt{2}/2$. Получили решение задачи:

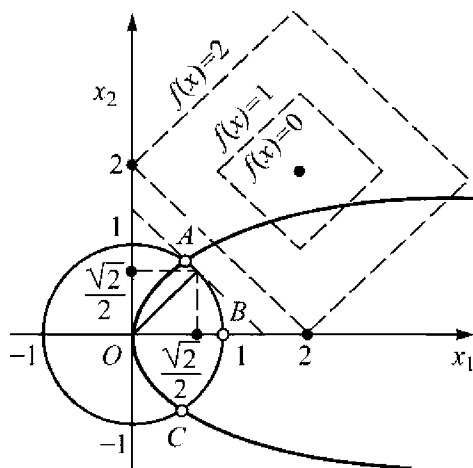


Рис. 1.9. Графическое решение задачи математического программирования

$$x_1^* = \frac{\sqrt{2}}{2}, \quad x_2^* = \frac{\sqrt{2}}{2}, \quad f_{\min}(x) = f\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right) = 2 \left| \frac{\sqrt{2}}{2} - 2 \right| \approx 2,6.$$

Нетрудно видеть, что то же решение будет и в том случае, если вместо условия $h(x) = 0$ рассмотреть условие $g_1(x) = x_1^2 + x_2^2 - 1 \leq 0$. Тогда допустимая область D будет заключена между дугами ABC и AOC (см. рис. 1.9). Однако минимальное значение функции $f(x)$ в области D будет достигнуто также в точке x^* с координатами $x_1^* = \sqrt{2}/2$, $x_2^* = \sqrt{2}/2$.

Пример 2. Пусть в задаче о питании (см. § 1.1)

$$\begin{aligned} c_1 = 2, \quad c_2 = 3, \quad a_{11} = 1, \quad a_{12} = 5, \quad a_{21} = 3, \quad a_{22} = 2, \\ a_{31} = 2, \quad a_{32} = 4, \quad a_{41} = 2, \quad a_{42} = 2, \quad a_{51} = 1, \quad a_{52} = 0, \\ b_1 = 10, \quad b_2 = 12, \quad b_3 = 16, \quad b_4 = 6, \quad b_5 = 1. \end{aligned}$$

Получили задачу линейного программирования: минимизировать функцию

$$f(x) = 2x_1 + 3x_2$$

при ограничениях:

$$\begin{aligned} \text{а) } x_1 + 5x_2 &\geq 10; & \text{б) } 3x_1 + 2x_2 &\geq 12; & \text{в) } 2x_1 + 4x_2 &\geq 16; \\ \text{г) } 2x_1 + 2x_2 &\geq 6; & \text{д) } x_1 &\geq 1; & \text{е) } x_1 &\geq 0, \quad x_2 &\geq 0. \end{aligned}$$

Построим область, определяемую неравенствами-ограничениями задачи (рис. 1.10). Строим сначала прямую $x_1 + 5x_2 = 10$ по двум точкам: $x_2 = 2$ при $x_1 = 0$, $x_1 = 10$ при $x_2 = 0$. Нанесем точки $(0, 2)$ и $(10, 0)$ на график и проведем прямую AB . Чтобы установить, какая часть плоскости определяется неравенством $x_1 + 5x_2 \geq 10$, подставим в него координаты точки $(0, 0)$. Получим

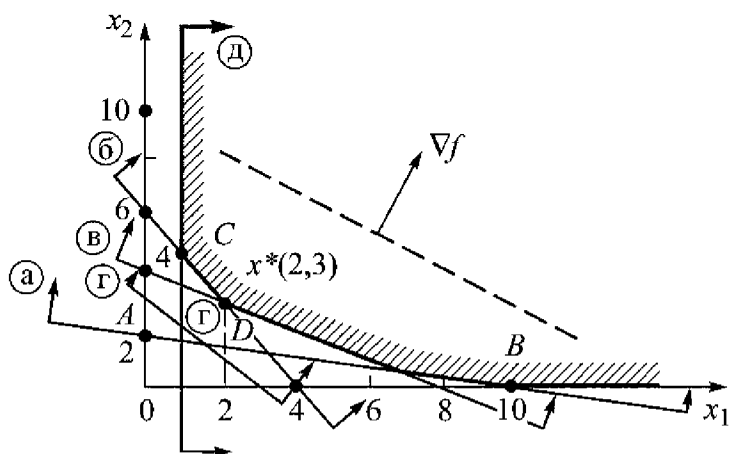


Рис. 1.10. Графическое решение задачи линейного программирования

$0 \geq 10$ — противоречие, т. е. неравенство определяет полуплоскость, не содержащую точку $(0, 0)$. Стрелки, направленные от прямой AB , указывают эту полуплоскость. Аналогично строим области, соответствующие другим неравенствам. Неравенство $x_1 \geq 0$ можно сразу исключить, так как оно «поглощается» неравенством $x_1 \geq 1$. Допустимой областью D является заштрихованная выпуклая неограниченная многоугольная область. Чтобы найти оптимальную точку, построим одну из линий уровня целевой функции и ее градиент. Пусть $c = 30$, т. е. $2x_1 + 3x_2 = 30$ (штриховая линия), а градиент ∇f имеет координаты $(2, 3)$. Поскольку требуется определить минимальное значение $f(x)$ на области допустимых значений, то перемещаем линию уровня параллельно самой себе в направлении антиградиента до тех пор, пока она будет находиться в допустимой области D . Точка «выхода» линии уровня из допустимой области является той точкой, где $f(x)$ примет минимальное значение: $x^* = (2, 3)$; $f_{\min}(x^*) = 2 \cdot 2 + 3 \cdot 3 = 13$.

На рис. 1.10 видно, что задачи математического программирования могут не иметь решения и могут иметь бесконечно много решений. Если бы при тех же ограничениях, какие были заданы в условии задачи, потребовалось максимизировать целевую функцию $f(x)$, то линию уровня пришлось бы перемещать в направлении градиента ∇f . Очевидно, в этом случае решения не существует, поскольку множество D не ограничено.

Теперь рассмотрим другую целевую функцию этой же задачи. Пусть требуется минимизировать функцию $f(x) = 3x_1 + 2x_2$. Очевидно, что линии уровня будут параллельны прямой $3x_1 + 2x_2 = 12$, т. е. линия уровня «выйдет» из допустимой области по отрезку прямой CD : все точки отрезка будут являться решениями задачи (бесконечное множество решений).

При решении задачи линейного программирования может оказаться, что ограничения противоречивы. Например, если ограничение г) записать в виде $2x_1 + 2x_2 \leq 6$, то это неравенство будет описывать полуплоскость, включающую точку $(0, 0)$, не имеющую общих точек с другими полуплоскостями (с решением неравенств а)–в)). Решение задачи в данном случае также не существует.

§ 1.8. Методы безусловной оптимизации

Задачи математического программирования в некоторых случаях можно свести к *безусловной минимизации* целевой функции $f(x)$, т. е. поиску локального минимума этой функции. Если локальный минимум на каком-то этапе не является и глобальным, то поиск глобального минимума и других локальных минимумов должен быть продолжен, при этом выбирают другие начальные точки итерационных процессов.

Как правило, изложенные алгоритмы применяют для нахождения точек минимума положительно определенной квадратичной формы n -го порядка. Для других видов функций указываются условия их применимости.

Методы, использующие только значения функции

Пусть задана целевая функция (функция качества) $f(x)$, производные которой могут быть разрывными либо не вычисляться явно. Такая ситуация возможна, например, если значения функции $f(x)$ заданы в табличной форме. В этом случае рассматривают два подхода: методы поиска и методы сопряженных направлений.

В методах поиска в основном используются одномерные пробные шаги. Эффективность методов поиска возрастает с увеличением степени сепарабельности функции задачи. Например, задача минимизации функции $f(x) = x_1^4 + 10000x_2^4$ методами поиска решается в два этапа несмотря на то, что линии уровня этой функции очень вытянуты (образуют овраг) и движение по любому направлению, не параллельному главной оси, весьма затруднено.

Методы сопряженных направлений обеспечивают достижение минимума положительно определенной квадратичной формы размерностью n за n^2 шагов. Направления векторов s_i и s_j называются *сопряженными относительно матрицы G* , если

$$s_i^T G s_j = 0, \quad i \neq j.$$

В частности, метод покоординатного спуска является одним из методов сопряженных направлений. При использовании этого метода точка минимума находится последовательно по одной из координат, когда другим координатам заданы фиксированные значения. Для квадратичной функции порядка n этим методом можно

достичь точку минимума за n итераций, если на одной итерации определяется точка минимума по одной координате.

Рассмотрим один из методов безусловной минимизации функций, использующий только значения функции $f(x)$. В этом методе применяется процедура аппроксимации матрицы вторых частных производных и градиента функции $f(x)$ за $0,5n(n+1)$ итераций. Для положительно определенной квадратичной формы этот метод является точным. Фактически здесь реализуется метод Ньютона (см. далее).

Пусть $f(x)$ — квадратичная форма, G — ее матрица вторых частных производных $\nabla^2 f(x)$, $\nabla f(x)$ — градиент функции $f(x)$, $\sigma_k = x^{k+1} - x^k$, где x^k — точка, полученная на k -й итерации. Тогда значение функции в точке x^{k+1} вычислим по формуле

$$f(x^{k+1}) = f(x^k) + \nabla^T f(x^k) \sigma_k + \frac{1}{2} \sigma_k^T G \sigma_k. \quad (1.11)$$

Предположим, что для перехода от x^k к x^{k+1} оптимизация ведется по направлению вектора σ_k , т. е.

$$\nabla^T f(x^{k+1}) \sigma_k = 0. \quad (1.12)$$

Таким образом,

$$f(x^k) = f(x^{k+1}) + \frac{1}{2} \sigma_k^T G \sigma_k. \quad (1.13)$$

В силу равенств (1.11) и (1.13) имеем

$$\nabla^T f(x^k) \sigma_k = -\sigma_k^T G \sigma_k$$

или

$$f(x^{k+1}) = f(x^k) - \frac{1}{2} \sigma_k^T G \sigma_k. \quad (1.14)$$

С помощью формулы (1.14) можно получить диагональные элементы матрицы G , считая первые n направляющих векторов равными $(1, 0, \dots, 0), \dots, (0, 0, \dots, 1)$.

Пусть x^0 — начальная точка, а векторы s_{i-1} при $i = 1, 2, \dots, n$ имеют координаты, равные единице на i -м месте и нулю на остальных местах. Тогда $x^{i+1} = x^i - \lambda_i s_i$, где число λ_i выбирается так, чтобы минимизировать функцию $f(x)$ по лучу s_{i-1} , выходящему из точки x^{i-1} ; значения λ_i , $i = 1, 2, \dots, n$, могут иметь любой знак. Диагональные элементы вычисляются по формуле

$$g_{i+1,i+1} = \frac{-2(f(x^{i+1}) - f(x^i))}{\lambda_i^2}, \quad i = 0, 1, \dots, n-1. \quad (1.15)$$

Для вычисления недиагональных элементов матрицы G поступают следующим образом. Пусть $x^k = x^n$ (значение x^n получено после n шагов, совершенных для вычисления диагональных элементов). Для $i = 1, 2, \dots, n$ и $j = i+1, \dots, n$ положим

$$s_{ij} = e_i + e_j,$$

где e_i — единичный вектор с единицей на i -м месте.

Выберем λ_{ij} так, чтобы функция $f(x)$ достигла минимума по направлению вектора s_{ij} , выходящему из точки x^k . Тогда

$$g_{ij} = \frac{2(-f(x^{k+1}) + f(x^k)) - \lambda_{ij}^2 g_{ii} - \lambda_{ij}^2 g_{jj}}{2\lambda_{ij}^2}, \quad (1.16)$$

$$i = 1, 2, \dots, n, \quad j = i+1, \dots, n.$$

Если квадратичная форма достигла минимума в выбранном направлении, то получим соответствующее $\lambda_{ij} = 0$. В этом случае можно сделать малый произвольный шаг и продолжать считать по приведенным формулам.

Поскольку после очередного шага в i -м направлении движения $S_{i-1} = S_i$, то вычисление градиента проведем по следующим формулам:

$$\frac{\partial f(x^i)}{\partial x_i} = 0, \quad \frac{\partial f(x^n)}{\partial x_i} = \frac{\partial f(x^i)}{\partial x_i} + \sum_{j=1}^n g_{ij}(x_j^n - x_j^i),$$

если только $f(x)$ — квадратичная форма, при этом учитываются условия

$$x_j^n = x_j^i, \quad j = 1, 2, \dots, i; \quad x_j^i = x_j^0, \quad j = i+1, \dots, n.$$

Следовательно,

$$\frac{\partial f(x^n)}{\partial x_i} = \sum_{j=i+1}^n g_{ij}(x_j^n - x_j^0). \quad (1.17)$$

Градиент в точке x^n можно получить, если вычислены все элементы матрицы G . Для любой другой точки y

$$\nabla f(y) = \nabla f(x^n) + G(y - x^n). \quad (1.18)$$

После того как матрица G вычислена, по формуле (1.18) можно получить $\nabla f(y)$ и вычислить направление $-G^{-1}\nabla f(y)$. Точка $y - G^{-1}\nabla f(y)$ будет безусловным минимумом положительно определенной квадратичной формы.

Пример. Необходимо решить следующую задачу:

$$f(x) = (1, 1)x + \frac{1}{2}x^T \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} x \rightarrow \min.$$

Определим диагональные элементы матрицы G .

Положим $x^0 = (0, 0)$; тогда $f(x^0) = 0$. Пусть $s_0 = (1, 0)$. Тогда $\lambda_0 = -1/2$, $\sigma_0 = (-1/2, 0)^T$, $x^1 = (-1/2, 0)$, $f(x^1) = -1/4$. По формуле (1.15) имеем $g_{11} = 2$.

Пусть $s_1 = (1, 0)$. Тогда получим $\lambda_1 = -1/2$, $\sigma_1 = (0, -1/2)^T$, $x^2 = (-1/2, -1/2)$, $f(x^2) = -3/8$. По формуле (1.15) имеем $g_{22} = 1$.

Переходим к вычислению недиагональных элементов матрицы G .

Пусть $s_2 = (1, 1)$. Тогда $\lambda_3 = 1/10$, $x^3 = (-2/5, -2/5)$, $f(x^3) = -2/5$. По формуле (1.16) имеем $g_{12} = -1$. Все элементы матрицы G найдены.

Для вычисления градиента ∇f^2 воспользуемся формулой (1.17):

$$\frac{\partial f(x^2)}{\partial x_1} = g_{12}(x_2^2 - x_2^0) = 1 \cdot \left(-\frac{1}{2} + 0\right) = -\frac{1}{2}, \quad \frac{\partial f(x^2)}{\partial x_2} = 0.$$

По формуле (1.18) имеем

$$\nabla f(x^3) = \begin{pmatrix} -\frac{1}{2} \\ 2 \\ 0 \end{pmatrix} + \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \left[\begin{pmatrix} -\frac{2}{5} \\ -\frac{2}{5} \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \right] = \begin{pmatrix} -\frac{1}{5} \\ 1 \\ \frac{1}{5} \end{pmatrix}.$$

Это значение точное. Теперь получим точки искомого безусловного минимума:

$$x^3 - G^{-1}\nabla f(x^3) = \begin{pmatrix} -\frac{2}{5} \\ 2 \\ -\frac{2}{5} \end{pmatrix} - \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} -\frac{1}{5} \\ \frac{1}{5} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

Задача решена.

Методы, использующие первые производные

В методах минимизации функции $f(x)$, в которых используют первые производные, учитываются только линейные члены разложения функции $f(x)$ в ряд Тейлора. Поэтому эти методы иногда называют *линейными методами*.

1. *Градиентный метод*. Численным методом безусловной минимизации функций является градиентный метод с алгоритмом

$$x^{i+1} = x^i - \lambda_i \frac{\partial f(x^i)}{\partial x}, \quad i = 0, 1, 2, \dots, \quad (1.19)$$

где x^0 — начальное приближение, λ_i — шаговый множитель на i -й

итерации, $\frac{\partial f(x^i)}{\partial x} \equiv \nabla f(x^i)$ — градиент функции $f(x)$ в точке x^i .

Очередное приближение x^{i+1} получается из предыдущего x^i путем движения в направлении *антиградиента* (в направлении наиболее быстрого убывания функции $f(x)$ в окрестности точки x^i).

Наиболее широко распространены две модификации градиентного метода:

1) простой градиентный метод, где шаговый множитель λ или остается постоянным на протяжении всей итерационной процедуры, или изменяется через какое-то число итераций;

2) метод наискорейшего спуска, в котором на каждой итерации шаговый множитель выбирается из условия минимума функции в направлении антиградиента, т. е. минимальное значение $\lambda_i > 0$ выбирается из условия

$$f\left(x^i - \lambda_i \frac{\partial f(x^i)}{\partial x}\right) = \min_{\lambda > 0} f\left(x^i - \lambda \frac{\partial f(x^i)}{\partial x}\right), \quad i = 0, 1, 2, \dots$$

В этой процедуре соседние градиенты ортогональны, т. е.

$$\nabla^T f(x^{i+1}) \nabla f(x^i) = 0 \quad \text{для всех } i.$$

Доказано, что всякая предельная точка \bar{x} последовательности $\{x^i\}$ стационарна, т. е. $\nabla f(\bar{x}) = 0$.

Выбор шагового множителя λ_i можно проводить и другими методами, в частности методом дробления. Если вектор s_k указывает направление убывания значений функции $f(x)$, то дробление шагового множителя осуществляется следующим образом.

Выбираются некоторые константы $\beta > 0$, $0 < \alpha < 1$ (часто $\alpha = 1/2$). При $\lambda = \beta$ проверяется условие

$$f(x^k + \lambda s_k) < f(x^k).$$

Если оно выполнено, то полагают $\lambda_k = \beta$. Если не выполнено, производится дробление шагового множителя, т. е. принимают $\lambda_k = \alpha\beta$. Далее вновь проверяется выполнение приведенного условия. Процесс дробления продолжается до тех пор, пока приведенное условие выполнится. Этот процесс не может быть бесконечным, поскольку вектор s_k указывает направление убывания значений функции $f(x)$.

Если проверяемое в процессе дробления шагового множителя условие оказывается выполненным на первом шаге при значении $\lambda = \beta$, то иногда бывает полезно увеличить шаг, положив $\lambda = \mu\beta$, где $\mu > 1$. Может оказаться, что умножение на μ следует повторить несколько раз. Последнее значение λ , при котором произошло уменьшение значения функции $f(x)$, и принимают за λ_k .

Градиентный метод имеет следующие недостатки, затрудняющие его применение на практике.

1. При минимизации положительно определенной квадратичной формы этот метод, вообще говоря, бесконечен.

2. Каждая итерация выполняется независимо от других, т. е. информация не накапливается и не используется для увеличения скорости сходимости итерационного процесса.

3. Скорость сходимости итерационного процесса во многом зависит от вида функции $f(x)$. Если отношение наибольшего собственного значения матрицы G вторых частных производных функции к наименьшему (коэффициент обусловленности матрицы G) в некоторой точке минимума велико (овражная функция), то траектория наискорейшего спуска в окрестности такой точки состоит из коротких зигзагообразных кусков. Возможно, понадобится сделать тысячи таких же шажков, прежде чем будет достигнута достаточная близость к предельной точке. Если коэффициент обусловленности матрицы G близок к единице, то линии (поверхности) уровня функции $f(x)$ принимают вид окружности (сферы) и градиентный метод быстро сходится к точке минимума. Поэтому в ряде случаев целесообразно перейти к новой системе координат, чтобы поверхности (линии) уровня функции $f(x)$ приняли вид, близкий к виду сфер (окружностей).

Для увеличения скорости сходимости и для минимизации овражных функций применяется метод «тяжелого шарика»:

$$x^{i+1} = x^i - \lambda_i \nabla f(x^i) + \beta(x^i - x^{i-1}),$$

где коэффициент β подбирается в числовом эксперименте, $0 < \beta < 1$.

2. *Метод сопряженных направлений.* Как уже отмечалось ранее, векторы s_i и s_j называются сопряженными относительно матрицы G , если $s_i^T G s_j = 0$, $i \neq j$.

Пусть функция $f(x)$ имеет вид

$$f(x) = b^T x + \frac{1}{2} x^T G x, \quad (1.20)$$

где G — положительно определенная матрица, b — вектор коэффициентов при линейных членах.

Пусть s_0, \dots, s_{n-1} — ненулевые векторы и числа λ_i таковы, что

$$f(x^{k+1}) = f\left(x^0 + \sum_{i=0}^{k-1} \lambda_i s_i + \lambda_k s_k\right)$$

есть минимум функции $f(x)$ в направлении вектора s_k , если начинать движение от точки

$$x^k = x^0 + \sum_{i=0}^{k-1} \lambda_i s_i, \quad k = 1, 2, \dots, n-1.$$

Точка x^n будет безусловным минимумом функции $f(x)$ на всем пространстве, если s_0, s_1, \dots, s_{n-1} — сопряженные векторы.

Теперь задача состоит в подборе подходящих сопряженных векторов. В этом случае положительно определенная квадратичная форма размерности n минимизируется за n или менее шагов.

Пусть x^0 — начальная точка и $s_0 = -\nabla f(x^0)$. Обозначим через x^{i+1} точку минимума функции $f(x)$ на луче, выходящем из x^i в направлении вектора s_i .

Положим

$$\beta_i = \frac{|\nabla f(x^{i+1})|^2}{|\nabla f(x^i)|^2}.$$

Новый вектор s_{i+1} вычисляется по формуле

$$s_{i+1} = -\nabla f(x^{i+1}) + \beta_i s_i.$$

Векторы s_i , $i = 0, 1, \dots, n-1$, оказываются сопряженными, если функция $f(x)$ задана в форме (1.20). Этот метод получил название *метода сопряженных градиентов*.

Алгоритм метода сопряженных градиентов можно представить в виде

$$\begin{aligned} x^{i+1} &= x^i - \lambda_i s_{i+1}, \quad i = 0, 1, 2, \dots, \\ s_1 &= \nabla f(x^0), \quad s_{i+1} = \nabla f(x^0) + \beta_i s_i, \\ \beta_i &= \frac{|\nabla f(x^i)|^2}{|\nabla f(x^{i-1})|^2}. \end{aligned}$$

Значение шагового множителя λ_i находится из условия

$$x f(x^i - \lambda_i s_{i+1}) = \min_{\lambda} f(x^i - \lambda s_{i+1}).$$

Анализ сходимости последовательности $\{x^i\}$ к точке минимума показывает, что метод сопряженных градиентов имеет примерно такую же область сходимости, что и метод наискорейшего спуска, но скорость сходимости — квадратичная. Метод сопряженных градиентов может быть применен к функциям произвольного вида. Однако в данном случае рекомендуется производить обновление направления вектора s_i либо через $n+1$ шагов, либо подобрать время обновления направления в числовом эксперименте, т. е. при обновлении направления в точке x^k вновь выбрать вектор $s_{k+1} = \nabla f(x^k)$.

Метод переменной метрики

Этот метод обеспечивает сходимость к точке минимума за n шагов для положительно определенной квадратичной формы порядка n .

Пусть x^0 — начальная точка, H_0 — произвольная положительно определенная матрица, являющаяся начальным приближением обратной матрицы вторых частных производных функции $f(x)$ в точке x^0 . Обозначим через s_i ненулевой направляющий вектор, полученный на i -й итерации, $s_i = -H_i \nabla f(x^i)$, $\sigma_i = \lambda_i s_i$, $x^{i+1} = x^i + \sigma_i$, вектор s_i является линейно независимым со всеми предыдущими векторами s_0, \dots, s_{i-1} . Общее требование к векторам s_i следующее:

$$\nabla^T f(x^i) s_i < 0 \quad \text{для всех } i.$$

Здесь λ_i выбирают так, чтобы минимизировать функцию $f(x)$ в направлении вектора s_i , выходящего из точки x^i . Поскольку матрица H_i положительно определена, то шаговый множитель λ_i должен быть больше нуля, если только x^i не является точкой минимума функции $f(x)$.

Пусть $y_i = \nabla f(x^{i+1}) - \nabla f(x^i)$. Тогда очередное приближение обратной матрицы вторых частных производных функции $f(x)$ вычисляется по формуле

$$H_{i+1} = H_i + \frac{\sigma_i \sigma_i^T}{\sigma_i^T y_i} - \frac{H_i y_i y_i^T H_i}{y_i^T H_i y_i}.$$

Если матрица H_i положительно определена, то матрица H_{i+1} тоже положительно определена, а этим и обеспечивается убывание функции $f(x)$ на каждом шаге. Векторы $\sigma_0, \sigma_1, \dots, \sigma_{n-1}$ сопряжены относительно матрицы A , если функция $f(x)$ задана в виде (1.20). В методе переменной метрики существует большая свобода в выборе s_i и H_0 .

Рассмотрим модифицированный метод переменной метрики. В данном методе новое приближение для обратной матрицы вторых частных производных $f(x)$ имеет вид

$$H_{i+1} = H_i + (\sigma_i - H_i y_i)(y_i^T (\sigma_i - H_i y_i))^{-1} (\sigma_i^T - y_i^T H_i), \quad i = 0, 1, 2, \dots$$

Если $y_i^T (\sigma_i - H_i y_i) = 0$, то новое приближение для H_i не вычисляют, а только выбирают новые направления движения. Поэтому полагают, что $y_i^T (\sigma_i - H_i y_i) \neq 0$.

После n преобразований матрицы H выбирают направление вектора $s^n = -H_n \nabla f(x^n)$. Тогда x^{n+1} будет искомым безусловным минимумом положительно определенной квадратичной формы. В этом методе на каждой итерации добавляется только один член к текущей обратной матрице вторых частных производных функции $f(x)$.

Пример 1. Минимизируем функцию, являющуюся квадратичной формой:

$$f(x) = (1, 1)x + \frac{1}{2}x^T \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} x \rightarrow \min.$$

Пусть $x^0 = (0, 0)^T$, $H_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$. Тогда $\nabla f(x^0) = (1, 1)^T$.

Пусть $s_0 = (-1, 0)^T$. Получим

$$\lambda_0 = \frac{1}{2}, \quad \sigma_0 = \left(-\frac{1}{2}, 0\right)^T,$$

$$x^1 = (0, 0)^T + \left(-\frac{1}{2}, 0\right)^T = \left(-\frac{1}{2}, 0\right)^T, \quad \nabla f^1 = \left(0, \frac{1}{2}\right)^T,$$

$$y_0 = \left(0, \frac{1}{2}\right)^T - (1, 1)^T = \left(-1, -\frac{1}{2}\right)^T, \quad \sigma_0 - H_0 y_0 = \left(-\frac{1}{2}, 0\right)^T,$$

$$y_0^T(\sigma_0 - H_0 y_0) = \frac{1}{2}, \quad H_1 = \left(-\frac{1}{2}\right) 2 \left(-\frac{1}{2}, 0\right) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix}.$$

Пусть $s_1 = (0, -1)^T$. Тогда

$$\lambda_1 = \frac{1}{2}, \quad \sigma_1 = \left(0, -\frac{1}{2}\right)^T,$$

$$x^2 = \left(-\frac{1}{2}, 0\right)^T + \left(0, -\frac{1}{2}\right)^T = \left(-\frac{1}{2}, -\frac{1}{2}\right)^T, \quad \nabla f^2 = \left(-\frac{1}{2}, 0\right)^T,$$

$$y_1 = \left(-\frac{1}{2}, 0\right)^T - \left(0, \frac{1}{2}\right)^T = \left(-\frac{1}{2}, -\frac{1}{2}\right)^T,$$

$$\sigma_1 - H_1 y_1 = \begin{pmatrix} 0 \\ -\frac{1}{2} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ -\frac{1}{2} \end{pmatrix},$$

$$y_1^T(\sigma_1 - H_1 y_1) = \frac{1}{8},$$

$$H_2 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{4} \\ -\frac{1}{2} \end{pmatrix} 8 \begin{pmatrix} \frac{1}{4} & -\frac{1}{2} \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}.$$

Матрица H_2 равна обратной к матрице $\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$.

На последнем шаге получаем безусловный минимум

$$x^3 = \begin{pmatrix} -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} + \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

Решение получено.

Если известна какая-нибудь квадратная подматрица порядка $n-r$ матрицы вторых частных производных порядка n функции $f(x)$, то для минимизации положительно определенной квадратичной формы потребуется только $n+1$ шагов алгоритма переменной метрики (при выполнении предыдущих предположений).

Пусть, например, известны вторые частные производные

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1, 2, \dots, n-r.$$

Обозначим квадратную подматрицу матрицы, составленной из вторых частных производных, через \tilde{G} . Тогда

$$G \equiv \nabla^2 f = \begin{pmatrix} \tilde{G} & a \\ a^T & b \end{pmatrix}.$$

Имеем

$$G^{-1} = \begin{pmatrix} \tilde{G}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -\tilde{G}^{-1} a \\ I \end{pmatrix} (-a^T \tilde{G}^{-1} a + b)^{-1} (-a^T \tilde{G}^{-1}, I),$$

где I — единичная матрица. Здесь второе слагаемое есть матрица ранга r . Ее можно найти за r шагов. При известной матрице \tilde{G}^{-1} можно вычислить G^{-1} .

Единственное отличие этого метода от прежней вычислительной процедуры состоит в том, что матрица H_0 берется в виде

$$H_0 = \begin{pmatrix} \tilde{G}^{-1} & 0 \\ 0 & 0 \end{pmatrix}.$$

Переход от матрицы H_k к H_{k+1} выполняется, как и прежде, но теперь H_k положительно полуопределена и не станет положи-

тельно определенной до тех пор, пока не будет полностью получена обратная матрица.

Достоинство метода переменной метрики состоит в том, что с его помощью удастся решать задачи большой размерности, структура которых такова, что можно вычислить вторые частные производные лишь для части переменных, а не для всех. Для задач большой размерности возможность провести минимизацию менее чем за n шагов существенна.

Пример 2. Минимизируем ту же функцию, что и в предыдущем примере:

$$f(x) = (1, 1)x + \frac{1}{2}x^T \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} x \rightarrow \min.$$

Допустим, что диагональный элемент g_{11} матрицы G известен и равен 2.

Пусть $H_0 = \begin{pmatrix} 1/2 & 0 \\ 0 & 0 \end{pmatrix}$, $x^0 = (1, -2)^T$. Тогда $f(x^0) = (1, 0)^T$.

Если $s_0 = (-1, -1)^T$, то

$$\lambda_0 = \frac{1}{5}, \quad \sigma_0 = \left(-\frac{1}{5}, -\frac{1}{5}\right)^T,$$

$$x^1 = (1, -2)^T + \left(-\frac{1}{5}, -\frac{1}{5}\right)^T = \left(\frac{4}{5}, -\frac{11}{5}\right)^T, \quad \nabla f^1 = \left(\frac{2}{5}, -\frac{2}{5}\right)^T,$$

$$y_0 = \left(\frac{2}{5}, -\frac{2}{5}\right)^T - (1, 0)^T = \left(-\frac{3}{5}, -\frac{2}{5}\right)^T, \quad \gamma_i = y_i^T (\sigma_i - H_i y_i),$$

$$\sigma_0 - H_0 y_0 = \begin{pmatrix} -\frac{1}{5} \\ -\frac{1}{5} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -\frac{3}{5} \\ -\frac{2}{5} \end{pmatrix} = \begin{pmatrix} -\frac{1}{10} \\ -\frac{2}{10} \end{pmatrix}, \quad \gamma_0 = \frac{1}{50}.$$

Найдем обратную матрицу:

$$H_1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{10} \\ \frac{2}{10} \end{pmatrix} \frac{1}{50} \begin{pmatrix} \frac{1}{10} & -\frac{2}{10} \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}.$$

Отсюда видно, что уже после первой итерации мы получили искомую обратную матрицу.

Использование вторых частных производных

Рассмотрим обобщенный метод Ньютона, в котором используются вторые частные производные минимизируемой функции $f(x)$, т. е. учитывается дополнительная информация о форме поверхности $f(x)$. Предположим, что матрица вторых частных производных $G(x) \equiv \nabla^2 f(x)$ невырождена. Итерационный процесс отыскания точки минимума функции $f(x)$ имеет вид

$$x^{i+1} = x^i - \lambda_i G^{-1}(x^i) \nabla f(x^i), \quad (1.21)$$

где шаговый множитель $\lambda_i > 0$ выбран так, чтобы минимизировать $f(x)$ по направлению вектора $-G^{-1}(x^i) \nabla f(x^i)$ от точки x^i . Если $\lambda_i \equiv 1$, то получим «чистый» метод Ньютона. Идея метода состоит в следующем: функция $f(x)$ заменяется двумя первыми членами ее разложения в ряд Тейлора и полученная квадратичная форма минимизируется.

Если $f(x)$ — положительно определенная квадратичная форма, то итерационный метод (1.21) при $\lambda_i \equiv 1$ позволяет достичь минимум за один шаг. Если $f(x)$ — произвольная выпуклая функция, то итерационный процесс (1.21) гарантирует ее монотонное убывание от итерации к итерации.

Однако метод Ньютона имеет следующие недостатки:

1) не всегда существует матрица, обратная к $G(x)$;
 2) в невыпуклых функциях, т. е. когда матрица $G(x)$ не является положительно определенной, не гарантировано монотонное убывание функции, если точка x^i не близка к точке минимума; тогда получим $\lambda_i = 0$ и процесс остановится в точке x^i ;

3) для некоторых функций с непрерывными вторыми частными производными бывает сложно аналитически вычислять производные.

В модифицированном методе Ньютона эти недостатки пытаются устранить. Направляющий вектор s_i вычисляется двумя способами. В обоих случаях $x^{i+1} = x^i + \lambda_i s_i$, причем шаговый множитель λ_i выбран наименьшим из всех $\lambda \geq 0$, для которых точка $x^i + \lambda_i s_i$ является локальным минимумом функции $f(x^i + \lambda s_i)$.

Эти два способа выбора вектора s_i следующие:

1) если матрица $G(x^i)$ имеет отрицательное собственное значение, то s_i — такой вектор, для которого

$$s_i^T G(x^i) s_i < 0, \quad s_i^T \nabla f(x^i) \leq 0; \quad (1.22)$$

2) если все собственные значения матрицы $G(x^i)$ больше или равны нулю, то выбираем s так, чтобы было либо

$$G(x^i) s = 0, \quad s^T \nabla f(x^i) < 0, \quad (1.23)$$

либо

$$G(x^i) s = -\nabla f(x^i). \quad (1.24)$$

Одновременно условия (1.23) и (1.24) выполняться не могут.

Единственным случаем, когда с помощью правил 1) и 2) нельзя указать ненулевой направляющий вектор s , является тот, когда $G(x^i)$ — положительная полуопределенная матрица и $\nabla f(x^i) = 0$, т. е. когда мы находимся в точке, удовлетворяющей необходимым условиям первого и второго порядка безусловного локального минимума функции $f(x)$.

Для невыпуклой функции несколько итераций по направлению вектора, удовлетворяющего условию (1.22), могут привести к значительному ускорению сходимости процесса минимизации.

Однако правила 1) и 2) не гарантируют, что полученная последовательность $\{x^i\}$ будет иметь предельные точки, удовлетворяющие необходимым условиям минимума.

В общем случае алгоритм модифицированного метода Ньютона имеет следующий вид.

1. Приводим матрицу $G(x^i)$ к виду

$$G(x^i) = L_i D_i L_i^T,$$

где L_i — невырожденная нижняя треугольная матрица, D_i — диагональная матрица.

2. Если все диагональные элементы матрицы D_i положительны, то берем

$$s_i = -G^{-1}(x^i) \nabla f(x^i).$$

3. Если некоторые диагональные элементы матрицы D_i отрицательны, то решаем уравнение $L_i^T t = a_i$, где a_i — вектор-столбец, j -я компонента которого равна нулю, когда j -й диагональный элемент матрицы D_i больше нуля, и равна единице, когда j -й диагональный элемент матрицы D_i не больше нуля. Положим $s_i = t$ при $t^T \nabla f(x^i) \leq 0$ и $s_i = -t$ в остальных случаях. Отметим, что s_i удовлетворяет условию (1.22).

4. Если все диагональные элементы матрицы D_i неотрицательны и по крайней мере один из них равен нулю, выбираем s по формуле (1.23) или (1.24).

Пример 1. Минимизировать функцию

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + \\ + 10,1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19,8(x_2 - 1)(x_4 - 1).$$

Теоретическое решение следующее:

$$x^* = (10, 10, 10, 10)^T, \quad f(x^*) = 0.$$

Берем $x^0 = (-3, -1, -3, -1)^T$. Обычные метод наискорейшего спуска и метод Ньютона приводили к решению, далекому от истинного:

$$x^* = (-1,07, 1,116, -0,86, 0,76)^T$$

при $f(x) = 7,89$.

В этой задаче матрица вторых частных производных имеет отрицательное собственное значение. Для решения модифицированным методом Ньютона потребовалось 24 итерации; на третьей и шестой итерациях использовалось условие (1.22), в остальных случаях применялся метод Ньютона.

Пример 2. Минимизировать функцию

$$f(x) = (x_1^4 - 3)^2 + x_2^4.$$

Пусть $x^0 = (10^{-3}, 10^2)$ — начальная точка.

Метод наискорейшего спуска приводит решение в окрестность точки $(0, 0)$.

В модифицированном методе Ньютона учитывается, что матрица

$$H_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -72 \cdot 10^{-6} & 0 \\ 0 & 12 \cdot 10^2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

имеет отрицательный диагональный элемент.

За счет выбора направляющего вектора движение идет перпендикулярно направлению оси x_2 . Первая итерация приводит к точке $x^1 = (\sqrt[4]{3}, 10)$, причем матрица вторых частных производных положительно определена. После восьми итераций получаем

$x^8 = (1,316, 1,535 \cdot 10^{-6})$, $f(x^8) = 8,08 \cdot 10^{-16}$. Точное решение следующее: $x^* = (\sqrt[4]{3}, 0)$, $f(x^*) = 0$.

Сравнение обычных методов наискорейшего спуска, сопряженных градиентов, переменной метрики, обобщенного метода Ньютона показывает, что наиболее «медленным» является метод наискорейшего спуска, наиболее «быстрым» — метод Ньютона. Так, в одном из примеров в процессе определения точки минимума функции обычного вида (не квадратичной формы) метод наискорейшего спуска не имел сходимости даже после 151 итерации, когда счет был прекращен; метод сопряженных градиентов сошелся за 40 итераций, метод переменной метрики — за 28, метод Ньютона — за 8. Объем вычислений, проводимых за одну итерацию, примерно одинаков для всех методов.

При минимизации овражных функций, когда велик коэффициент обусловленности для матрицы вторых производных минимизируемой функции, процесс минимизации «останавливается» на дне оврага, не достигая точки минимума. Пусть это будет точка A_1 . Затем из новой точки, удаленной от начального нулевого приближения, начинается новый процесс поиска точки минимума. Он «остановится» в точке A_2 . По прямой A_1A_2 в сторону точки с меньшим значением функции $f(x)$ делается шаг, т. е. задается шаговый множитель, значение которого выбирается экспериментально, но должно быть больше значения шагового множителя градиентного метода. Из полученной точки делается спуск в точку A_3 . Затем движение происходит по прямой A_2A_3 и т. д.

ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

§ 2.1. Математическая постановка задачи линейного программирования

В общем виде задачи математического программирования решить практически невозможно. Целесообразно рассматривать отдельные классы (виды) задач. Удастся сформулировать алгоритм решения, приемлемый только для определенного класса задач. Наиболее разработанными в математическом программировании являются методы решения задач линейного программирования (ЛП).

В задачах ЛП целевая функция линейна, а условия-ограничения содержат линейные равенства и линейные неравенства, в частности в них могут входить условия $x \geq 0$ или не входить. Примерами задач линейного программирования являются уже сформулированные в гл. 1 задача о питании, транспортная задача и т. п. Одна и та же задача ЛП может быть записана в различных формах. Говорят, что задача ЛП записана в *канонической форме*, если все ее ограничения, кроме $x_j \geq 0, j = 1, 2, \dots, n$, представляют собой *равенства*. Если все ограничения имеют вид неравенств, то задача записана в *стандартной форме*.

Для записи задачи линейного программирования в различных формах применяются следующие приемы.

1. Точку минимума функции $f(x)$ можно находить как точку максимума функции $-f(x)$.

2. Ограничения в виде неравенств

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

можно представить в виде равенств

$$\sum_{j=1}^n a_{ij}x_j - x_i = b_i,$$

используя новые переменные $x_i, i = n+1, n+2, \dots, m, x_i \geq 0$, называемые *слабыми*.

Для неравенства

$$\sum_{j=1}^n a_{ij} x_j \leq b_i$$

можно добавить слабую переменную x_i и получить равенство

$$\sum_{j=1}^n a_{ij} x_j + x_i = b_i.$$

3. Ограничение в виде равенства $\sum_{j=1}^n a_{ij} x_j = b_i$ можно заменить двумя неравенствами:

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

Если имеется m равенств $\sum_{j=1}^n a_{ij} x_j = b_i$, $i = 1, 2, \dots, m$, их можно заменить $m + 1$ неравенствами

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, 2, \dots, m, \quad \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} x_j - b_i \right) \leq 0.$$

4. Если на величину переменной x_j , $j = 1, 2, \dots, n$, не наложено условие неотрицательности, ее можно заменить двумя неотрицательными переменными x_j^+ и x_j^- , положив

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

Если имеется n таких переменных x_j , $j = 1, 2, \dots, n$, то их можно заменить $n + 1$ неотрицательными переменными x_j^1 и x_0 , положив $x_j = x_j^1 - x_0$.

Система ограничений в виде равенств и неравенств задает выпуклое множество — выпуклый многогранник. Это множество может быть ограниченным и неограниченным. Целевая функция задачи ЛП является выпуклой функцией. Таким образом, задача линейного программирования является частным случаем задачи выпуклого программирования.

Рассмотрим систему ограничений задачи ЛП в виде равенств

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m, \quad n > m. \quad (2.1)$$

Система (2.1) линейных уравнений называется *совместной*, если она имеет по крайней мере одно решение. Система (2.1) называется *избыточной*, если одно из уравнений можно выразить в виде линейной комбинации остальных. Система (2.1) *несовместна*, если ранг матрицы $\|a_{ij}\|$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, равен r , ранг *расширенной матрицы* этой системы (с присоединенным столбцом b_i) больше r .

В системе (2.1) число n переменных (неизвестных x) больше, чем число m уравнений. Предположим, что ранг этой системы равен m (система *неизбыточна*) и что система (2.1) совместна. Тогда m переменных из общего их числа образуют *базисные переменные*, а остальные $n - m$ переменных называют *свободными*. Система (2.1) в этом случае будет иметь бесконечно много решений, так как свободным переменным можно задавать любые значения, для которых находят значения базисных переменных. Решение системы (2.1) называют *базисным*, если все свободные переменные равны нулю. Если система уравнений имеет решение, то она имеет и базисное решение. Решение системы уравнений (2.1) называют *допустимым*, если все его компоненты неотрицательны. Если система линейных уравнений обладает допустимым решением, то она имеет и базисное допустимое решение. Совокупность всех допустимых решений системы (2.1) есть выпуклое множество, или, другими словами, множество решений задачи линейного программирования выпукло. Поскольку это множество образовано плоскостями (гиперплоскостями), то оно имеет вид выпуклого многогранника. *Базисное допустимое решение* соответствует крайней точке выпуклого многогранника (его грани или вершине).

Если существует оптимальное решение задачи линейного программирования, то существует *базисное оптимальное решение*.

Целевая функция задачи ЛП есть уравнение плоскости (или гиперплоскости для более трех переменных). Пусть в вершинах (крайних точках) выпуклого многоугольника мы установили «столбы», высота которых определяет значение целевой функции в данной вершине. На эти «столбы» наложим плоскость (графическое представление целевой функции). Очевидно, что максимальное и минимальное значения целевая функция задачи линейного программирования достигает либо в вершине выпуклого многогранника, либо на одной из его граней. Таким образом, решением (решениями) задачи ЛП является точка, находящаяся в вершине выпуклого многогранника. Значит, для того чтобы найти решение,

необходимо вычислять значения целевой функции в вершинах выпуклого многогранника, заданного условиями-ограничениями задачи.

§ 2.2. Симплекс-метод — основной метод решения задач линейного программирования

Рассмотрим задачу линейного программирования, записанную в канонической форме:

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

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m, \quad m \leq n, \quad x_j \geq 0, \quad j = 1, 2, \dots, n.$$

Предположим, что решение этой задачи существует. Чтобы найти оптимальное решение, необходимо найти допустимые базисные решения, а из них выбрать оптимальное базисное решение. Для этого мы должны поочередно из столбцов матрицы $\|a_{ij}\|$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, выбирать m столбцов и решать систему m уравнений с m неизвестными. Такой метод требует решения $C_n^m = \frac{n!}{m!(n-m)!}$ систем уравнений, что практически невозможно даже для небольших значений n .

В 1949 г. американский математик Дж. Данциг разработал симплекс-метод, ставший основным для решения задач линейного программирования.

Приведем пример использования симплекс-метода для решения следующей задачи. Минимизировать функцию

$$f(x) = 3 - x_4 + x_5$$

при ограничениях

$$x_2 + 2x_4 + 3x_5 - 7 = 0,$$

$$x_3 - x_4 - 3x_5 - 2 = 0,$$

$$x_1 + x_4 + x_5 - 2 = 0,$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0, \quad x_5 \geq 0.$$

Определитель матрицы, составленной из коэффициентов при неизвестных x_1, x_2, x_3 в условиях-ограничениях, имеет вид

$$\begin{vmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{vmatrix}$$

и не равен нулю. Поэтому ранг матрицы ограничений равен трем, базисными переменными являются x_1, x_2, x_3 , а свободными переменными — x_4, x_5 .

Выразим базисные переменные через свободные:

$$x_1 = 2 - x_4 - x_5,$$

$$x_2 = 7 - 2x_4 - 3x_5,$$

$$x_3 = 2 + x_4 + x_5.$$

Базисное решение $x_1 = 2, x_2 = 7, x_3 = 2$ (при нулевых значениях свободных переменных x_4 и x_5) в данном случае является допустимым (значения x_1, x_2, x_3 положительны). Значение целевой функции при таких значениях переменных есть $f(x) = 3$, но оно может быть уменьшено, если увеличить значение переменной x_4 , входящей с отрицательным коэффициентом. Очевидно, что увеличивать x_4 можно до тех пор, пока не будут нарушены условия-ограничения задачи, в частности пока переменные x_1, x_2 и x_3 будут неотрицательны. Например, если $x_4 = 2, x_5 = 0$, то $x_1 = 0, x_2 = 3, x_3 = 4$ — новое допустимое решение. При $x_5 = 0$ имеем $x_1 = 0$, если $x_4 = 2$; $x_2 = 0$, если $x_4 = 3,5$; $x_3 = 0$, если $x_4 = -2$. Чтобы ни одна из переменных x_1, x_2, x_3 не стала отрицательной, необходимо выбрать наименьшее положительное отношение элементов столбца свободных членов к соответствующим коэффициентам при x_4 . Если $x_4 = 2$, то значение x_1 становится равным нулю, таким образом x_1 переводим в свободные переменные, а $x_4 = 2$ — в базисные переменные. Ограничения и целевую функцию теперь необходимо выразить через x_1 и x_5 :

$$x_3 = 2 - x_1 - x_5,$$

$$x_2 = 3 + 2x_1 - x_5,$$

$$x_3 = 4 - x_1 - 2x_5,$$

$$f(x) = 3 - 2 + x_1 + x_5 + x_5 = 1 + x_1 + 2x_5.$$

В данном случае любое увеличение значений свободных переменных x_1 и x_5 ведет к увеличению (но не к уменьшению) значений целевой функции, т. е. получили оптимальное решение:

$$x_1 = 0, \quad x_5 = 0, \quad x_2 = 3, \quad x_3 = 4, \quad x_4 = 2,$$

$$f_{\min}(x) = 1.$$

Какие выводы можно сделать из этого примера? Во-первых, необходимо так разделить базисные и свободные переменные, чтобы получить допустимое базисное решение, а затем выразить базисные переменные и целевую функцию через свободные переменные. Во-вторых, по знаку коэффициентов при неизвестных в целевой функции следует определить:

а) достигнуто ли уже оптимальное решение (т. е. нет отрицательных коэффициентов);

б) значение какой переменной надо увеличить, т. е. какую переменную следует перевести в свободные.

Другими словами, определяя минимальное положительное отношение элементов столбца свободных членов к коэффициентам при новой свободной переменной, находим базисную переменную, которую необходимо перевести из базисных в свободные. После этого выражаем условия-ограничения и целевую функцию через новые свободные переменные.

Процесс повторяют до тех пор, пока не будет получено оптимальное решение. Если среди коэффициентов при неизвестных в целевой функции существует положительный, а все коэффициенты в условиях-ограничениях в соответствующем столбце неположительны, то задача линейного программирования не имеет оптимального решения, минимальное значение целевой функции равно $-\infty$.

Полученные результаты можно сформулировать в виде следующих теорем:

Теорема 1. Если множество допустимых решений задачи линейного программирования в стандартной форме не является пустым, то это множество содержит допустимое базисное решение.

Теорема 2. Множество допустимых решений задачи линейного программирования является выпуклым.

Теорема 3. Допустимые базисные решения задачи линейного программирования в стандартной форме являются крайними (угловыми) точками множества ее допустимых решений.

Теорема 4. Крайние (угловые) точки множества допустимых решений задачи линейного программирования в стандартной форме соответствуют допустимым базисным решениям.

Замечание. У множества допустимых решений задачи линейного программирования, где ранг системы $L < N$ числа переменных, число крайних точек, а следовательно число допустимых решений, конечно и не превышает $C_N^L = \frac{N!}{L!(N-L)!}$.

Теорема 5. Если в некоторой точке множества допустимых решений задачи линейного программирования ее целевая функция достигает оптимума, то она будет принимать оптимальное значение хотя бы в одной крайней точке множества допустимых решений.

Если целевая функция достигает оптимума в нескольких крайних точках множества допустимых решений, то она достигает оптимума и в любой их выпуклой комбинации.

Выпуклой комбинацией векторов X_K , $K = 1, \dots, m$, называется линейная комбинация $\lambda_1 X_1 + \dots + \lambda_m X_m$ этих векторов, коэффициенты λ_K которой удовлетворяют условиям $\lambda_K \geq 0$, $K = 1, \dots, m$,

$$\sum_{K=1}^m \lambda_K = 1.$$

Доказательства этих теорем приведены в учебнике: Волков И.К., Загоруйко Е.А. Исследование операций / Учебник для студентов высших технических учебных заведений. Под ред. В.С. Зарубина и А.П. Крищенко. М.: Изд. МГТУ им Н.Э. Баумана, 2009. 435 с.

В симплекс-методе *несовместность* (пустая допустимая область) выражается в том, что в оптимальной таблице среди базисных переменных присутствует слабая переменная, имеющая положительное значение.

Вырожденность и альтернативные оптимумы. В симплекс-методе крайняя точка *вырождена* тогда, и только тогда, когда один или более элементов столбца базисных переменных, равны нулю. Каждой невырожденной крайней точке соответствует одна симплекс-таблица, тогда как одной вырожденной точке могут соответствовать несколько разных таблиц.

В симплекс-методе может быть не одно оптимальное решение (альтернативные оптимумы).

Если крайняя точка оптимальной таблицы не вырождена, то альтернативные оптимумы существуют тогда, и только тогда, когда существует в строке целевой функции небазисный элемент, равный нулю.

Если крайняя точка оптимальной таблицы вырождена, то альтернативные оптимумы *могут существовать* тогда, когда существует в строке целевой функции небазисный элемент, равный нулю.

Любая оптимальная крайняя точка может быть получена исходя из другой крайней точки только с помощью операции замещения с ведущим элементом, проводимой с оптимальными базисами. Если крайняя точка оптимальной таблицы не вырождена, то в операции замещения для получения другой оптимальной точке используется в качестве вводимой та небазисная переменная, для которой существует в строке целевой функции небазисный элемент, равный нулю. Если в столбце для этой переменной невозможно найти ведущий элемент, то это означает, что рассматриваемой переменной отвечает неограниченное ребро.

Если крайняя точка оптимальной таблицы вырождена, то применяется та же процедура. Приведет ли это к другой крайней точке или просто к другой оптимальной таблице, соответствующей той же крайней точке, зависит от задачи.

В приложении П2 приводится описание обучающих компьютерных программ, реализующих решение задач симплекс-методом. Программа позволяет либо сразу получить оптимальное решение задачи, либо провести решение задачи по этапам.

Рассмотрим геометрическую интерпретацию симплекс-метода.

В условия одной из первых задач линейного программирования, для которых Данциг разработал вычислительный метод, входили ограничения вида

$$\sum_{j=1}^n x_j = 1, \quad x_j \geq 0, \quad j = 1, 2, \dots, n.$$

Эти ограничения в n -мерном пространстве определяют *симплекс*. Симплекс трехмерного пространства изображен на рис. 2.1. Рассмотрим неравенство $x_1 + x_2 \leq b_1$ при условиях $x_1 \geq 0$, $x_2 \geq 0$. Область решения этого неравенства показана на рис. 2.2. Данное неравенство можно преобразовать в уравнение введением слабой переменной x_3 . Тогда получим систему

$$x_1 + x_2 + x_3 = b_1, \quad x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0.$$

Областью решений этой системы является ΔABC , показанный на рис. 2.1, если принять, что $A = B = C = b_1$. Каждой точке треугольной области ABC соответствует точка заштрихованной области на рис. 2.2. Соответствие можно устанавливать, проецируя треугольную область ABC на плоскость x_1, x_2 . Если придать слабой переменной x_3 постоянное значение c , то x_1 и x_2 должны удовлетворять уравнению $x_1 + x_2 = b_1 - c$, которое является уравнением прямой, параллельной $x_1 + x_2 = b_1$. Если слабая переменная равна нулю, то $x_1 + x_2 = b_1$. Таким образом, значение слабой переменной может служить мерой близости точки из треугольной области ABC к границе $x_1 + x_2 = b_1$ полуплоскости, определяемой исходным неравенством.

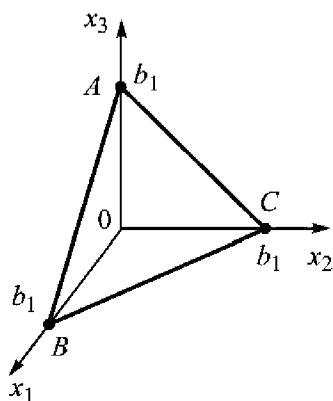


Рис. 2.1. Симплекс трехмерного пространства

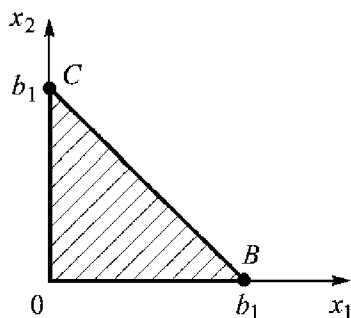


Рис. 2.2. Область решения неравенства

В общем случае при использовании симплекс-метода процедуру поиска решения начинают с рассмотрения допустимой вершины (крайней точки), а затем переходят в соседнюю вершину так, чтобы значение целевой функции «улучшилось».

Если $S \subset \mathbb{R}^n$, то точка \bar{x} называется *крайней точкой* (вершиной) множества S тогда и только тогда, когда не существует точек $x_1, x_2 \in S$, $x_1 \neq x_2$, таких, что $\bar{x} = \lambda x_1 + (1 - \lambda)x_2$ для $\lambda \in (0, 1)$. Все крайние точки принадлежат границе множества.

В пространстве векторов свободных переменных возрастание от нуля значения одной из свободных переменных, при котором остальные свободные переменные остаются равными нулю, равносильно движению из начала системы координат, образованной свободными переменными, по соответствующей координатной оси. При этом, поскольку $n - 1$ свободных переменных равны нулю,

$n-1$ ограничений задачи выполняются как равенства. Другими словами, все соседние с началом координат вершины (которые соответствуют текущему решению) связаны с началом координат $n-1$ ребрами выпуклого многогранника. Возрастание от нуля значения некоторой свободной переменной может привести к тому, что эта переменная станет базисной.

Для того чтобы получить в качестве решения координаты вершины, необходимо заменить одну из базисных переменных на свободную, т. е. произвести перемещение вдоль соответствующей координатной оси до тех пор, пока не будет достигнута другая вершина. Если двигаться дальше, то будет нарушено условие неотрицательности переменных. Таким образом, в симплекс-методе происходят движение в локальной координатной системе, началом которой является точка, соответствующая текущему решению, и перемещение вдоль ребра к той соседней вершине, в которой значение целевой функции «улучшается». После перехода в новую вершину рассматривают новую систему координат с началом в этой вершине. Если движение осуществляют согласно критерию (выбирают минимальный отрицательный коэффициент при неизвестных в целевой функции), то это соответствует спуску по самому крутому ребру из всех пересекающихся в начале координат. Величину изменения целевой функции за одну итерацию определяют как углом наклона (крутизной) ребра, так и длиной ребра. Более точно, она равна минимальному значению по i величины $|c_i b_i / a_{ij}|$ для данного j , где a_{ij} — i -й элемент вектор-столбца a_j для j -й свободной переменной.

Замечание. Процедура решения задач симплекс-методом представляет собой последовательное построение системы таблиц. Предполагается, что в первую таблицу внесено допустимое базисное решение. В задачах, описывающих реальные системы, допустимое базисное решение подобрать сложно. Для этого решают вспомогательную задачу ЛП, которая позволяет не только найти допустимое базисное решение, но и установить, совместна ли система ограничений исходной задачи (*метод искусственного базиса*).

Пусть система ограничений исходной задачи записана в следующем виде:

$$b_i - \sum_{j=1}^n a_{ij} x_j = 0, \quad i = 1, 2, \dots, m,$$

где $b_i \geq 0$, $i = 1, 2, \dots, m$.

Такую запись можно получить, умножив при необходимости уравнения на -1 . Введем новые переменные

$$\xi_i = b_i - \sum_{j=1}^n a_{ij} x_j, \quad (2.2)$$

переменные ξ_i являются базисными, и рассмотрим новую целевую функцию

$$f(\xi) = \sum_{i=1}^m \xi_i \rightarrow \min. \quad (2.3)$$

Допустимое решение для задачи (2.2), (2.3) сразу задано. В процессе решения задачи возможны два случая:

1) $\min f(\xi) = 0$, $\xi_i = 0$, $i = 1, 2, \dots, m$ (все ξ_i стали свободными переменными), — полученное решение x_j , $j = 1, 2, \dots, n$, является допустимым решением исходной задачи ЛП;

2) $\min f(\xi) > 0$ — система ограничений исходной задачи несовместна.

В первом случае можно отметить две особенности.

1. Целевая функция $f(\xi)$ достигла своего минимума, равного нулю, а некоторые из переменных ξ_i находятся среди базисных, хотя и равны нулю. При этом можно не обращать внимание на знаки в строке для целевой функции (можно любую свободную переменную выводить в базисные), так как значение целевой функции не изменится, однако должны выполняться условия, обеспечивающие допустимость нового базисного решения (необходимо рассмотреть минимальное положительное отношение).

2. Возможна ситуация, когда даже после выполнения условий предыдущего пункта в строке для базисной переменной ξ_i нет положительных элементов (нельзя получить положительное отношение). Это означает, что переменные, входящие в уравнение для ξ_i с ненулевыми коэффициентами, в данной задаче должны быть равны нулю. В процессе дальнейшего решения их необходимо исключить из рассмотрения.

Наряду с решением вспомогательной задачи линейного программирования (2.2), (2.3) преобразуется и целевая функция исходной задачи, которая приписывается к задаче (2.2), (2.3) в виде дополнительной строки (см. далее табл. 2.1).

Аналитически процедура симплекс-метода представляется следующим образом.

Пусть задана целевая функция

$$z = cx$$

при условиях

$$Ax = b, \quad x > 0$$

или

$$\begin{pmatrix} 1 & -c \\ 0 & A \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}.$$

Если выбраны базисные переменные x_b и свободные переменные равны нулю, то получим

$$z = c_b x_b$$

при условии

$$Bx_b = b,$$

где c_b — коэффициенты целевой функции (элементы вектора c), относящиеся к базисным переменным; B — квадратная матрица, составленная из столбцов матрицы A для базисных переменных.

Текущее решение удовлетворяет следующему выражению:

$$\begin{pmatrix} z \\ x_b \end{pmatrix} = \begin{pmatrix} 1 & -c_b \\ 0 & B \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ b \end{pmatrix} = \begin{pmatrix} 1 & -c_b B^{-1} \\ 0 & B^{-1} \end{pmatrix} \begin{pmatrix} 0 \\ b \end{pmatrix} = \begin{pmatrix} c_b B^{-1} b \\ B^{-1} b \end{pmatrix}.$$

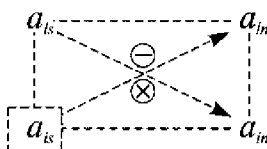
Тогда симплекс-таблица исходной задачи имеет вид

$$\begin{pmatrix} 1 & c_b B^{-1} \\ 0 & B^{-1} \end{pmatrix} \begin{pmatrix} 1 & -c \\ 0 & A \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} = \begin{pmatrix} 1 & c_b B^{-1} \\ 0 & B^{-1} \end{pmatrix} \begin{pmatrix} 0 \\ b \end{pmatrix}$$

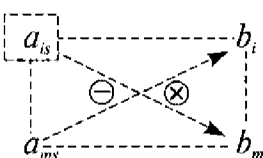
и состоит из следующих столбцов: $(z, x_b)^T$ — базис, $(c_b B^{-1} P_i - c_i, B^{-1} P_i)^T$ — значения коэффициентов целевой функции z и коэффициентов при переменных x , где P_i — i -й столбец матрицы A , c_i — i -й коэффициент целевой функции z , $(c_b B^{-1} b, B^{-1} b)^T$ — значения целевой функции и базисных переменных, т. е. решение.

Заметим, что вычислять надо только матрицу B^{-1} .

ника. Например, новое значение элемента a_{ln} вычисляется по формуле (рядом приведена схема расчета)

$$a_{ln}^H = \frac{a_{is}a_{ln} - a_{ls}a_{in}}{a_{is}},$$


а новые значения элементов столбца b_m свободных членов — по формуле

$$b_m^H = \frac{a_{is}b_m - a_{ms}b_i}{a_{is}}.$$


Из правила прямоугольника следует, что если в разрешающей строке (столбце) есть нулевые элементы, то элементы столбцов (строк), пересекающих эти нулевые элементы, остаются без изменения.

В процессе решения задачи линейного программирования симплекс-методом возможно *заикливание*. Поясним его суть. Пусть в процессе решения задачи ЛП на некотором шаге симплекс-метода число наименьших положительных отношений свободных членов к элементам разрешающего столбца оказалось больше одного, т. е. выбор разрешающего элемента неоднозначен. После этого шага все упомянутые свободные члены, за исключением свободного члена разрешающей строки, обратятся в нуль. Такой случай называют *вырождением*: совпадают две вершины (или большее число вершин) выпуклого многогранника D , при этом ребро (или ребра), соединяющие эти вершины, стягиваются в точку.

В алгоритме симплекс-метода каждый шаг означает переход по ребру от данной вершины многогранника D к соседней (расположенной на том же ребре), а при вырождении — совпадении двух соседних вершин — алгоритм может потерять монотонность, т. е. может случиться, что после указанного шага мы остались в той же вершине, только выраженной с помощью другого набора из n уравнений, относящихся к этой вершине. Если продолжать решение симплекс-методом, то не исключено, что после некоторого числа шагов мы вернемся к уже взятой ранее вершине и процесс начнет повторяться. Произойдет заикливание. Если в процессе решения

проводилось запоминание уже испытанных ребер, то для прерывания заикливания достаточно сменить генеральный элемент.

Существуют алгоритмы, в которых автоматически предусмотрены меры против заикливания.

§ 2.4. Задача планирования выпуска продукции пошивочного предприятия

Выпуск продукции пошивочного предприятия может быть определен из решения задачи линейного программирования. Решать ее будем симплекс-методом.

Задача. На предприятии планируется выпуск двух видов костюмов: мужских и женских. Для пошива женского костюма требуется 1 м шерсти, 2 м лавсана и 1 человеко-день трудозатрат; для пошива мужского костюма — 3,5 м шерсти, 0,5 м лавсана и также 1 человеко-день трудозатрат. Всего для пошива этих костюмов имеется 350 м шерсти, 240 м лавсана и 150 человеко-дней трудозатрат. По плану количество костюмов не должно быть менее 110 и необходимо обеспечить прибыль не менее 1400 усл. ед. Требуется определить оптимальное число костюмов каждого вида, обеспечивающее максимальную прибыль, если прибыль от реализации женского костюма составляет 10 усл. ед., а от мужского — 20 усл. ед.

Решение. Пусть x_1 — число женских костюмов, а x_2 — мужских. Прибыль от женских костюмов составляет $10x_1$ усл. ед., а от мужских — $20x_2$ усл. ед., т. е. необходимо максимизировать целевую функцию

$$f(x) = 10x_1 + 20x_2.$$

Расход шерсти составляет $x_1 + 3,5x_2$ м, лавсана $2x_1 + 0,5x_2$ м, трудовых ресурсов — $x_1 + x_2$ человеко-дней. Поэтому условия-ограничения задачи имеют следующий вид:

$$x_1 + 3,5x_2 \leq 350,$$

$$2x_1 + 0,5x_2 \leq 240,$$

$$x_1 + x_2 \leq 150,$$

$$x_1 + x_2 \geq 110,$$

$$10x_1 + 20x_2 \geq 1400,$$

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

Первые три неравенства описывают ограничения по ресурсам, четвертое и пятое — соответственно плановое задание по общему числу костюмов и ограничение по прибыли.

Для решения задачи симплекс-методом сведем систему ограничений к равенствам путем введения неотрицательных слабых переменных x_3, x_4, x_5, x_6, x_7 (первое и второе ограничения умножим на 2, а пятое разделим на 10):

$$\begin{aligned} 2x_1 + 7x_2 + x_3 &= 700, \\ 4x_1 + x_2 + x_4 &= 480, \\ x_1 + x_2 + x_5 &= 150, \\ -x_1 - x_2 + x_6 &= -110, \\ -x_1 - 2x_2 + x_7 &= -140, \\ x_j &\geq 0, \quad j = 1, \dots, 7. \end{aligned} \tag{2.4}$$

Первым этапом симплекс-метода является отыскание опорного решения — допустимого базисного решения, с которого начинается поиск оптимального решения. Чтобы решение было опорным, базисные переменные должны быть неотрицательными, т. е. элементы $b_i, i = 1, \dots, 5$, столбца свободных членов должны быть неотрицательными. В задачах небольшой размерности опорное решение легко определить. В данном случае в качестве базисных переменных можно взять x_3, x_4, x_5, x_6, x_7 , но такое базисное решение не является допустимым (опорным), так как b_4 и b_5 отрицательны. Для поиска опорного решения необходимо сформировать дополнительную фиктивную целевую функцию $\varphi(x)$, элементы которой равны сумме элементов строк, отражающих те ограничения, где $b_i < 0$. В симплекс-таблице для $\varphi(x)$ вводится и заполняется дополнительная строка, получаемая суммированием соответствующих элементов строк с отрицательными значениями b_i (в данном случае 4-я и 5-я строки). Теперь с помощью симплекс-метода находится максимум фиктивной целевой функции $\varphi(x)$. Если $\max \varphi(x) = 0$ и при этом все коэффициенты в строке для $\varphi(x)$ будут нулевыми, то базисное решение, соответствующее этой таблице, будет опорным. Тогда, исключая строку для $\varphi(x)$, переходим к отысканию оптимального решения исходной задачи. Если $\max \varphi(x) \neq 0$, то система ограничений задачи противоречива. Может иметь место случай, когда функция $\varphi(x)$ достигла своего максимума, равного нулю, а среди элементов строки $\varphi(x)$ суще-

столбца $-x_2$: $\frac{700}{7} = 100$, $\frac{480}{1} = 480$, $\frac{150}{1} = 150$, $\frac{-110}{-1} = 110$, $\frac{-140}{-2} = 70$. Минимальное значение, равное 70, указывает, что пе-

ременную x_7 надо перевести в свободные переменные, а *генеральным* (*разрешающим*) элементом является -2 (обведен). Для получения следующей симплекс-таблицы применим метод полного исключения Жордана (табл. 2.2).

Таблица 2.2

Первая итерация

Базисные переменные	b_i	$-x_1$	$-x_2$	$-x_3$	$-x_4$	$-x_5$	$-x_6$	$-x_7$	Контроль
x_3	210	$-3/2$	0	1	0	0	0	$7/2$	213
x_4	410	$7/2$	0	0	1	0	0	$1/2$	415
x_5	80	$1/2$	0	0	0	1	0	$1/2$	82
x_6	-40	$-1/2$	0	0	0	0	1	$-1/2$	-40
x_7	70	$1/2$	1	0	0	0	0	$-1/2$	71
$f(x)$	1400	0	0	0	0	0	0	-10	1390
$\varphi(x)$	-40	$-1/2$	0	0	0	0	1	$-1/2$	-40

Разрешающую строку x_7 делим на -2 и заносим в табл. 2.2; столбец $-x_2$ заполняем нулями. Элементы столбцов $-x_3$, $-x_4$, $-x_5$, $-x_6$ переносим без изменения, так как они пересекают нулевые элементы разрешающей строки. Остальные элементы таблицы пересчитываем по правилу прямоугольника. Проверяем, совпадает ли сумма чисел в строке с числом в контрольном (последнем) столбце. Если совпадения нет, произошла ошибка в расчете. Вместе с функцией $\varphi(x)$ пересчитывают и целевую функцию $f(x)$.

Продолжаем максимизировать $\varphi(x)$. В базисные переменные можно перевести x_1 или x_7 , так как они имеют равные отрицательные коэффициенты $-1/2$ в строке $\varphi(x)$. Переводим x_1 в базисные переменные. По минимуму положительных отношений b_i к элементам столбца $-x_1$ выбираем элемент x_6 , который надо перевести в свободные переменные; генеральным элементом является $-1/2$, значит, x_6 — разрешающая строка и $-x_1$ — разрешающий столбец. Запишем результаты в виде табл. 2.3.

Таблица 2.3

Вторая итерация

Базисные переменные	b_i	$-x_1$	$-x_2$	$-x_3$	$-x_4$	$-x_5$	$-x_6$	$-x_7$	Контроль
x_3	330	0	0	1	0	0	-3	5	333
x_4	130	0	0	0	1	0	7	-3	135
x_5	40	0	0	0	0	1	1	0	42
x_1	80	1	0	0	0	0	-2	1	80
x_2	30	0	1	0	0	0	1	-1	31
$f(x)$	1400	0	0	0	0	0	0	-10	1390
$\varphi(x)$	0	0	0	0	0	0	0	0	0

Из табл. 2.3 видно, что достигнут максимум фиктивной целевой функции ($\varphi(x) = 0$) и все коэффициенты в строке $\varphi(x)$ равны нулю, т. е. получено *опорное* решение $x_6 = x_7 = 0$, $x_1 = 80$, $x_2 = 30$, $x_3 = 330$, $x_4 = 130$, $x_5 = 40$. Это решение не является оптимальным, так как в строке $f(x)$ имеется отрицательный коэффициент.

Продолжаем улучшать решение симплекс-методом. Строку $\varphi(x)$ исключаем. Генеральным будет элемент, равный 5; x_7 переводим в базис (вместо x_3). Полученные результаты запишем в виде табл. 2.4, затем в виде табл. 2.5. В табл. 2.4 обведен генеральный элемент.

Таблица 2.4

Третья итерация

Базисные переменные	b_i	$-x_1$	$-x_2$	$-x_3$	$-x_4$	$-x_5$	$-x_6$	$-x_7$	Контроль
x_7	66	0	0	1/5	0	0	-3/5	1	333/5
x_4	328	0	0	3/5	1	0	26/5	0	1674/5
x_5	40	0	0	0	0	1	1	0	42
x_1	14	1	0	-1/5	0	0	-7/5	0	67/5
x_2	96	0	1	1/5	0	0	-2/5	0	488/5
$f(x)$	2060	0	0	2	0	0	-6	0	2056

Таблица 2.5

Оптимальное решение

Базисные переменные	b_i	$-x_1$	$-x_2$	$-x_3$	$-x_4$	$-x_5$	$-x_6$	$-x_7$	Контроль
x_7	90	0	0	1/5	0	3/5	0	1	459/5
x_4	120	0	0	3/5	1	-26/5	0	0	582/5
x_6	40	0	0	0	0	1	1	0	42
x_1	70	1	0	-1/5	0	7/5	0	0	361/5
x_2	80	0	1	1/5	0	-2/5	0	0	404/5
$f(x)$	2300	0	0	2	0	6	0	0	2308

В табл. 2.5 все коэффициенты строки $f(x)$ неотрицательны, значит, максимум функции $f(x)$ достигнут и получено соответствующее ему решение: $x_1 = 70$, $x_2 = 80$, $f_{\max}(x) = 2300$. Таким образом, максимальная прибыль составит 2300 усл. ед. при пошиве 70 женских и 80 мужских костюмов. Слабые переменные оказались равными $x_3 = 0$, $x_4 = 120$, $x_5 = 0$, $x_6 = 40$, $x_7 = 90$. Значения x_3 , x_4 , x_5 показывают остатки ресурсов: шерсть и трудовые ресурсы израсходованы полностью, лавсана осталось 120 м; значения x_6 и x_7 показывают, на сколько перевыполнены плановые задания по числу костюмов и по прибыли (учитываем, что $x_7 = 90 \cdot 10 = 900$ усл. ед.).

Замечание. Обратим внимание на тот факт, что в рассмотренной задаче оптимальное решение мы устанавливали по наличию неотрицательных коэффициентов в строке максимизируемой целевой функции, а в задаче § 2.2 — по наличию неотрицательных коэффициентов в минимизируемой целевой функции. Здесь никакого противоречия нет: в нашей задаче переменные указаны со знаком минус, поэтому и рассматривались неотрицательные коэффициенты.

Рассмотрим графическое решение этой задачи. Поскольку задача двумерная, то решим ее графически. Система неравенств-ограничений определяет многоугольник допустимых решений (рис. 2.3).

Сначала определим полуплоскости, задаваемые неравенствами-ограничениями задачи. Для этого построим прямые, заменив в

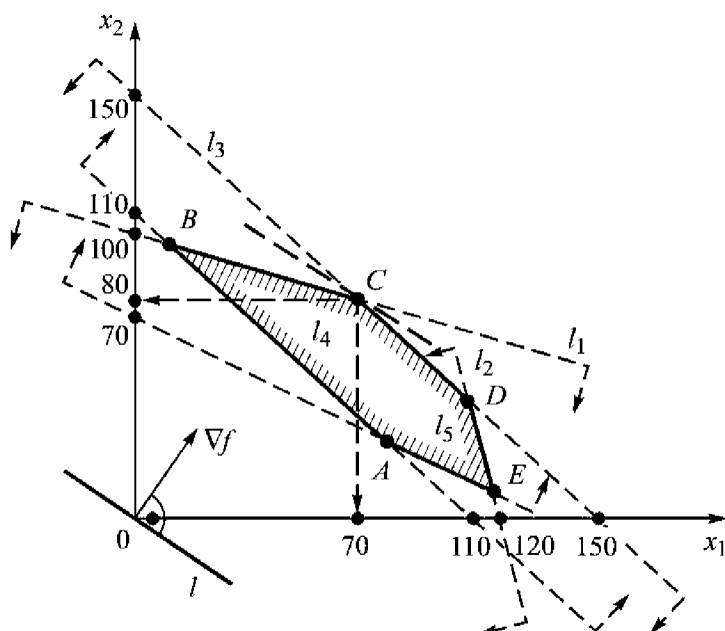


Рис. 2.3. Графическое решение задачи планирования выпуска продукции пошивочного предприятия

ограничениях знаки неравенств знаками равенств. Чтобы выяснить, какую часть плоскости описывает неравенство, подставим в него пробную точку, например $(0, 0)$, и установим, удовлетворяет ли она неравенству. Если неравенство выполнено, то искомая полуплоскость включает точку $(0, 0)$. В противном случае выбирают другую половину плоскости.

Для первого неравенства прямую l_1 ($x_1 + 3,5x_2 = 350$) строим по точкам $x_1 = 0$, $x_2 = 350/3,5 = 100$ и $x_2 = 0$, $x_1 = 350$. Пробная точка $(0, 0)$ удовлетворяет неравенству $0 < 350$, т. е. точка $(0, 0)$ входит в искомую полуплоскость (она отмечена стрелками от прямой l_1). Прямую l_2 ($2x_1 + 0,5x_2 = 240$) строим аналогично по двум точкам: $x_1 = 0$, $x_2 = 240/0,5 = 480$ и $x_2 = 0$, $x_1 = 240/2 = 120$. Точка $(0, 0)$ принадлежит искомой полуплоскости. Рассмотрим последнее неравенство $10x_1 + 20x_2 \geq 1400$; ему соответствует прямая l_5 ($x_1 + 2x_2 = 140$) также построенная по двум точкам: $x_1 = 0$, $x_2 = 140/2 = 70$ и $x_2 = 0$, $x_1 = 140$. Точка $(0, 0)$ не удовлетворяет неравенству $0 \geq 1400$, т. е. необходимо рассматривать полуплоскость, не содержащую точку $(0, 0)$. Пересечение пяти полуплоско-

стей дает выпуклый пятиугольник $ABCDE$. Для нахождения максимума функции $f(x)$ надо построить линию уровня. Пусть $f(x)=0$, тогда уравнение линии уровня l будет прямая $10x_1 + 20x_2 = 0$, проходящая через начало координат параллельно прямой l_5 . Градиент целевой функции $\nabla f = \{10; 20\}$ показывает направление ее возрастания. Прямую l перемещаем параллельно самой себе в направлении ∇f до тех пор, пока она не выйдет из области D . Получаем точку C — точку пересечения прямых l_1 и l_3 :

$$x_1 + 3,5x_2 = 350,$$

$$x_1 + x_2 = 150.$$

Решая полученную систему уравнений, находим оптимальное решение — координаты точки C ($x_1 = 70$, $x_2 = 80$) и вычисляем максимальное значение целевой функции

$$f_{\max}(x) = 10 \cdot 70 + 20 \cdot 80 = 2300 \text{ усл. ед.}$$

Замечание. Допустим, что в рассматриваемой задаче требовалось найти минимум целевой функции

$$f(x) = 10x_1 + 20x_2.$$

В этом случае линия уровня «вошла» бы в область по линии l_5 , т. е. все точки отрезка AE являлись бы оптимальным решением (бесконечное множество решений).

§ 2.5. Двойственность в задачах линейного программирования

В § 1.6 было показано, что иногда проще решать двойственную задачу математического программирования, чем прямую. Согласно теории двойственности (см. § 1.6) для каждой задачи ЛП можно построить другую задачу линейного программирования, называемую *двойственной*. Переход к решению двойственной задачи в ряде случаев имеет преимущества при построении алгоритмов решения задач ЛП.

Запишем обе задачи.

Прямая задача:

минимизировать функцию

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

при условиях

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

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

$$x_j \geq 0, \quad j = 1, 2, \dots, l,$$

$$x_j \geq 0, \quad j = l+1, \dots, n.$$

Двойственная задача:

максимизировать функцию

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

при условиях

$$y_i \geq 0, \quad i = 1, 2, \dots, k,$$

$$y_i \geq 0, \quad i = k+1, \dots, m,$$

$$\sum_{i=1}^m a_{ji} y_i \leq c_j, \quad j = 1, 2, \dots, l,$$

$$\sum_{i=1}^m a_{ji} y_i = c_j, \quad j = l+1, \dots, n.$$

Симметричность обеих задач очевидна. Неравенству одной из них соответствует неотрицательная переменная другой, а равенству — свободная переменная другой. Задача, двойственная к двойственной задаче, есть исходная (прямая) задача. Таким образом, любую из этой пары задач можно считать прямой.

Для стандартного и канонического видов задачи ЛП двойственные задачи можно записать следующим образом.

Стандартный вид

Прямая задача:

минимизировать функцию

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

при условиях

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

$$x_j \geq 0, \quad j = 1, 2, \dots, n.$$

Двойственная задача:

максимизировать функцию

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

при условиях

$$\sum_{i=1}^m a_{ji} y_i \leq c_j, \quad j = 1, 2, \dots, n,$$

$$y_i \geq 0, \quad i = 1, 2, \dots, m.$$

Канонический вид

Прямая задача:

минимизировать функцию

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

Двойственная задача:

максимизировать функцию

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

2) одна из задач не имеет ни одного допустимого решения, а другая имеет по крайней мере одно допустимое решение, но не имеет оптимального решения (целевая функция на множестве допустимых решений не ограничена);

3) ни одна пара задач не имеет допустимых решений.

Между решениями двойственных друг другу задач ЛП существуют и другие соотношения, которые устанавливаются теоремами о *дополняющей нежесткости*: для того чтобы допустимые решения x и y прямой и двойственной задач были оптимальными, необходимо и достаточно, чтобы в точке экстремума выполнялись следующие соотношения:

$$y(Ax - b) = 0, \quad (2.5)$$

$$(c - yA)x = 0. \quad (2.6)$$

Условие (2.5) равносильно условиям:

а) если $y_i > 0$, то $\sum_{j=1}^n a_{ij}x_j = b_i$, $i = 1, 2, \dots, m$;

б) если $\sum_{j=1}^n a_{ij}x_j > b_i$, то $y_i = 0$, $i = 1, 2, \dots, m$.

Условие (2.6) равносильно условиям:

в) если $c_j > \sum_{i=1}^m a_{ij}y_i$, то $x_j = 0$, $j = 1, 2, \dots, n$;

г) если $x_j > 0$, то $c_j = \sum_{i=1}^m a_{ij}y_i$, $j = 1, 2, \dots, n$.

Эти условия представляют собой условия дополняющей нежесткости в слабой форме. Выпишем условия дополняющей нежесткости в сильной форме:

д) если $y_i = 0$, то $\sum_{j=1}^n a_{ij}x_j - b_i > 0$, $i = 1, 2, \dots, m$;

е) если $\sum_{j=1}^n a_{ij}x_j - b_i = 0$, то $y_i > 0$, $i = 1, 2, \dots, m$.

Может случиться, что условия $y_i = 0$ и $\sum_{j=1}^n a_{ij}x_j = b_i$ выполняются одновременно. Однако всегда существует по крайней мере одна пара оптимальных решений, для которых условия $y_i = 0$ и

$\sum_{j=1}^n a_{ij}x_j = b_i$ не могут выполняться одновременно.

Если известно оптимальное решение одной из задач, то с помощью условий дополняющей нежесткости можно найти оптимальное решение другой задачи. Для этого нужно записать условия дополняющей нежесткости для прямой и двойственной задачи. Пусть известно оптимальное решение прямой задачи. Тогда для ненулевых значений этого решения получим из условия двойственности систему уравнений для оптимальных решений двойственной задачи. По второму условию дополняющей нежесткости, определив, когда $\sum_{j=1}^n a_{ij}x_j - b_i \neq 0$, найдем двойственные переменные, равные нулю. С учетом этих ограничений из системы уравнений для двойственных переменных определим их оптимальные значения.

Отметим, что в полной оптимальной симплекс-таблице любой задачи линейного программирования в строке целевой функции находится оптимальное решение двойственной задачи.

Рассмотрим геометрическую интерпретацию теории двойственности в задачах ЛП. Выберем задачу ЛП стандартного вида: минимизировать функцию

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

при условиях

$$\sum_{j=1}^n a_{ij}x_j \geq b_i, \quad i=1, 2, \dots, m, \quad x_j \geq 0, \quad j=1, 2, \dots, n.$$

В § 1.3–1.5 было показано, что обычным условием существования безусловного экстремума функции во внутренней точке является равенство нулю градиента функции в этой точке. Если при этом должны выполняться некоторые ограничения, налагаемые на переменные, в виде равенств, то условием существования экстремума в допустимой точке является требование, чтобы в этой точке градиент функции и нормали к поверхностям, заданным соответствующими ограничениями, были «сонаправлены». Более точно, градиент функции в этой точке должен быть неотрицательной линейной комбинацией нормалей к поверхностям-ограничениям. В задаче ЛП каждое неравенство определяет допустимую область — полупространство. Для того чтобы допустимая точка x была оптимальной, необходимо, чтобы градиент целевой функции в точке x выражался в виде неотрицательной линейной комбинации направляющих век-

торов тех и только тех ограничений, которые в точке x обращаются в равенства, т. е. градиент целевой функции (вектор c) есть неотрицательная линейная комбинация нормалей (векторов) a_i^* для ограничений, обращающихся в равенство

$$c = \sum_{i=1}^{m+n} y_i a_i^*, \quad y_i > 0 \Rightarrow (a_i^*, x) = b_i^*,$$

где y_i — соответствующие коэффициенты линейной комбинации векторов a_i^* .

Из условий дополняющей нежесткости в слабой форме следует:

а) если $y_i > 0$, то $(a_i, x) - b_i = 0$, $i = 1, 2, \dots, m+n$;

б) если $(a_i, x) - b_i > 0$, то $y_i = 0$, $i = 1, 2, \dots, m+n$.

Из условий дополняющей нежесткости в сильной форме следует:

а) если $y_i = 0$, то $(a_i, x) - b_i > 0$, $i = 1, 2, \dots, m+n$;

б) если $(a_i, x) - b_i = 0$, то $y_i > 0$, $i = 1, 2, \dots, m+n$.

На рис. 2.4 изображены три гиперплоскости $(a_i, x) - b_i = 0$, $i = 1, 2, 3$, и нормали к ним a_1 , a_2 , a_3 . Если вектор c такой, как показано на рис. 2.4, то он может быть выражен в виде неотрицательной линейной комбинации векторов a_1 и a_2 ; вершина, обозначенная кружком, соответствует оптимальному решению. Здесь выполнены условия дополняющей нежесткости как в слабой, так и в сильной форме: $y_3 > 0 \Leftrightarrow (a_3, x) - b_3 > 0$, $y_1 > 0 \Leftrightarrow (a_1, x) - b_1 = 0$, $y_2 > 0 \Leftrightarrow (a_2, x) - b_2 = 0$. Если вектор c таков, как показано на рис. 2.5, т. е. c — нормаль к одной из гиперплоскостей, $(a_1, x) - b_1 = 0$, то оптимальная вершина, обозначенная кружком, не удовлетворяет сильной форме условия дополняющей нежесткости, поскольку $y_2 = 0$ и $(a_2, x) - b_2 = 0$. Однако точка, помеченная крестом,

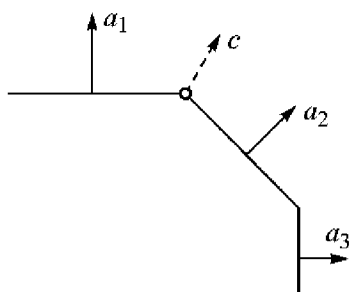


Рис. 2.4. Пример выполнения условия дополняющей нежесткости

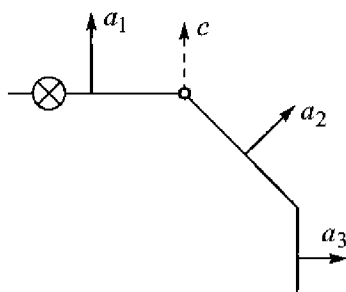


Рис. 2.5. Пример нарушения сильной формы условия дополняющей нежесткости

стом на рис. 2.5 и являющаяся оптимальным решением, удовлетворяет и слабой, и сильной формам дополняющей нежесткости: $y_1 > 0 \Leftrightarrow (a_1, x) - b_1 = 0$, $y_2 = 0 \Leftrightarrow (a_2, x) - b_2 > 0$, $y_3 = 0 \Leftrightarrow (a_3, x) - b_3 > 0$.

Для решения задач ЛП разработан так называемый *двойственный симплекс-метод*. Процедуру начинают с нахождения двойственно допустимого решения, когда одновременно $b_i \geq 0$ и $c_j \geq 0$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, и сохраняют условие двойственной допустимости на протяжении всех шагов. Этот метод реализуется посредством таких же таблиц, как и в прямом симплекс-методе.

Двойственный симплекс-метод позволяет решать задачи ЛП и в случае, когда, после формирования единичной матрицы из коэффициентов для m базисных переменных, в ней содержатся некоторые элементы с отрицательными значениями в столбце свободных членов b_i , т. е. мы не находимся в области допустимых решений. Получим так называемый *псевдоплан*, если $\Delta_j \leq 0$:

$$\Delta_j = c_j - z_j = c_j - \sum_{i=1}^m c_i a_{ij}, \quad j = 1, 2, \dots, n,$$

где c_i , $i = 1, 2, \dots, m$, — коэффициенты целевой функции исходной задачи, стоящие в базисных столбцах; a_{ij} — соответствующие коэффициенты условий-ограничений в симплекс-таблице.

Если в псевдоплане есть хотя бы одно отрицательное число $b_i < 0$, такое, что для него все соответствующие коэффициенты $a_{ij} \geq 0$, $j = 1, \dots, n$, то исходная задача не имеет решения. Если в псевдоплане имеются числа $b_i < 0$, такие, что для некоторых из них существуют числа $a_{ij} < 0$, то можно перейти к новому псевдоплану (при котором значение целевой функции задачи на максимум не уменьшится). Когда все b_i будут неотрицательными, а все $\Delta_j \leq 0$, то будет получено оптимальное решение. Но здесь сначала определяется, какая переменная должна быть выведена из базиса, а затем — какая должна быть введена в базис.

Процедура двойственного симплекс-метода состоит в следующем:

- 1) находим псевдоплан задачи;
- 2) проверяем оптимальность этого псевдоплана:
 - псевдоплан оптимален;
 - задача неразрешима;
 - переход к новому псевдоплану;
- 3) выбираем разрешающую строку с помощью определения наибольшего по модулю отрицательного элемента b_i в столбце

свободных членов и разрешающий столбец с помощью нахождения наименьшего по модулю отношения

$$\min(\Delta_j/a_{ij}), \quad a_{ij} < 0, \quad \Delta_j < 0, \quad j=1, 2, \dots, n;$$

4) переводим i -ю базисную переменную в свободные переменные, при этом переменная j -го столбца становится базисной переменной;

5) переходим к п. 2).

Всегда существует возможность выбора: решать прямую или двойственную задачу, использовать прямой или двойственный симплекс-метод. Выбирают ту модификацию задачи, которую проще решать. Например, если исходная задача содержит переменные, на которые не наложено условие неотрицательности, то удобнее решать двойственную задачу. Прежде чем записать двойственную задачу, необходимо в исходной прямой задаче освободиться от ограничений в виде равенств, поскольку они будут порождать в двойственной задаче переменные, принимающие значения на всей действительной оси \mathbb{R}^1 .

В симплекс-таблице оптимального решения прямой задачи ЛП записано и решение двойственной к ней задачи. Чтобы это понять, надо элементы строки, в которой выписаны коэффициенты целевой функции, представить в виде $c_j - z_j$, $z_j = c_b^T y_j$, $j=1, 2, \dots, n$, где c_b — вектор, состоящий из коэффициентов целевой функции *исходной задачи* («т» — знак транспонирования), стоящих в базисных клетках оптимального решения, y_j — элементы j -го столбца симплекс-таблицы оптимального решения, и добавить в симплекс-таблицу дополнительную строку z_j . На пересечении строки z_j и столбцов базисных переменных исходной задачи (обычно это последние m столбцов) находится оптимальное решение двойственной задачи. Таким образом, оптимальное решение y^* двойственной задачи — это m последних элементов строки z_j оптимальной симплекс-таблицы прямой задачи; а оптимальным решением x^* прямой задачи являются n последних элементов строки z_j оптимальной таблицы двойственной задачи.

Известны, кроме того, методы одновременного решения прямой и двойственной задач, например *метод последовательного сокращения невязок*, где величина $\sum_{j=1}^n a_{ij}x_j - b_i$, $i=1, 2, \dots, m$, является невязкой при фиксированных значениях x_j .

§ 2.6. Задача оптимальной организации поставки грузов от поставщиков к потребителям (транспортная задача)

Симплекс-методом возможно решить любую задачу ЛП. Однако существует много методов решения задач линейного программирования, в которых учитываются конкретные особенности решаемой задачи, а потому более эффективные. Решим таким методом *транспортную задачу*.

Задача. Перевозится однородный груз из трех пунктов A_1 , A_2 , A_3 к четырем местам назначения B_1 , B_2 , B_3 , B_4 . Из пункта A_1 может быть направлено 50 т, из A_2 — 40 т, из A_3 — 20 т. В пункты назначения должен поступить груз: в пункт B_1 — 30 т, в B_2 — 25 т, в B_3 — 35 т, в B_4 — 20 т. Расстояния c_{ij} , $i=1, 2, 3$, $j=1, \dots, 4$, от i -го поставщика к j -му потребителю приведены в углах клеток табл. 2.6.

Таблица 2.6

Исходные данные

Пункт	B_1		B_2		B_3		B_4		Запасы
A_1	³ x_{11}		² x_{12}		⁴ x_{13}		¹ x_{14}		$a_1 = 50$
A_2	² x_{21}		³ x_{22}		¹ x_{23}		⁵ x_{24}		$a_2 = 40$
A_3	³ x_{31}		² x_{32}		⁴ x_{33}		⁴ x_{34}		$a_3 = 20$
Потребности	$b_1 = 30$		$b_2 = 25$		$b_3 = 35$		$b_4 = 20$		110

Необходимо составить план перевозок, обеспечивающий наименьший общий пробег транспорта в тонно-километрах при условии, что все запасы будут вывезены, а потребитель получит точно необходимое количество груза.

Мы сформулировали сбалансированную транспортную задачу при условии, что количество груза, имеющееся у поставщиков, равно количеству груза, необходимому потребителю. Несбалансированная транспортная задача сводится к сбалансированной путем введения фиктивного поставщика, если потребности превышают предложения, или фиктивного потребителя в противном случае. Расстояния в фиктивной строке (столбце) указываются равными нулю. Сбалансированная и несбалансированная задачи решаются по одному алгоритму.

Решение. Пусть x_{ij} — количество груза, которое будет доставлено из i -го пункта отправления в j -й пункт назначения, $i = 1, 2, 3$, $j = 1, \dots, 4$. Необходимо минимизировать целевую функцию

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

при условиях, что весь груз от поставщика должен быть вывезен:

$$\sum_{j=1}^4 x_{ij} = a_i, \quad i = 1, 2, 3,$$

и каждый потребитель получит необходимое ему количество груза:

$$\sum_{i=1}^3 x_{ij} = b_j, \quad j = 1, \dots, 4.$$

Особенностью данной задачи является то, что в ограничениях все коэффициенты при неизвестных равны единице. Это облегчает вычисления по симплекс-методу.

Число переменных в данной задаче равно в общем случае mn , где m — число поставщиков, n — число потребителей. Число уравнений в системе ограничений равно $m + n$. Однако нетрудно увидеть, что одно из этих уравнений может быть получено из других. Так, если определены количества груза, имеющиеся у всех отправителей и необходимые всем получателям, кроме одного, то спрос последнего легко установить как разность между общим запасом и общей потребностью остальных получателей, т. е. система ограничений содержит $m + n - 1$ независимых уравнений с mn неизвестными. Число базисных переменных также будет равно $m + n - 1$, остальные переменные являются свободными.

Алгоритм решения транспортной задачи сравним с алгоритмом симплекс-метода.

1. Нахождение допустимого базисного (опорного) решения. Транспортную задачу решают несколькими методами.

А. Метод северо-западного угла. Предположим, что весь груз из первого пункта перевезен к первому потребителю. Если потребности первого потребителя оказались выше возможностей первого по-

ставщика, то перевозим груз от второго поставщика. Если запасы первого поставщика выше потребностей первого потребителя, то остаток запасов первого поставщика передаем второму потребителю и т. д. Мы должны заполнить $m + n - 1$ клетку. Может оказаться, что число заполненных клеток меньше $m + n - 1$ (случай *вырождения*), тогда оставшиеся клетки заполняем нулями — это так называемые *условные поставки*. Процесс получения опорного решения рассматриваемой задачи методом северо-западного угла представлен в табл. 2.7–2.10.

Таблица 2.7

Первая итерация

Пункт	B_1	B_2	B_3	B_4	a_i
A_1	30				50* 20
A_2					40
A_3					20
b_j	30*	25	35	20	110

Таблица 2.8

Вторая итерация

Пункт	B_2	B_3	B_4	a_i
A_1	20			20*
A_2	5			40* 35
A_3			20	20
b_j	25*	35	20	80

Таблица 2.9

Третья итерация

Пункт	B_3	B_4	a_i
A_2	35		35*
A_3		20	20
b_j	35	20	55

Таблица 2.10

Четвертая итерация

Пункт	B_1	B_2	B_3	B_4	a_i
A_1	³ 30	² 20	⁴	¹	50
A_2	²	³ 5	¹ 35	⁵	40
A_3	³	²	⁴	⁴ 20	20
b_j	30	25	35	20	110

Помеченные звездочкой значения в табл. 2.7–2.9 обозначают предыдущие данные, которые изменились и стали равны числу, стоящему после звездочки, если это число не равно нулю.

За четыре итерации мы заполнили в таблице перевозок (см. табл. 2.10) пять клеток вместо шести, однако выполнили условия-ограничения. Необходимо ввести нулевую клетку — условную поставку. Пусть это будет клетка (1, 4): $x_{14} = 0$. Получили опорное решение: $x_{11} = 30$, $x_{12} = 20$, $x_{14} = 0$, $x_{22} = 5$, $x_{23} = 35$, $x_{34} = 20$. Значением целевой функции при таком решении является

$$\begin{aligned}
 f(x) &= 3 \cdot 30 + 2 \cdot 20 + 1 \cdot 0 + 1 \cdot 35 + 3 \cdot 5 + 4 \cdot 20 = \\
 &= 260 \text{ (тонно-километров).}
 \end{aligned}$$

Б. *Метод учета наименьших расстояний (стоимостей) перевозок.* Данный метод аналогичен методу северо-западного угла, только заполняются в первую очередь те клетки, для которых указанные расстояния (стоимости) наименьшие.

2. Проверка оптимальности полученного плана перевозок. Рассмотрим двойственную задачу к поставленной транспортной задаче. В качестве двойственных переменных введем величины $\alpha_1, \alpha_2, \alpha_3$, соответствующие первым трем ограничениям, и $\beta_1, \beta_2, \beta_3, \beta_4$ — остальным ограничениям. Эти переменные называют *потенциалами*, а метод решения транспортной задачи — *методом потенциалов*.

Целевая функция двойственной задачи имеет вид

$$\varphi(\alpha, \beta) = 50\alpha_1 + 40\alpha_2 + 20\alpha_3 + 30\beta_1 + 25\beta_2 + 35\beta_3 + 20\beta_4 \rightarrow \max$$

при условиях (для базисных клеток)

$$\begin{aligned} \alpha_1 + \beta_1 &\leq 3, & \alpha_1 + \beta_2 &\leq 2, & \alpha_1 + \beta_4 &\leq 1, \\ \alpha_2 + \beta_2 &\leq 3, & \alpha_2 + \beta_3 &\leq 1, & \alpha_3 + \beta_4 &\leq 4. \end{aligned} \quad (2.7)$$

Для нахождения оптимального плана перевозки в исходной задаче условия-ограничения двойственной задачи выполнялись бы как равенства, поскольку согласно условиям дополняющей нежесткости если в оптимальном плане исходной задачи значение какой-либо переменной строго больше нуля, то соответствующее ограничение двойственной задачи становится равенством при подстановке в него оптимального плана. Получим систему шести уравнений с семью переменными (2.7). Поскольку количество независимых переменных в данной системе равно $3 + 4 - 1 = 6$, то одна переменная является свободной. Пусть это будет α_1 . Положим $\alpha_1 = 0$, получим $\beta_1 = 3, \beta_2 = 2, \beta_4 = 1, \alpha_2 = 1, \beta_3 = 0, \alpha_3 = 3$. Составим таблицу перевозок для данной итерации (табл. 2.11).

Таблица 2.11

Первая итерация

Потенциал	$\beta_1 = 3$	$\beta_2 = 2$	$\beta_3 = 0$	$\beta_4 = 1$
$\alpha_1 = 0$	³ 30	² 20	⁴	¹ 0
$\alpha_2 = 1$	²	³ 5	¹ 35	⁵
$\alpha_3 = 3$	³	²	⁴	⁴ 20

Полученное решение $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \beta_4$ подставляем в ограничения двойственной задачи, не вошедшие в систему уравнений (2.7), т. е. соответствующие пустым клеткам. Если эти ограничения являются верными неравенствами для найденного решения, то проверяемый допустимый план исходной задачи является оптимальным. В противном случае он оптимальным не является. Для пустых клеток имеем

$$\begin{aligned}\alpha_1 + \beta_3 &\leq 4, & 0 &\leq 4, \\ \alpha_2 + \beta_1 &\leq 2, & 4 &\leq 2, \\ \alpha_2 + \beta_4 &\leq 5, & 2 &\leq 5, \\ \alpha_3 + \beta_1 &\leq 3, & 6 &\leq 3, \\ \alpha_3 + \beta_2 &\leq 2, & 5 &\leq 2, \\ \alpha_3 + \beta_3 &\leq 4, & 4 &\leq 4.\end{aligned}\tag{2.8}$$

Второе, четвертое и пятое неравенства являются неверными, поэтому решение не является оптимальным. Необходимо провести улучшение плана.

3. Составление нового допустимого плана. Наметив свободную переменную, которую надо перевести в базисные переменные, определим базисную переменную, переводимую в свободные переменные. Переменные, соответствующие свободным клеткам (2, 1), (3, 1) и (3, 2) табл. 2.11, в которых неравенства (2.8) нарушаются, могут быть переведены в базисные переменные. Выбираем клетку (3, 2), т. е. переменную x_{32} .

В симплекс-методе для выбора генерального элемента рассматриваются положительные отношения в столбце x_{32} . В матрице перевозок положительные коэффициенты в столбце x_{32} равны +1 и отвечают тем базисным клеткам, которые соответствуют отрицательным вершинам некоторой замкнутой ломаной линии, называемой *циклом пересчета* для x_{32} . Следовательно, генеральным элементом является базисная переменная из числа всех переменных, отвечающих отрицательным вершинам цикла пересчета, значение которой минимально.

Цикл пересчета — это замкнутая ломаная линия, начинающаяся в свободной клетке. Координаты всех остальных вершин ломаной помещены в базисные клетки. Клетки соединены звеньями, расположенными вдоль строк и столбцов матрицы.

В каждую вершину входят только два звена, причем одно из них расположено в строке, другое в столбце. Никакие три вершины, встречающиеся подряд при обходе, не лежат на одной прямой. Если циклом служит самопересекающаяся линия, то точки самопересечения не могут быть ее вершинами. Свободной клетке в цикле присваивают знак «+», другим вершинам — чередующиеся по ходу знаки «-», «+», «-» и т. д. Построим цикл пересчета для свободной клетки (3, 2) (табл. 2.12). В отрицательных вершинах цикла пересчета стоят два числа: 20, 20; минимальным из них является 20. Поскольку число положительных и отрицательных вершин одинаково, баланс не нарушится, если в отрицательных вершинах вычесть число a , а в положительных вершинах прибавить это же число a . Вычитая минимальное из чисел, стоящих в отрицательных вершинах, мы получаем новую свободную переменную x_{34} , а базисной станет x_{32} . Прибавим 20 в положительных вершинах цикла, вычтем 20 в отрицательных вершинах, получим второе допустимое решение (табл. 2.13).

Таблица 2.12

Цикл пересчета для клетки (3, 2)

Потенциал	$\beta_1 = 3$	$\beta_2 = 2$	$\beta_3 = 0$	$\beta_4 = 1$
$\alpha_1 = 0$	30	20 -		+ 0
$\alpha_2 = 1$		5	35	
$\alpha_3 = 3$			+	- 20

Таблица 2.13

Второе допустимое решение

Потенциал	$\beta_1 = 3$	$\beta_2 = 2$	$\beta_3 = 0$	$\beta_4 = 1$
$\alpha_1 = 0$	³⁻ 30	²⁺ 0	⁴	¹ 20
$\alpha_2 = 1$	²⁺	³⁻ 5	¹ 35	⁵
$\alpha_3 = 3$	³	² 20	⁴	⁴

4. Проверка оптимальности вновь полученного решения. Находим значения потенциалов для данного допустимого решения:

$$\alpha_1 + \beta_1 = 3, \quad \alpha_1 + \beta_2 = 2, \quad \alpha_1 + \beta_4 = 1,$$

$$\alpha_2 + \beta_2 = 3, \quad \alpha_2 + \beta_3 = 1, \quad \alpha_3 + \beta_2 = 2.$$

Пусть $\alpha_1 = 0$, тогда $\beta_1 = 3$, $\beta_2 = 2$, $\beta_4 = 1$, $\alpha_2 = 1$, $\beta_3 = 0$, $\alpha_3 = 0$. Очевидно, что для клетки (2,1) табл. 2.13 неравенство

$\alpha_2 + \beta_1 \leq 2$ не выполняется: $4 \leq 2$, т. е. данное решение не является оптимальным. Процедуры 3 и 4 повторяем до тех пор, пока не достигнем оптимального решения.

Таблица 2.14

Оптимальное решение

Потенциал	$\beta_1 = 3$	$\beta_2 = 2$	$\beta_3 = 0$	$\beta_4 = 1$	a_i
$\alpha_1 = 0$	³ 25	² 5	⁴	¹ 20	50
$\alpha_2 = -1$	² 5	³	¹ 35	⁵	40
$\alpha_3 = 0$	³	² 20	⁴	⁴	20
b_j	30	25	35	20	110

Цикл пересчета для клетки (2,1) показан в табл. 2.13. Третье допустимое решение приведено в табл. 2.14. Здесь же указаны значения рассчитанных для данной итерации потенциалов. Нетрудно убедиться в том, что для всех свободных клеток сумма потенциалов меньше c_{ij} , т. е. получено оптимальное решение. Оно означает, что от первого поставщика к первому потребителю требуется перевезти 25 т груза, ко второму — 5 т, к четвертому — 20 т; от второго поставщика надо перевезти к первому потребителю 5 т груза, к третьему — 35 т; от третьего поставщика необходимо перевезти 20 т груза только ко второму потребителю. Вычислим минимальное значение целевой функции:

$$f_{\min}(x) = 3 \cdot 25 + 2 \cdot 5 + 1 \cdot 20 + 2 \cdot 5 + 1 \cdot 35 + 2 \cdot 20 = \\ = 190 \text{ (тонно-километров).}$$

В приложении ПЗ приводится описание программы (инструкция пользователю) для решения транспортной задачи. Может быть получено сразу оптимальное решение или решение по этапам.

§ 2.7. Задача о перевозках с перегрузкой

Далее, в гл. 3, будет показано, что сетевой подход к решению транспортных задач позволяет использовать алгоритм, рассмотренный в § 2.6, для решения более сложных задач. В транспортной задаче предполагается, что ни в одном маршруте, соединяющем источник (поставщика) с некоторым стоком (потребителем), не могут быть использованы другие источники и стоки в качестве промежуточных пунктов. Если считать допустимой перевозку грузов из источника в сток через другие источники и стоки, то

новая задача может быть сведена к обычной транспортной задаче. Объем вычислений, естественно, возрастает, так как в сеть будут включены дополнительные маршруты, соединяющие каждый источник со всеми другими источниками и каждый сток со всеми другими стоками. Считаем, что транспортные затраты c_{ij} , соответствующие дополнительным маршрутам, известны. Новая задача сводится к модифицированной транспортной задаче, в которой предложения и спрос, соответствующие дополнительным маршрутам, заданы таким образом, что они не влияют на выбор маршрутов, осуществляемый в основном алгоритме. Выполнение последнего требования необходимо, поскольку ограничения на поток по дополнительным маршрутам, задаваемые предложением и спросом, являются фиктивными и вводятся только для вычислительных целей.

Пусть d — минимальная из величин $\sum_i a_i$ и $\sum_j b_j$ — суммарных единиц предложения и спроса, т. е. d — это реальный поток количества груза, перемещающегося по модифицированной сети. Тогда очевидно, что величину спроса \bar{b}_i i -го исходного источника и величину предложения \bar{a}_j j -го исходного стока можно принять равными d , т. е. весь поток может протекать через один источник или через один сток. Поскольку $\bar{b}_i = \bar{a}_j = d$, то исходные величины предложения и спроса должны быть увеличены на \bar{d} . Если увеличить исходные данные на $\bar{d} < d$, то некоторые планы перевозок, допустимые в исходной задаче, в модифицированной задаче станут недопустимыми. Выбор $\bar{d} > d$ не повлияет на решение задачи. Поэтому выбирают d .

Рассмотрим традиционную транспортную задачу (табл. 2.15).

Таблица 2.15

Исходная транспортная задача

Пункт	B_1	B_2	B_3	Запасы
A_1	3	4	7	20
A_2	6	3	2	10
Потребности	10	12	8	30

Пусть в ней каждый источник и каждый сток являются одновременно и промежуточными пунктами (узлами). Здесь

$$\sum_{i=1}^2 a_i = \sum_{j=1}^3 b_j = 30,$$

следовательно, в модифицированной задаче надо выбрать $d = 30$.

Новая задача о перевозках может быть сведена к транспортной задаче со следующими величинами предложения и спроса для стоков и источников:

- 1) $\bar{a}_i = a_i + 30$, $i = 1, 2$, для исходных источников;
- 2) $\bar{b}_i = 30$, $i = 1, 2$, для исходных источников;
- 3) $\bar{a}_j = 30$, $j = 1, 2, 3$, для исходных стоков;
- 4) $\bar{b}_j = b_j + 30$, $j = 1, 2, 3$, для исходных стоков.

Матрица условий модифицированной транспортной задачи для решения задачи перевозок с перегрузкой приведена в табл. 2.16. Транспортные затраты c_{ij} для дополнительных маршрутов (строки \bar{B}_1, \bar{B}_2 и \bar{B}_3 и два последних столбца \bar{A}_1, \bar{A}_2 в табл. 2.16) предполагаются известными.

Таблица 2.16

Модифицированная транспортная задача

Пункт	Исходные стоки			Исходные источники		Запасы
	B_1	B_2	B_3	\bar{A}_1	\bar{A}_2	
A_1	3	4	7	0	5	$20 + 30 = 50$
A_2	6	3	2	3	0	$10 + 30 = 40$
\bar{B}_1	0	5	4	2	5	30
\bar{B}_2	9	0	1	3	2	30
\bar{B}_3	2	4	0	2	6	30
Потребности	$10 + 30 = 40$	$12 + 30 = 42$	$8 + 30 = 38$	30	30	180

Эту задачу можно решить методом потенциалов, рассмотренным в § 2.6. Решив модифицированную транспортную задачу, тем самым решим поставленную задачу о перевозках.

§ 2.8. Целочисленное линейное программирование

Рассмотрим следующую задачу линейного программирования: максимизировать функцию

$$f(x) = a_{00} - a_{01}x_1 - a_{02}x_2 - \dots - a_{0n}x_n \quad (2.9)$$

1) содержит все допустимые целочисленные точки исходной задачи линейного программирования (поскольку является выпуклой оболочкой этих точек);

2) все крайние точки новой области целочисленны.

Поэтому любое базисное оптимальное решение модифицированной задачи ЛП имеет своими компонентами целые числа и является оптимальным решением исходной задачи целочисленного программирования.

Как только будут введены дополнительные ограничения, можно решать модифицированную задачу ЛП любым обычным методом и полученное базисное оптимальное решение автоматически будет целочисленным.

Представленный ниже целочисленный алгоритм обладает следующими свойствами:

1) все дополнительные ограничения сохраняют допустимые точки исходной целочисленной задачи;

2) за конечное число шагов создается достаточное число дополнительных ограничений для того, чтобы оптимальное решение модифицированной задачи было целочисленным;

3) дополнительные ограничения (гиперплоскости) проходят по крайней мере через одну целочисленную точку, хотя и не обязательно находящуюся внутри выпуклой оболочки;

4) каждое новое ограничение сокращает область допустимых решений исходной задачи целочисленного программирования.

Заметим, что оптимальное решение исходной задачи может быть получено прежде, чем допустимая область будет сокращена до области $OFGH$. Оптимальное целочисленное решение определяется пересечением n гиперплоскостей, что обеспечивается процедурой отсечения; некоторые гиперплоскости могут соответствовать ограничениям исходной задачи.

Задачу целочисленного программирования также можно записать в виде табл. 2.17.

Обычно в ограничения задачи (2.9) включают тривиальные соотношения $x_j = -(-x_j)$, $j = 1, 2, \dots, n$, которые образуют первые n строк в симплекс-таблице (ниже строки-заголовка). Запись переменных в виде $-x_1, -x_2, \dots, -x_n$ исторически используется в целочисленном программировании. Будем через a_j , $j = 0, 1, \dots, n$, обозначать j -й столбец текущей таблицы и через a_{ij} , $i = 0, 1, \dots, n + m$, $j = 0, 1, \dots, n$, — элемент i -й строки и j -го столб-

ца таблицы. Предполагается, что все a_{ij} в исходной таблице целые. Следовательно, все слабые переменные x_{n+1}, \dots, x_{n+m} должны быть также неотрицательными целыми числами.

Таблица 2.17

**Запись задачи целочисленного
программирования**

Переменные	1	$-x_1$...	$-x_n$
x_1	0	-1	...	0
...
x_n	0	0	...	-1
x_{n+1}	$a_{n+1,0}$	$a_{n+1,1}$...	$a_{n+1,n}$
...
x_{n+m}	$a_{n+m,0}$	$a_{n+m,1}$...	$a_{n+m,n}$
$f(x)$	a_{00}	a_{01}	...	a_{0n}

Сначала задачу целочисленного программирования рассматривают как задачу ЛП и решают ее с помощью прямого или двойственного симплекс-метода. Используя двойственный симплекс-метод (см. § 2.5), в первую очередь получают переменную, которую исключают из базиса. Она определяется наибольшим по модулю отрицательным элементом столбца сводных членов, поэтому не надо решать вспомогательную задачу ЛП. Чтобы определить переменную, вводимую в базис, рассматривают отношение элементов строки для целевой функции и соответствующих отрицательных элементов разрешающей строки. Наименьшее по значению отношение (в задаче максимизации) определяет переменную, вводимую в базис. Операция замещения проводится методом Жордана. В конце алгоритма получаем $a_{i0} \geq 0$, $i = 1, 2, \dots, n + m$, и $a_{0j} \geq 0$, $j = 1, 2, \dots, n$. Если $a_{i0} \geq 0$ целые для всех i , то найдено оптимальное решение целочисленной задачи. В этом случае решение получается сразу, без использования ограничений целочисленности. Если элементы $a_{i0} \geq 0$, не все целые, то к ограничениям (2.9) добавляют еще одно. Новое ограничение записывают внизу таблицы так, чтобы задача перестала быть прямо допустимой, т. е. $a_{i0} < 0$ для $i = n + m + 1$. Затем используют двойственный симплекс-метод, чтобы получить все $a_{i0} \geq 0$. Если a_{i0} являются нецелыми, в таблицу добавляют новые ограничения до тех пор, пока все $a_{i0} \geq 0$, $i = 1, 2, \dots, n + m$, станут целыми и по-прежнему неотрицательными.

Если после введения дополнительного ограничения текущая таблица перестает быть прямо допустимой, то текущее решение, представляющее собой вершину многогранника решений, не удовлетворяет этому дополнительному ограничению. Другими словами, дополнительное ограничение отсекает часть пространства решений. Если дополнительные ограничения не отсекают ни одной целочисленной точки пространства решений исходной задачи, то вполне вероятно, что после введения достаточного числа дополнительных ограничений вершины суженного множества решений будут целочисленными. Тогда, используя симплекс-метод, можно найти оптимальное целочисленное решение. Трудность задачи состоит в систематическом получении дополнительных ограничений и доказательстве конечности алгоритма.

Каждый раз после проведения итерации симплекс-метода происходит изменение множества небазисных переменных. Изменяется и таблица.

Если требования целочисленности относятся лишь к некоторым переменным, то такие задачи называются *частично целочисленными*.

Решение задачи целочисленного программирования находят последовательным решением задач, каждая из которых получается из предыдущей с помощью дополнительного ограничения

$$\sum_j \gamma_{ij} x_j \geq f(b_i),$$

где γ_{ij} определяются из следующих соотношений:

1) для x_j , которые могут принимать целочисленные значения, имеем

$$\gamma_{ij} = \begin{cases} a_{ij} & \text{при } a_{ij} \geq 0, \\ \frac{f(b_i)}{1 - f(b_i)} |a_{ij}| & \text{при } a_{ij} < 0; \end{cases}$$

2) для x_j , которые могут принимать только целочисленные значения, имеем

$$\gamma_{ij} = \begin{cases} f(a_{ij}) & \text{при } f(a_{ij}) \leq f(b_i), \\ \frac{f(b_i)}{1 - f(b_i)} (1 - f(a_{ij})) & \text{при } f(a_{ij}) > f(b_i). \end{cases}$$

Напомним, что $f(a_{ij}) = a_{ij} - [a_{ij}]$; здесь $[a_{ij}]$ — ближайшее к a_{ij} целое.

Приведем алгоритм решения задачи целочисленного программирования.

Шаг 1. Решить задачу целочисленного программирования как задачу ЛП с помощью прямого или двойственного симплекс-метода. Если получено оптимальное решение задачи ЛП, то все $a_{i0} \geq 0$, $i = 1, 2, \dots, n + m$, и $a_{0j} \geq 0$, $j = 1, 2, \dots, n$.

Шаг 2. Если все a_{i0} целые, то задача решена и решение получено без использования дополнительных ограничений. В противном случае пусть a_{i0} — первая нецелочисленная компонента в столбце a_0 . Тогда i -я строка называется *производящей*. Записать внизу таблицы коэффициенты уравнения, используя данные производящей строки:

$$s = -f_{i0} - \sum_j f_{ij}(-x_j), \quad (2.10)$$

где $f_{i0} = a_{i0} - [a_{i0}]$, $f_{i0} \geq 0$, $f_{ij} = a_{ij} - [a_{ij}]$; здесь $[\alpha]$ — ближайшее к числу α целое.

Например, некоторая базисная переменная имеет нецелочисленное значение, а уравнение для нее имеет вид

$$x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_4 = \frac{7}{2}.$$

Дополнительное ограничение (2.10) в данном случае будет иметь следующий вид:

$$f(1)x_2 + f\left(\frac{1}{2}\right)x_3 + f\left(-\frac{1}{2}\right)x_4 \geq f\left(\frac{7}{2}\right)$$

или

$$\frac{1}{2}x_3 + \frac{1}{2}x_4 \geq \frac{1}{2},$$

т. е. $x_3 + x_4 \geq 1$.

В симплекс-таблицу вносим $x_3 + x_4 - x_5 = 1$, $x_5 \geq 0$ — свободная переменная, которую ввели, чтобы неравенство превратить в равенство.

Для другого уравнения вида

$$\frac{1}{3}x_1 + x_2 + \frac{1}{3}x_4 = \frac{4}{3}$$

ограничение (2.10) имеет вид

$$f\left(\frac{1}{3}\right)x_1 + f(1)x_2 + f\left(\frac{1}{3}\right)x_4 \geq f\left(\frac{4}{3}\right)$$

или

$$\frac{1}{3}x_1 + \frac{1}{3}x_4 \geq \frac{1}{3},$$

т. е. $x_1 + x_4 \geq 1$.

Переменную s называют *слабой переменной Гомори*, а уравнение (2.10) — *отсечением Гомори*. Выполним одну итерацию двойственного симплекс-метода, используя в качестве *ведущей* строки отсечение Гомори (2.10). При этом таблица останется двойственно допустимой. Повторяем шаг 2 до тех пор, пока все a_{i0} , $i = 1, 2, \dots, n + m$, не станут целыми неотрицательными. Если a_{i0} на некотором шаге остаются отрицательными, следующий шаг (двойственного) симплекс-метода выполняется без введения отсечения Гомори. (Если a_{00} становится отрицательным, нулевую строку не выбирают в качестве производящей. Если a_{00} становится нецелым, следует выбрать нулевую строку в качестве производящей.)

В приведенном ниже числовом примере все дополнительные ограничения сохраняются на протяжении вычислений. Это сделано для того, чтобы показать, что эти дополнительные ограничения представляют собой неравенства. Причем, если эти неравенства выразить через исходные небазисные переменные, они будут иметь целые коэффициенты.

Если сохранять все строки, соответствующие слабым переменным Гомори, то эти слабые переменные могут стать базисными. Если слабая переменная Гомори вошла в базис с неотрицательным значением, то соответствующая строка представляет собой неравенство, справедливое при текущем решении, и эта строка может быть вычеркнута. Если слабая переменная Гомори становится базисной с отрицательным значением, соответствующую строку следует использовать в качестве ведущей. Если сохранять все строки, соответствующие всем отсечениям Гомори, то, вообще говоря, потребуется меньшее число дополнительных ограничений, однако увеличение таблицы предпочтительнее введения лишних дополнительных ограничений.

Приведем пример, иллюстрирующий изложенный алгоритм.

Пример. Рассмотрим задачу целочисленного программирования: максимизировать функцию

$$f(x) = 4x_1 + 5x_2 + x_3$$

при условиях

$$3x_1 + 2x_2 \leq 10,$$

$$x_1 + 4x_2 \leq 11,$$

$$3x_1 + 3x_2 + x_3 \leq 13,$$

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

Введя слабые переменные x_4, x_5, x_6 , получаем значения, записанные в табл. 2.18.

Таблица 2.18

Исходные данные задачи

Переменные	1	$-x_1$	$-x_2$	$-x_3$
x_1	0	-1	0	0
x_2	0	0	-1	0
x_3	0	0	0	-1
x_4	10	3	2	0
x_5	11	1	4	0
x_6	13	3	3	1
$f(x)$	0	-4	-5	-1

Решая задачу ЛП (ведущий элемент обведен), получаем новые значения, записанные в табл. 2.19–2.21.

Таблица 2.19

Первая итерация

Переменные	1	$-x_1$	$-x_5$	$-x_3$
x_1	0	-1	0	0
x_2	11/4	1/4	1/4	0
x_3	0	0	0	-1
x_4	18/4	10/4	-2/4	0
x_5	0	0	-1	0
x_6	19/4	9/4	-3/4	1
$f(x)$	55/4	-11/4	5/4	-1

Таблица 2.20

Вторая итерация

Переменные	1	$-x_4$	$-x_5$	$-x_3$
x_1	18/10	4/10	-2/10	0
x_2	23/10	-1/10	3/10	0
x_3	0	0	0	-1
x_4	0	-1	0	0
x_5	0	0	-1	0
x_6	7/4	-9/10	-3/10	<u>1</u>
$f(x)$	187/10	11/10	-7/10	-1

Таблица 2.21

Оптимальное решение задачи ЛП

Ведущий столбец



Переменные	1	$-x_4$	$-x_5$	$-x_6$
x_1	18/10	4/10	-2/10	0
x_2	23/10	-1/10	3/10	0
x_3	7/10	-9/10	-3/10	1
x_4	0	-1	0	0
x_5	0	0	-1	0
x_6	0	0	0	-1
s_1	-7/10	-1/10	<u>-7/10</u>	0
$f(x)$	194/10	2/10	4/10	1

← Произво-
дящая строка

Получено оптимальное решение задачи линейного программирования:

$$f(x) = \frac{194}{10}, \quad x_1 = \frac{18}{10}, \quad x_2 = \frac{23}{10}, \quad x_3 = \frac{7}{10}.$$

Данное решение нецелочисленное. Согласно шагу 2 дописываем в табл. 2.21 строку s_1 , содержащую коэффициенты уравнения отсечения и назначаем в таблице производящую строку и ведущий столбец.

Проводя замену базисных переменных, получим оптимальное целочисленное решение (табл. 2.22).

Таблица 2.22

Оптимальное целочисленное решение

Переменные	1	$-x_4$	$-s_1$	$-x_6$
x_1	2	$3/7$	$-2/7$	0
x_2	2	$-1/7$	$3/7$	0
x_3	1	$-6/7$	$-3/7$	1
x_4	0	-1	0	0
x_5	1	$1/7$	$-10/7$	0
x_6	0	0	0	-1
s_1	0	0	-1	0
$f(x)$	19	$1/7$	$4/7$	1

Оптимальным целочисленным решением является

$$x_1 = 2, \quad x_2 = 2, \quad x_3 = 1, \quad f_{\max}(x) = 19.$$

Выразив x_4 , x_5 и x_6 через исходные небазисные переменные x_1 , x_2 и x_3 , получим неравенство $s_1 \geq 0$ с целыми коэффициентами:

$$s_1 = -\frac{7}{10} + \frac{1}{10}(10 - 3x_1 - 2x_2) + \frac{7}{10}(11 - x_1 - 4x_2) \geq 0,$$

или $x_1 + 3x_2 \leq 8$.

Чтобы получить матрицу, полностью целочисленную, продолжим введение отсечений и решение задачи, результаты этого запишем в виде табл. 2.23, 2.24.

Таблица 2.23

Введение отсечения s_2

Ведущий столбец



Переменные	1	$-x_4$	$-s_1$	$-x_6$
x_1	2	$3/7$	$-2/7$	0
x_2	2	$-1/7$	$3/7$	0
x_3	1	$-6/7$	$-3/7$	1
x_4	0	-1	0	0
x_5	1	$1/7$	$-10/7$	0
x_6	0	0	0	-1
s_1	0	0	-1	0
s_2	0	$-1/7$	$-4/7$	0
$f(x)$	19	$1/7$	$4/7$	1

← Производящая строка

Таблица 2.24

Полностью целочисленная матрица

Переменные	1	$-s_2$	$-s_1$	$-x_6$
x_1	2	3	-2	0
x_2	2	-1	1	0
x_3	1	-6	3	1
x_4	0	-7	4	0
x_5	1	1	-2	0
x_6	0	0	0	-1
s_1	0	0	-1	0
s_2	0	-1	0	0
$f(x)$	19	1	0	1

В приложении Пб приводится описание компьютерной программы (инструкция пользователю) для решения задач целочисленного программирования.

§ 2.9. Задача о назначениях (проблема выбора)

Рассмотрим задачу о распределении механизмов (работников) для выполнения n конкретных действий (заданий) таким образом, чтобы каждый механизм выполнял только одно действие и чтобы при заданной производительности каждого механизма на каждом задании суммарный эффект был максимальным.

Обозначим через c_{ij} , $i, j = 1, 2, \dots, n$, производительность i -го механизма на j -й работе. Тогда данная задача, известная под названием *задачи о назначениях*, будет заключаться в таком выборе элементов из матрицы $\|c_{ij}\|$ по одному из каждой строки и каждого столбца, чтобы их сумма $c_{1j_1} + c_{2j_2} + \dots + c_{nj_n}$, $j_k \neq j_l$ при $k \neq l$, была максимальной.

Обозначив через x_{ij} переменную, равную единице, если i -й механизм выполняет j -ю работу, и равную нулю, если он эту работу не выполняет, мы сведем данную задачу к следующей задаче линейного программирования.

Среди всех неотрицательных целочисленных решений системы $2n$ уравнений

$$x_{i1} + x_{i2} + \dots + x_{in} = 1, \quad i = 1, 2, \dots, n,$$

$$x_{1j} + x_{2j} + \dots + x_{nj} = 1, \quad j = 1, 2, \dots, n,$$

задающих условия того, что каждый механизм выполняет только одну работу и что каждая работа обеспечивается только одним механизмом, найти то решение, которое максимизирует функцию

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

описывающую суммарную производительность всех механизмов. Эту задачу решают с помощью сетевых методов, менее громоздких, чем симплекс-метод (см. гл. 3).

Пример. Имеются три механизма A_1 , A_2 , A_3 , каждый из которых может быть использован на каждом из трех видов работы B_1 , B_2 , B_3 с производительностью (в условных единицах), заданной следующей таблицей:

Узел	B_1	B_2	B_3
A_1	1	2	3
A_2	2	4	1
A_3	3	1	5

Требуется так распределить механизмы по одному на каждом виде работы, чтобы суммарная производительность всех механизмов была максимальной.

Обозначим, как рекомендовалось выше, через x_{ij} переменную, равную единице, если механизм A_i назначен на работу B_j , и равную нулю, если механизм A_i не назначен на работу B_j . Тогда суммарная производительность механизмов описывается функцией

$$f(x) = x_{11} + 2x_{12} + 3x_{13} + 2x_{21} + 4x_{22} + x_{23} + 3x_{31} + x_{32} + 5x_{33}$$

при условиях-ограничениях

$$x_{11} + x_{12} + x_{13} = 1,$$

$$x_{21} + x_{22} + x_{23} = 1,$$

$$x_{31} + x_{32} + x_{33} = 1,$$

$$x_{11} + x_{21} + x_{31} = 1,$$

$$x_{12} + x_{22} + x_{32} = 1,$$

$$x_{13} + x_{23} + x_{33} = 1.$$

Необходимо найти максимум функции $f(x)$ при этих ограничениях.

Запишем условия задачи в виде табл. 2.25, в которой проведем выделение базисных переменных.

Таблица 2.25

Исходная таблица

Переменные	$-x_{11}$	$-x_{12}$	$-x_{13}$	$-x_{21}$	$-x_{22}$	$-x_{23}$	$-x_{31}$	$-x_{32}$	$-x_{33}$	1
0	<u>1</u>	1	1	0	0	0	0	0	0	1
0	0	0	0	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1	1	1	1
0	1	0	0	1	0	0	1	0	0	1
0	0	1	0	0	1	0	0	1	0	1
0	0	0	1	0	0	1	0	0	1	1
$f(x)$	-1	-2	-3	-2	-4	-1	-3	-1	-5	0

После исключения нулевых строк и вычеркивания соответствующих столбцов получим табл. 2.26. Среди свободных членов табл. 2.26 есть отрицательный, поэтому опорное решение еще не получено.

Таблица 2.26

Разделение на базисные и свободные переменные

Переменные	$-x_{22}$	$-x_{23}$	$-x_{32}$	$-x_{33}$	1
x_{11}	<u>-1</u>	-1	-1	-1	-1
x_{21}	1	1	0	0	1
x_{31}	0	0	1	1	1
x_{12}	1	0	1	0	1
x_{13}	0	1	0	1	1
$f(x)$	1	3	3	0	9

Применив метод исключения Жордана (разрешающий элемент -1 отмечен), получим значения, записанные в табл. 2.27.

В табл. 2.27 все свободные члены неотрицательны, поэтому можно перейти к отысканию оптимального решения. Выбирая разрешающий элемент (обведен) и используя модифицированный метод исключения Жордана, получим окончательно значения, записанные в табл. 2.28.

Таблица 2.27

Допустимое базисное решение

Переменные	$-x_{11}$	$-x_{23}$	$-x_{32}$	$-x_{33}$	1
x_{22}	-1	1	1	1	1
x_{21}	1	0	-1	-1	0
x_{31}	0	0	1	1	1
x_{12}	1	-1	0	-1	0
x_{13}	0	1	0	1	1
$f(x)$	-1	4	4	1	10

Таблица 2.28

Оптимальное решение

Переменные	$-x_{21}$	$-x_{23}$	$-x_{32}$	$-x_{33}$	1
x_{22}					1
x_{11}					0
x_{31}					1
x_{12}					0
x_{13}					1
$f(x)$	1	4	3	0	10

Из табл. 2.28 следует, что максимальная суммарная производительность всех механизмов равна 10 (условным единицам) и достигается при

$$x_{11} = 0, \quad x_{12} = 0, \quad x_{13} = 1, \quad x_{21} = 0, \quad x_{22} = 1,$$

$$x_{23} = 0, \quad x_{31} = 1, \quad x_{32} = 0, \quad x_{33} = 0,$$

т. е. механизм A_1 выполняет вид работы B_3 , механизм A_2 — вид работы B_2 и механизм A_3 — вид работы B_1 .

Поскольку в $f(x)$ -строке табл. 2.28 есть нулевой элемент, то полученное оптимальное решение задачи не единственное. Сделав шаг модифицированного метода исключения Жордана с разрешающим элементом из столбца, содержащего нуль в $f(x)$ -строке, можно найти другое оптимальное решение этой же задачи.

§ 2.10. Задачи о покрытии множества

Одним из вариантов задачи размещения производства является *задача о покрытии множества*, т. е. задача определения мест размещения каких-либо элементов и их количества. Например, задача размещения складов, при котором расстояние от склада до каждого потребителя не превышает 100 км, может быть сформулирована как задача о покрытии множества. Другим примером такой задачи является задача определения числа и места размещения на территории города студенческих общежитий, при котором каждый студент тратит на дорогу до учебного заведения не более одного часа, или задача размещения пожарных команд, при котором расстояние до любой точки города «покрывается» за 5 мин.

Задача о покрытии множества может быть записана следующим образом:

$$f(x) = \sum_{j=1}^n c_j x_j \rightarrow \min \quad (2.11)$$

при ограничении

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\geq 1, & i = 1, 2, \dots, m, \\ x_j &\in \{0; 1\}, & j = 1, 2, \dots, n. \end{aligned}$$

Величины a_{ij} называются *коэффициентами покрытия* и принимают значения, равные единице, если i -й потребитель находится в пределах j -й области (т. е. покрывается j -й областью), в противном случае a_{ij} равны нулю. Аналогично, x_j принимает значение, равное единице, если в j -й области расположен некоторый объект, и нулю в противном случае. Ограничения в задаче требуют, чтобы каждый из m потребителей был «покрыт» по крайней мере одним из объектов. Цель в этом случае состоит в том, чтобы «покрыть» потребителей с минимальными затратами, причем c_j — стоимость помещения объекта в j -ю область.

Поясним смысл термина «покрытие». Если имеется комплекс жилых строений и решается задача размещения пожарных команд, то i -е жилое строение считается «покрытым», если пожарная команда находится в пределах пяти минут езды от этого строения; аналогично, если существует i -й потребитель и речь идет о размещении заводов, один из которых должен удовлетворять спросу этого потребителя, то последний считается «покрытым» при усло-

вии, что завод расположен, например, в местах 1, 2 или 3. Таким образом, $a_{i1} = a_{i2} = a_{i3} = 1$, и все остальные a_{ij} равны нулю для $j \neq 1, 2, 3$.

Поскольку (2.11) является и задачей целочисленного линейного программирования, то любой приемлемый для ее решения метод может быть использован для решения задачи (2.11). Однако из-за особой структуры задачи (2.11) специально для ее решения были разработаны методы *неявного перебора секущей плоскости, отсечения, эвристические*.

Обычно задача о покрытии множества при решении проблемы размещения состоит в определении минимального числа объектов, необходимых для удовлетворения (покрытия) запросов некоторого множества потребителей. В подобной ситуации задача (2.11) сводится к так называемой задаче о *полном покрытии*, которая получается из (2.11), если положить все c_{ij} равными единице. Некоторые задачи размещения экстренных служб могут быть сформулированы как задачи о полном покрытии.

Помимо задачи о полном покрытии возможна постановка *задачи о частичном покрытии*. Если задача о полном покрытии состоит в определении мест расположения объектов и их минимального числа, при котором удовлетворяются все потребители, то задача о частичном покрытии связана с определением размещения заданного количества объектов, при котором удовлетворяется максимальное число потребителей. Практически не всегда можно обеспечить такое количество объектов, которое полностью удовлетворяло бы всех потребителей. Обычно количество имеющихся в распоряжении объектов достаточно только для частичного покрытия множества потребителей. В таких ситуациях целесообразно задачу о размещении свести к задаче о частичном покрытии.

Математически задача о частичном покрытии может быть записана в следующем виде:

$$\begin{aligned} \tilde{f}(x) = \sum_{i=1}^m \max a_{ij} x_i \rightarrow \max, \quad j = 1, 2, \dots, n, \\ \sum_{j=1}^n x_j \leq K, \quad x_j \in \{0; 1\}, \quad 1, 2, \dots, n, \end{aligned} \quad (2.12)$$

где K — максимальное число объектов, подлежащих размещению, величины a_{ij} и x_j те же, что и в задаче (2.11).

Из вида функционала $\max a_{ij} x_j$, используемого в целевой функции задачи (2.12), следует, что если некоторое место распо-

ложения потребителя «покрывается» более чем одним размещаемым объектом, то при вычислении функции $\tilde{f}(x)$ учитывается только максимальная величина a_{ij} . Из ограничений в задаче (2.12) следует, что в лучшем случае существует K объектов для размещения. Заметим, что если функция $\tilde{f}(x)$ равна m — числу потребителей, то это значит, что K достаточно велико, чтобы полностью удовлетворить всех потребителей. Таким образом, задача о полном покрытии может быть сведена к задаче о частичном покрытии (2.12) для различных значений K , и решение задачи (2.12) при наименьшем значении K , для которого $\tilde{f}(x) = m$, будет оптимальным решением задачи о полном покрытии.

Однако обычно задачу о полном покрытии не решают таким образом, поскольку существуют более эффективные методы ее решения.

Точное решение задачи (2.12) может быть получено с помощью методов динамического программирования, метода ветвей и границ и двойственных методов. Однако они не приемлемы с вычислительной точки зрения при $n \geq 20$ и $K \geq 10$, и поэтому для решения задачи (2.12) были разработаны эвристические методы.

§ 2.11. Дробно-линейное программирование

В дробно-линейном программировании (ДЛП) целевая функция является дробно-линейной, т. е. имеет вид

$$z = \frac{c^T x + \alpha}{d^T x + \beta},$$

где α и β — скалярные константы, c и d — векторы, x — вектор искомых переменных, $x \in D$,

$$D = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0, b \in \mathbb{R}^m\}.$$

Таким образом, в качестве целевой функции используется отношение двух линейных функций; условия-ограничения задачи остаются линейными: линейные равенства и неравенства.

Обычно предполагают, что знаменатель целевой функции положителен и не обращается в нуль в допустимой области D .

Задачи ДЛП решают в тех приложениях, когда оптимизируются относительные показатели. Особенно часто такие задачи встречаются в области финансовой деятельности: планировании доходов корпораций, управлении статьями банковского баланса и т. п.

Поверхности уровня целевой функции в задаче ДЛП линейны. Покажем это. Пусть значение целевой функции равно z_0 . Тогда

$$c^T x + \alpha = z_0 (d^T x + \beta),$$

или $(c - z_0 d)^T x = z_0 \beta + \alpha$, т. е. получили линейное уравнение, когда знаменатель целевой функции на допустимом множестве D не равен нулю.

Таким образом, если задача ДЛП имеет оптимальное решение, то по крайней мере одна крайняя точка из D будет оптимальной. Однако линии уровня целевой функции расходятся как лучи от множества вращения размерности $n - 2$. Множество вращения — это множество всех точек пересечения нулевой линии уровня числителя $(c^T x + \alpha = 0)$ с нулевой линией уровня знаменателя $(d^T x + \beta = 0)$, т. е. множество точек, удовлетворяющих системе уравнений

$$c^T x = -\alpha,$$

$$d^T x = -\beta.$$

Вращая линию уровня целевой функции в направлении вектора d против часовой стрелки, увеличиваем значение целевой функции; вращая линию уровня по часовой стрелке — уменьшаем значение целевой функции.

Если знаменатель целевой функции отрицателен на допустимом множестве D , то следует дробь умножить на -1 , не изменяя при этом условия максимума или минимума целевой функции.

Для решения задач ДЛП применяют метод преобразования переменных и процедуру обновления целевой функции.

1. Преобразование переменных. Путем преобразования переменных задачу ДЛП при положительном на допустимом множестве D знаменателе сводят к задаче линейного программирования.

Делаем замену переменных $\rho = (d^T x + \beta)^{-1}$ и $y_i = x_i \rho$ для всех i . Задача ДЛП принимает вид

$$c^T y + \alpha \rho \rightarrow \max$$

при условиях

$$\frac{Ay}{\rho} = b,$$

$$(d^T x + \beta)\rho = 1,$$

$$0 \leq y \in \mathbb{R}^n, \quad 0 \leq \rho \in \mathbb{R}^1.$$

Появляется новая переменная ρ . Получили задачу линейного программирования

$$c^T y + \alpha\beta \rightarrow \max$$

при условиях

$$Ay - b\rho = 0,$$

$$d^T y + \beta\rho = 1,$$

$$0 \leq y \in \mathbb{R}^n, \quad 0 \leq \rho \in \mathbb{R}^1,$$

имеющих $m+1$ ограничений и $n+1$ переменных.

Пример. Решить задачу ДЛП

$$z = \frac{x_2 - 5}{-x_1 - x_2 + 9} \rightarrow \max$$

при условиях

$$2x_1 + 5x_2 \geq 10,$$

$$4x_1 + 3x_2 \leq 20,$$

$$-x_1 + x_2 \leq 2,$$

$$x_1, x_2 \geq 0.$$

После замены переменных получим задачу линейного программирования

$$y_2 - 5\rho \rightarrow \max$$

при условиях

$$2y_1 + 5y_2 - 10\rho \geq 0,$$

$$4y_1 + 3y_2 - 20\rho \leq 0,$$

$$-y_1 + y_2 - 2\rho \leq 0,$$

$$-y_1 - y_2 + 9\rho = 1,$$

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

Решая эту задачу симплекс-методом, получим $y_1 = 2/3$, $y_2 = 4/3$, $\rho = 1/3$, или оптимальное решение $x_1 = 2$, $x_2 = 4$.

2. Обновление целевой функции. Задача ДЛП решается как последовательность задач ЛП, где на каждой итерации пересчитывается градиент целевой функции

$$z = \frac{(d^T x + \beta)c - (c^T x + \alpha)d}{(d^T x + \beta)^2}$$

в полученной оптимальной точке. С новым значением градиента решается новая задача линейного программирования. Процесс повторяется до тех пор, пока решение не будет изменяться в заданных пределах.

На допустимом множестве D могут быть следующие случаи:

1) знаменатель принимает на D как положительные, так и отрицательные значения, тогда целевая функция z не имеет ни конечного максимума, ни конечного минимума;

2) знаменатель всюду на D равен нулю, тогда всем точкам из D соответствуют неопределенные значения целевой функции z ;

3) векторы c и d коллинеарны:

а) множество вращения пусто, тогда нулевые линии уровня числителя и знаменателя параллельны, но не совпадают друг с другом; целевая функция z не ограничена сверху и не определена в крайней точке;

б) множество вращения не пусто, тогда числитель и знаменатель имеют идентичные нулевые линии уровня; целевая функция z постоянна на D , кроме некоторых точек, где z имеет значение $0/0$;

4) векторы c и d не коллинеарны:

а) D — подмножество множества вращения, тогда всем точкам из D соответствуют значения z вида $0/0$;

б) знаменатель всюду равен нулю, т. е. $z = 0$ везде, кроме тех точек из D , принадлежащих множеству вращения, где целевая функция z принимает значения $0/0$;

в) на D существуют точки, в которых знаменатель в целевой функции z не равен нулю; тогда могут быть:

- конечные минимумы и конечные максимумы;
- конечные минимумы, но неограниченные максимумы;
- неограниченные минимумы и максимумы.

§ 2.12. Анализ устойчивости оптимального решения задачи линейного программирования

Необходимо знать, как изменится решение задачи линейного программирования при изменении значений ее параметров: коэффициентов целевой функции, элементов матрицы и правой части условий-ограничений. А особенно важно знать, при каких изменениях параметров задачи оптимальное решение этой задачи остается неизменным. Параметры задачи линейного программирования можно варьировать за счет изменения условий функционирования описываемых объектов (например, меняются цены на комплектующие изделия, на трудовые ресурсы, меняется стоимость продукции на рынке и т. д.). Эти изменяющиеся параметры влекут неопределенность параметров задачи и являются в данном случае детерминированными величинами.

В других случаях параметры задачи ЛП являются случайными величинами, и тогда важно знать, как может изменяться решение задачи в зависимости от изменения исходных данных. При этом необходимо иметь, по крайней мере, сведения о математическом ожидании и дисперсии этих случайных величин, если нет возможности оценить их функции распределения. В таком случае неопределенностям значений параметров необходимо указать соответствующую им доверительную вероятность.

Как правило, в подобных случаях для получения ответа решают серию прямых близких задач, изменяя значения параметров.

Особенностью задач ЛП является тот факт, что полученное оптимальное решение может не меняться при изменении значений параметров целевой функции и условий-ограничений в достаточно широких пределах. Более привычным является «непрерывный» вариант: при небольших изменениях параметров задачи обязательно изменяется и решение — координаты точки оптимума. Чтобы оценить изменения решений в задачах ЛП, можно использовать параметрическое программирование, когда определяется поведение решения задачи ЛП в зависимости от параметра t , введенного в коэффициенты целевой функции и в элементы матрицы и правой части условий-ограничений. Однако эти процедуры громоздки даже для одного параметра t , в то же время введением только одного параметра t не удастся описать все возможные изменения параметров задачи. Рассмотрим алгоритм, который позволяет определить допустимые множества значений параметров задачи ЛП, не приводящих к изменению найденного оптимального решения, и таким образом указать диапазон изменения каждого параметра.

Имеем задачу линейного программирования (с одним критерием):

$$z = c^T x \rightarrow \max \quad (2.13)$$

при условиях

$$A_1 x \geq b_1, \quad (2.14)$$

$$A_2 x = b_2, \quad (2.15)$$

$$x \geq 0,$$

где c — вектор коэффициентов целевой функции, z — целевая функция, x — вектор исходных (структурных) переменных, b_1 и b_2 — векторы правых частей условий, A_1 и A_2 — матрицы системы ограничений соответственно неравенств и равенств.

Для аналитического решения задача ЛП записывается в канонической форме:

$$z = c^T x \rightarrow \max$$

при условиях

$$Ax = b, \quad x \geq 0, \quad (2.16)$$

где x — вектор, включающий в себя исходные и дополнительные (слабые) переменные; A — прямоугольная матрица размерности $m \times n$, расширенная за счет столбцов дополнительных переменных, преобразующих неравенства (2.14) в равенства; b — вектор правых частей условий (объединяет векторы b_1 и b_2); c — вектор коэффициентов целевой функции. Будем исследовать устойчивость точки оптимального решения задачи (2.13)–(2.15) при следующих допущениях и предположениях:

1) неопределенность или погрешность имеют только коэффициенты c целевой функции;

2) неопределенность или погрешность имеют только элементы вектора b ;

3) неопределенность или погрешность имеют только элементы матриц A_1 и A_2 (т. е. элементы матрицы A , кроме элементов, относящихся к дополнительным переменным).

Алгоритмы учета других комбинаций неопределенностей основываются на этих случаях.

Будем далее полагать, что задача ЛП решается с помощью симплекс-метода.

1. Неопределенность в коэффициентах целевой функции. Точка оптимума в симплекс-методе определяется условием

$$c_j - z_j \leq 0, \quad z_j = (c_b, a_j), \quad j = 1, 2, \dots, n,$$

где c_b — вектор из элементов c_j , относящихся к базисным переменным; a_j — j -й столбец матрицы A ; (c_b, a_j) — скалярное произведение. Целевая функция имеет вид $z = c_b b$.

Пусть элементы c_j имеют неопределенность Δc_j , т. е. имеют вид $c_j + \Delta c_j$, $j = 1, 2, \dots, n$; тогда условие оптимума будет определяться величиной

$$c_j + \Delta c_j - ((c_b + \Delta c_b), a_j) = (c_j - (c_b, a_j)) + (\Delta c_j - (\Delta c_b, a_j)), \quad (2.17)$$

$$j = 1, 2, \dots, n.$$

Нарушение условия оптимальности зависит от конкретных значений последнего слагаемого в выражении (2.17): если $\Delta c_j - (\Delta c_b, a_j) \leq 0$ для всех $j = 1, 2, \dots, n$, то оптимальное решение не изменяется; при наличии хотя бы одного неравенства $\Delta c_j - (\Delta c_b, a_j) > 0$ возможно изменение оптимального решения.

Система неравенств

$$(c_j - (c_b, a_j)) + (\Delta c_j - (\Delta c_b, a_j)) \leq 0, \quad (2.18)$$

$$j = 1, 2, \dots, n,$$

определяет то множество значений элементов Δc_j , $j = 1, 2, \dots, n$, в которое входят и элементы вектора Δc_b , не нарушающие условия оптимума в данной точке.

Пример 1. Рассмотрим задачу ЛП о выпуске продукции предприятием (см. § 2.4). Математическая модель этой задачи имеет следующий вид:

$$z = 10x_1 + 20x_2 \rightarrow \max \quad (2.19)$$

при условиях

$$\begin{aligned} x_1 + 3,5x_2 &\leq 350, \\ 2x_1 + 0,5x_2 &\leq 240, \\ x_1 + x_2 &\leq 150, \\ x_1 + x_2 &\geq 110, \\ 10x_1 + 20x_2 &\geq 1400, \\ x_1, x_2 &\geq 0. \end{aligned}$$

Оптимальное решение задачи приведено в следующей симплекс-таблице (табл. 2.29).

Таблица 2.29

Оптимальное решение

Базис	b	$-x_1$	$-x_2$	$-x_3$	$-x_4$	$-x_5$	$-x_6$	$-x_7$
x_7	90	0	0	1/5	0	3/5	0	1
x_4	120	0	0	3/5	1	-26/5	0	0
x_6	40	0	0	0	0	1	1	0
x_1	70	1	0	-1/5	0	7/5	0	0
x_2	80	0	1	1/5	0	-2/5	0	0
z	2300	0	0	2	0	6	0	0

Элементы строки z для переменных x_j , $j=1, 2, \dots, 7$, взятых с противоположным знаком, и есть значения $c_j - z_j = c_j - (c_b, a_j)$, по которым судят об оптимальности решения. Различие в знаках обусловлено правилами заполнения симплекс-таблиц.

По условию неопределенность Δc_j содержится только в коэффициентах исходной целевой функции, поэтому $\Delta c_j \neq 0$ для $j=1, 2$, $\Delta c_j = 0$ для $j=3, \dots, 7$, $\Delta c_b = \{0; 0; 0; \Delta c_1; \Delta c_2\}$.

Рассчитаем значения $\Delta_j = \Delta c_j - (\Delta c_b, a_j)$ для различных j :

$$\Delta_j = 0, \quad j=1, 2, 4, 6, 7,$$

$$\Delta_3 = \frac{1}{5}\Delta c_1 - \frac{1}{5}\Delta c_2, \quad j=3,$$

$$\Delta_5 = -\frac{7}{5}\Delta c_1 + \frac{2}{5}\Delta c_2, \quad j=5.$$

Таким образом, на выбор оптимальной точки в данном случае оказывают влияние только коэффициенты при $j=3$ и $j=5$. Точка оптимума не изменится для тех значений Δc_1 и Δc_2 , при которых согласно (2.18) имеем

$$-2 + \frac{1}{5}\Delta c_1 - \frac{1}{5}\Delta c_2 \leq 0,$$

$$-6 - \frac{7}{5}\Delta c_1 + \frac{2}{5}\Delta c_2 \leq 0.$$

Множество D значений Δc_1 , Δc_2 представляет собой часть плоскости, заключенную между двумя лучами, выходящими из точки $(-10, -20)$ в направлениях векторов $v_1 = \{1; 1\}$ и $v_2 = \{1; 3,5\}$.

Проблему устойчивости точки оптимума в задаче линейного программирования при неопределенности только в коэффициентах целевой функции можно рассмотреть геометрически, на плоскости. Плоскость, вектор нормали которой определен коэффициентами целевой функции, в точке оптимума можно повернуть таким образом, чтобы она коснулась граней выпуклого многогранника, содержащих эту точку. Диапазон изменения углов поворота плоскости и будет определять допустимый разброс значений коэффициентов целевой функции.

Для простоты изложения материала все уравнения линейных поверхностей запишем в нормальном (нормированном) виде. Таким образом, мы будем использовать направляющие косинусы плоскостей; в трехмерном пространстве это $\cos \alpha$, $\cos \beta$, $\cos \gamma$. Чтобы определить грани многогранника, содержащие точку оптимума, подставим координаты точки оптимума в ограничения (2.14) и (2.15). Те ограничения, которые являются равенствами, определяют искомые грани.

Пусть некоторая грань в трехмерном пространстве задана уравнением

$$ax + by + cu + d = 0;$$

нормальное уравнение этой плоскости имеет вид

$$(\cos \alpha)x + (\cos \beta)y + (\cos \gamma)u - p = 0,$$

где x , y , u — текущие координаты; a , b , c , d — параметры уравнения плоскости; p — параметр, определяющий расстояние до плоскости от начала координат;

$$\cos \alpha = \frac{a}{\sqrt{a^2 + b^2 + c^2}}, \quad \cos \beta = \frac{b}{\sqrt{a^2 + b^2 + c^2}}, \quad \cos \gamma = \frac{c}{\sqrt{a^2 + b^2 + c^2}}.$$

Для плоскости, соответствующей целевой функции с параметрами c_1 , c_2 , c_3 , получим

$$\cos \alpha_0 = \frac{c_1}{\sqrt{c_1^2 + c_2^2 + c_3^2}}, \quad \cos \beta_0 = \frac{c_2}{\sqrt{c_1^2 + c_2^2 + c_3^2}},$$

$$\cos \gamma_0 = \frac{c_3}{\sqrt{c_1^2 + c_2^2 + c_3^2}}.$$

В рассматриваемом случае важны не сами значения c_1 , c_2 , c_3 , а их отношения:

$$\frac{c_1}{c_2} = \frac{\cos \alpha_0}{\cos \beta_0}, \quad \frac{c_1}{c_3} = \frac{\cos \alpha_0}{\cos \gamma_0}, \quad \frac{c_2}{c_3} = \frac{\cos \beta_0}{\cos \gamma_0}. \quad (2.20)$$

Если некоторая i -я грань многогранника имеет направляющие косинусы $\cos \alpha_i$, $\cos \beta_i$, $\cos \gamma_i$, то всякая плоскость, направляющие косинусы которой будут иметь значения между $\cos \alpha_0$ и $\cos \alpha_i$, $\cos \beta_0$ и $\cos \beta_i$, $\cos \gamma_0$ и $\cos \gamma_i$, не изменит положение оптимальной точки. Анализируя таким образом все грани, которым принадлежит оптимальная точка, найдем диапазоны отношений параметров (коэффициентов) a , b и c , которым должны удовлетворять плоскости, проходящие через оптимальную точку и не изменяющие оптимальное решение задачи. Отсюда легко получить возможный диапазон неопределенностей Δc_j в значениях коэффициентов целевой функции, не влияющих на решение задачи.

Пример 2. Рассмотрим задачу, описанную в примере 1.

Определим грани многоугольника, которым принадлежит точка оптимума. Подставим оптимальное решение задачи $x_1 = 70$, $x_2 = 80$ в исходную модель (2.19) и получим, что оптимальной точкой является точка пересечения прямых $x_1 + 3,5x_2 = 350$ и $x_1 + x_2 = 150$.

Вектор нормали целевой функции имеет координаты $\{10; 20\}$, вектор нормали первой прямой — $\{1; 3,5\}$, вектор нормали второй прямой — $\{1; 1\}$. Вектор нормали целевой функции можно повернуть так, чтобы он совпал либо с вектором нормали первой прямой, либо с вектором нормали второй прямой, т. е. согласно (2.20) имеем

$$\frac{\cos \beta_1}{\cos \alpha_1} \geq \frac{c_2 + \Delta c_2}{c_1 + \Delta c_1} \geq \frac{\cos \beta_2}{\cos \alpha_2},$$

$$\cos \alpha_1 = 0,275, \quad \cos \beta_1 = 0,962, \quad \cos \alpha_2 = \cos \beta_2 = 0,707.$$

Окончательно условие (2.20) принимает вид

$$3,5 \geq \frac{c_2 + \Delta c_2}{c_1 + \Delta c_1} \geq 1.$$

Таким образом, если отношение коэффициентов целевой функции лежит в пределах от 1 до 3,5, то координаты оптимальной точки не изменятся.

Полученный результат полезно сравнить с приведенным выше аналитическим расчетом.

2. Неопределенность только в координатах вектора правой части. При изменении значений правой части исходной системы (2.14), (2.15) в процессе решения изменяется только столбец свободных членов b и значение целевой функции z . В точке оптимума столбец свободных членов не содержит отрицательных элементов. Таким образом, неопределенность в координатах вектора правой части до тех пор не влияет на оптимальное решение, пока не появятся отрицательные элементы в столбце свободных членов.

Представим матрицу полного ранга A из условий (2.16) в виде двух блоков N и B : $A = (N; B)$, где N состоит из небазисных столбцов A , а B — из базисных. Аналогично, векторы c и x состоят из базисных и небазисных координат: $c = (c_N; c_B)$, $x = (x_N; x_B)$. Следовательно, имеем

$$Ax = b,$$

$$Nx_N + Bx_B = b,$$

$$B^{-1}Nx_N + B^{-1}Bx_B = B^{-1}b,$$

$$x_B = B^{-1}b,$$

так как $x_N = 0$. Отсюда следует, что столбец свободных членов $b + \Delta b$ на любой итерации может быть получен умножением матрицы B^{-1} на вектор b в исходной постановке задачи (2.16). Сам же блок B формируется из тех столбцов исходной матрицы A из условий (2.16), номера которых на данной итерации определяют базисные переменные. С другой стороны, j -й столбец в матрице A (матрица A после преобразований становится матрицей Y) для данной итерации имеет вид

$$y_j = B^{-1}a_j,$$

где a_j — j -й столбец исходной матрицы A из условий (2.16).

Столбцы матрицы B^{-1} — это столбцы тех переменных текущей матрицы Y , которые были базисными в исходной таблице (как правило, это последние m столбцов), так как $Y = B^{-1}A$, а столбцы A , относящиеся к базису исходной таблицы, образуют единичную матрицу, т. е. матрица B^{-1} всегда присутствует в процессе решения. Исходный вектор b и неопределенность его значе-

ния Δb задаются по условию задачи. Таким образом получают новый столбец свободных членов. Если исходный вектор равен $b + \Delta b$, то можно найти новый столбец свободных членов:

$$B^{-1}(b + \Delta b) = B^{-1}b + B^{-1}\Delta b = x_B + B^{-1}\Delta b.$$

Поскольку в столбце свободных членов оптимального решения из симплекс-таблицы не должно быть отрицательных элементов, то условие стабильности примет вид

$$x_B + B^{-1}\Delta b \geq 0.$$

Пример 3. Рассмотрим описанную в примере 1 задачу, предполагая, что исходный вектор правой части $b = \{350; 240; 150; -110; -1400\}$ имеет неопределенность $\Delta b = \{\Delta b_1; \Delta b_2; \Delta b_3; \Delta b_4; \Delta b_5\}$, т. е. необходимо рассматривать вектор

$$\{350 + \Delta b_1; 240 + \Delta b_2; 150 + \Delta b_3; -110 + \Delta b_4; -1400 + \Delta b_5\}.$$

и определить диапазон значений $\Delta b_1, \dots, \Delta b_5$, при которых оптимальное решение не изменится.

Матрица B^{-1} занимает пять последних столбцов симплекс-таблицы оптимального решения (см. табл. 2.29):

$$B^{-1} = \begin{pmatrix} 1/5 & 0 & 3/5 & 0 & 1 \\ 3/5 & 1 & -26/5 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ -1/5 & 0 & 7/5 & 0 & 0 \\ 1/5 & 0 & -2/5 & 0 & 0 \end{pmatrix}.$$

Условия стабильности оптимального решения данной задачи имеют вид

$$90 + \Delta b_1 + \frac{3}{5}\Delta b_3 \geq 0,$$

$$120 + \Delta b_2 - \frac{26}{5}\Delta b_3 \geq 0,$$

$$40 + \Delta b_3 \geq 0,$$

$$70 + \Delta b_4 + \frac{7}{5}\Delta b_3 \geq 0,$$

$$80 + \Delta b_5 - \frac{2}{5}\Delta b_3 \geq 0.$$

Отсюда находятся те комбинации значений величин Δb_i , $i = 1, 2, \dots, 5$, при которых оптимальное решение не изменится.

3. Неопределенность в элементах матриц A_1 и A_2 . Если неопределенность имеет место в отдельных элементах матриц A_1 и A_2 , то можно воспользоваться тем, что элементы матрицы Y оптимального решения определяются через матрицу B^{-1} и исходную матрицу A из условия (2.16):

$$y_j = B^{-1}a_j, \quad j = 1, 2, \dots, n.$$

Пусть только один элемент a_{ij} имеет неопределенность Δa_{ij} . В матрице Y оптимального решения получим элемент $y_{ij} = B_i^{-1}(a_j + \Delta a_j)$. По знаку рассматриваемого элемента y_{ij} принимается соответствующее решение. Этот подход приемлем для небольшого числа элементов матриц, имеющих неопределенность.

Рассмотрим более общий метод учета неопределенностей в элементах матриц A_1 , A_2 . Пусть все элементы матриц A_1 и A_2 имеют неопределенности Δa_{ij} , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$. Перенесем неопределенности Δa_{ij} в правую часть уравнений и объединим их с неопределенностями Δb_i . В таком случае надо вычислить неопределенность i -й линейной комбинации $\sum_j a_{ij}x_j$ и добавить ее

к неопределенности правой части Δb_i . Теперь возникает вопрос, каким образом (согласно какой гипотезе) вычислить неопределенность линейной комбинации и согласно какой гипотезе присоединить ее к неопределенности правой части. От выбора этих гипотез зависит конкретный алгоритм вычисления неопределенностей линейной комбинации.

Рассмотрим две гипотезы. В первом случае неопределенность Δa_{ij} является детерминированной величиной (например, показывает изменение цены некоторого ресурса); во втором (в задачах ЛП с другим физическим содержанием) неопределенность Δa_{ij} будет случайной величиной, для которой известны математическое ожидание и дисперсия. Тогда в первом случае неопределенность i -й линейной комбинации $\sum_j a_{ij}x_j$ будет равна $\sum_j \Delta a_{ij}x_j$, а полная неопределенность i -й координаты вектора правой части есть

$$\Delta_{\Sigma} b_i = \sum_j \Delta a_{ij}x_j + \Delta b_i. \quad (2.21)$$

Во втором случае мы будем исходить из того, что все участвующие в расчете неопределенности носят случайный характер и можно вычислить дисперсию линейной комбинации, полагая Δa_{ij} , $i = 1, \dots, m$, $j = 1, 2, \dots, n$, независимыми случайными величинами с известными дисперсиями $D(a_{ij})$, а затем суммировать дисперсию линейной комбинации и неопределенности правой части Δb_i , полагая величину $(\Delta b_i)^2$ равной дисперсии b_i , т. е. $D(b_i) = (\Delta b_i)^2$.

Таким образом, дисперсию линейной комбинации определяем по формуле

$$D\left(\sum_{j=1}^n a_{ij} x_j\right) = \sum_{j=1}^n x_j^2 D(a_{ij}),$$

а полную неопределенность $\Delta_{\Sigma} b_i$ (погрешность) i -й координаты вектора правой части системы (2.16) — по формуле

$$\Delta_{\Sigma} b_i = \sqrt{D\left(\sum_{j=1}^n a_{ij} x_j\right) + D(b_i)} = \sqrt{\sum_{j=1}^n x_j^2 D(a_{ij}) + (\Delta b_i)^2}. \quad (2.22)$$

В формулах (2.21) и (2.22) значения x_j , $j = 1, 2, \dots, n$, берутся из симплекс-таблицы оптимального решения. Далее анализируем ситуацию так же, как в случае существования неопределенности только в правой части. Очевидно, что исследование можно проводить подобным образом и для случая $\Delta b_i = 0$, $i = 1, 2, \dots, m$.

СЕТЕВЫЕ И ПОТОКОВЫЕ ЗАДАЧИ

§ 3.1. Основные определения и приложения сетевых и потоковых моделей

Многие задачи линейного программирования могут быть сформулированы как *сетевые*. Примером таких задач является транспортная задача (см. ранее рис. 1.2). Однако можно указать и такие сетевые задачи, которые нельзя сформулировать в виде задачи ЛП. Например, таковой является *задача коммивояжера*.

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

Основные определения в сетевых и потоковых задачах

Введем основные определения и понятия. *Сеть* состоит из множества *узлов* N_i , $i=1,2,\dots,k$ (называемых также *вершинами*, или *точками соединения*), и множества *дуг* A_{ij} , $i,j=1,2,\dots,k$ (называемых также *звеньями* или *ребрами*), которые связывают узлы N_i и N_j . Если дуга имеет определенную ориентацию (направление), то ее называют *ориентированной*, или *направленной*.

Сеть называют *связной*, если при любом разбиении множества узлов сети на подмножества X и \bar{X} найдется дуга A_{ij} или дуга A_{ji} , связывающая i -й узел N_i , принадлежащий подмножеству X , $N_i \in X$, и j -й узел N_j , принадлежащий подмножеству \bar{X} , $N_j \in \bar{X}$.

Будем рассматривать только *связные сети* и будем считать, что между любыми двумя узлами N_i и N_j имеется не более одной ориентированной дуги A_{ij} и одной ориентированной дуги A_{ji} либо имеется только одна неориентированная дуга A_{ij} . Одну неориентированную дугу можно заменить двумя ориентированными.

Существование *петель* (дуг, ведущих из некоторого узла в тот же узел) исключается.

Последовательность узлов и дуг $N_1, A_{12}, N_2, A_{23}, \dots, N_{k-1}, A_{k-1,k}, N_k$ сети называют *цепью* (*ориентированной цепью*), ведущей из узла N_1 в узел N_k . Если $N_1 = N_k$, то такую последовательность называют *ориентированным циклом*. Цепь называют *простой*, если она не содержит циклов.

Путем называют последовательность $N_1, A_{12}, N_2, A_{23}, \dots, N_{k-1}, A_{k-1,k}, N_k$, где N_1, N_2, \dots, N_k — узлы сети и либо $A_{i,i+1}$, либо $A_{i+1,i}$, $i = 1, 2, \dots, k-1$, — дуга сети. Путь отличается от цепи тем, что при движении от узла N_1 к узлу N_k можно пройти дугу сети и в направлении, противоположном ее ориентации. Для неориентированных сетей понятия цепи и пути совпадают. В частности, циклами являются контуры.

Контуром называют конечную цепь, начальный и конечный узлы которой совпадают. Очевидно, что циклы являются *замкнутыми путями*, а контуры *замкнутыми цепями*. *Вырожденный цикл* называют *петлей*. Петля образуется одним узлом и одной дугой и поэтому является как контуром, так и циклом.

В связной сети для любых двух различных узлов существует по крайней мере один соединяющий их путь (или цепь). Частным случаем связных ориентированных и неориентированных сетей являются *деревья*. Если множество всех узлов и дуг задать в виде графа $G = (N, A)$, где N — множество узлов, A — множество дуг, то дерево определяют как связное подмножество (*подграф*) G_1 множества (графа) G , не содержащее циклов, т. е. для любых двух узлов дерева существует единственный путь, соединяющий их. В сети, содержащей n узлов, подграф из k узлов ($k \leq n$) является деревом, если выполнены любые два из следующих условий:

- 1) подграф является связным;
- 2) подграф не имеет циклов;
- 3) число дуг в подграфе равно $k - 1$.

Остовным связующим деревом (*остовом*) называют дерево, содержащее все узлы сети. Если сеть содержит n узлов, дерево с n узлами и $n - 1$ дугами является *остовом*. *Кратчайшим* (*максимальным*) *остовом графа* (*сети*) называют дерево с минимальным (*максимальным*) весом среди всех связующих деревьев этого графа. *Вес дерева* определяют как сумму весов (длин) его дуг. *Весом* (*длиной*) *дуги* называют число, соответствующее некоторой харак-

теристике дуги (расстоянию, стоимости и т. п.). Каждой дуге A_{ij} (или (i, j)) ставят в соответствие положительное число b_{ij} , называемое *пропускной способностью дуги (ребра)*.

В сети выделяют два специальных узла: один из них называют *источником* N_s , а другой — *стоком* N_t .

Сеть можно рассматривать как водопроводную систему, в которой трубы соответствуют дугам, источник воды — источнику N_s , сток воды — стоку N_t , а соединения между трубами — остальным узлам сети. В качестве пропускной способности дуги выступает поперечное сечение трубы.

Потоком из источника N_s (или s) в сток N_t (или t) сети называют множество неотрицательных чисел x_{ij} , поставленных в соответствие некоторой дуге сети, если эти числа удовлетворяют следующим линейным условиям-ограничениям:

$$\sum_i x_{ij} - \sum_k x_{jk} = \begin{cases} -v, & \text{если } j = s, \\ 0, & \text{если } j \neq s, j \neq t, \\ v, & \text{если } j = t, \end{cases} \quad (3.1)$$

$$v \geq 0, \quad 0 \leq x_{ij} \leq b_{ij}, \quad i, j \in \mathbb{N}. \quad (3.2)$$

Здесь первая сумма берется по дугам, ведущим в узел N_j , а вторая сумма — по дугам, ведущим из узла N_j . Неотрицательное число v называют *величиной потока*. Число x_{ij} называют *потоком* по дуге A_{ij} , или *дуговым потоком*. Ограничения (3.1) выражают тот факт, что в каждый узел (кроме источника и стока) приходит столько потока, сколько из него уходит (условие сохранения потока). Ограничение (3.2) означает, что поток x_{ij} по дуге ограничен пропускной способностью дуги b_{ij} .

Очевидно, что задача нахождения величины максимального потока в любой сети является задачей ЛП: максимизировать функцию $v = \sum_j x_{sj}$ при условиях-ограничениях (3.1), (3.2). Однако в си-

лу специфики задачи сетевые методы решения здесь оказываются более эффективными, чем симплекс-метод, применяемый для решения общей задачи линейного программирования.

Если сеть является цепью $N_1, A_2, N_2, \dots, N_k$ с источником N_1 и стоком N_k , максимальная величина потока, который может быть пропущен через сеть, ограничивается минимальной пропускной способностью дуг этой сети. Дуга с минимальной пропускной способностью является *узким местом* в сети. В произвольной сети

узкое место определяют *разрезом*. Пусть X — некоторое подмножество узлов сети, \bar{X} — дополнение подмножества X (объединение X и \bar{X} определяет множество узлов сети). *Разрезом* (X, \bar{X}) называют множество всех дуг A_{ij} , для которых $N_i \in X$, $N_j \in \bar{X}$. Таким образом, разрез представляет собой множество дуг, удаление которых из сети превращает сеть в несвязанную. Разрез (X, \bar{X}) называют *разделяющим* узлы N_s и N_t (или *отделяющим* узел N_s от узла N_t), если $N_s \in X$, $N_t \in \bar{X}$. *Пропускной способностью* $c(X, \bar{X})$ разреза, или *величиной* разреза, называют сумму $\sum_{i,j} b_{ij}$, которую берут по всем ориентированным дугам, соединяющим $N_i \in X$ и $N_j \in \bar{X}$. При определении разреза учитывают все дуги между узлами подмножества X и подмножества \bar{X} , а при определении пропускной способности разреза — только пропускные способности дуг, соединяющих узлы $N_i \in X$ с узлами $N_j \in \bar{X}$, ориентированные дуги, соединяющие узлы $N_i \in X$ с узлами $N_j \in \bar{X}$, не учитывают. Поэтому в общем случае $c(X, \bar{X}) \neq c(\bar{X}, X)$.

Ясно, что в силу ограничений (3.1), (3.2) максимальный поток меньше или равен пропускной способности любого разреза, разделяющего N_s и N_t :

$$v \leq \sum_{\substack{N_i \in X \\ N_j \in \bar{X}}} x_{ij} - \sum_{\substack{N_i \in \bar{X} \\ N_j \in X}} x_{ji}.$$

В любой сети величина максимального потока из источника N_s в сток N_t всегда равна минимальной пропускной способности всех разрезов, разделяющих N_s и N_t . Разрез, разделяющий N_s и N_t и обладающий минимальной пропускной способностью, называют *минимальным разрезом*. Эти утверждения объединены в теореме о максимальном потоке и минимальном разрезе.

Обозначим через F_{st} множество неотрицательных чисел x_{ij} , удовлетворяющих ограничениям (3.1), (3.2). Будем называть путь из N_s в N_t *увеличивающим поток* F_{st} , если $x_{ij} < b_{ij}$ на всех прямых дугах и $x_{ij} > 0$ на всех обратных дугах этого пути. Поток F_{st} максимален тогда и только тогда, когда не существует пути, увеличивающего поток F_{st} .

Проверить унимодулярность матрицы A , вычислив всевозможные миноры, довольно сложно. Существуют достаточные, но не необходимые условия абсолютной унимодулярности матрицы A , которые проверить гораздо легче. Матрица A абсолютно унимодулярна, если:

- 1) каждый ее элемент равен 0, +1, -1;
- 2) каждый ее столбец содержит не более двух ненулевых элементов;
- 3) строки матрицы A можно разбить на два непересекающихся множества R_1 и R_2 таким образом, что:
 - а) если столбец из A содержит два ненулевых элемента одного знака, то один из них входит в R_1 , другой — в R_2 ;
 - б) если столбец из A содержит два ненулевых элемента с противоположными знаками, то оба они входят либо в R_1 , либо в R_2 .

Приложение потоковых моделей

На практике 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 *кратчайшей цепи* заключается в следующем. Заданы множества дуг и узлов. Каждой дуге A_{ij} поставлена в соответствие величина c_{ij} , равная стоимости единицы потока по этой дуге. Требуется найти цепь из источника N_s в сток N_t , минимизирующую стоимость единицы потока из N_s в N_t . Примером подобной задачи может быть задача о замене устаревшего оборудования новым. Здесь минимизируются общие затраты на закупку и обслуживание оборудования, если ликвидационная стоимость при различных сроках службы оборудования, а также эксплуатационные расходы и расходы на техническое обслуживание и текущий ремонт в каждый промежуток времени считаются известными. Обычно задают и величину периода планирования.

Значительное место в приложениях занимают сетевые задачи, связанные с планированием и составлением расписания выполнения работ по осуществлению больших проектов, по проведению научных исследований и опытных, конструкторских разработок. Сетевые задачи решают также при составлении расписаний движения транспорта, календарного планирования, распределения ресурсов и т. д.

В более общей интерпретации дугу сети представляют как звено в механизме, предназначенном для транспортировки потока, который может протекать по дуге в единицу времени, что определяет пропускную способность дуги.

Многие обобщения задачи о максимальном потоке, по существу, сводят к поиску максимального потока в сети, к легко осуществимому поиску некоторых цепей сети. Примером одной из таких задач является *задача о потоке в сети с несколькими источниками и стоками*, когда заданы мощности источников и возможности (спрос) стоков. В данной задаче множество всех узлов разбивается на подмножества источников S , промежуточных узлов R и стоков T . Каждому узлу $N_i \in S$ ставится в соответствие неотрицательное число a_i (*возможности*), а каждому узлу $N_j \in T$ — неотрицательное число b_j (*спрос*).

Возникает система ограничений

$$\sum_j x_{ij} - \sum_k x_{ki} = a_i, \quad N_i \in S,$$

$$\sum_j x_{ij} - \sum_k x_{ki} = 0, \quad N_i \in R,$$

$$\sum_i x_{ij} - \sum_k x_{kj} = b_j, \quad N_j \in T,$$

$$0 \leq x_{ij} \leq b_{ij}.$$

Если потоку разрешается течь из любого источника в любой сток, то эта задача легко сводится к задаче с одним источником и одним стоком путем добавления одного дополнительного источника и одного дополнительного стока (рис. 3.1). Помимо этого добавляют новые ориентированные дуги, ведущие из дополнительного источника S во все стоки N_i и имеющие пропускные способности a_i , а также ориентированные дуги с пропускными

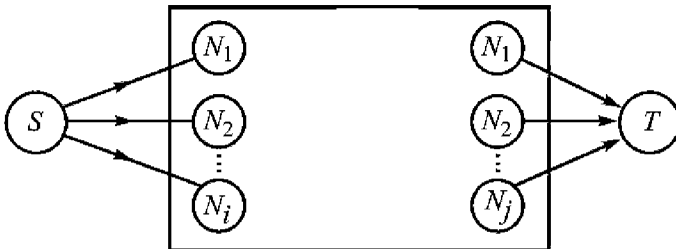


Рис. 3.1. Сведение сетевой задачи к задаче с одним источником и одним стоком

способностями b_{ij} , ведущие из каждого источника N_j в дополнительный сток T . Теперь задача об удовлетворении требуемого спроса заданным предложениям (возможностям) сводится к нахождению максимального потока в расширенной сети.

Если в сети с несколькими источниками и стоками поток должен идти из определенных источников в заданные стоки, возникает так называемая *задача о многопродуктовых потоках в сети*.

Получим еще одно обобщение *задачи о потоке*, если для каждой дуги введем ограничение на пропускную способность и сверху, и снизу (рис. 3.2), т. е.

$$0 \leq l_{ij} \leq x_{ij} \leq b_{ij}.$$

Требуется определить, существует ли поток из источника N_s в сток N_t , удовлетворяющий на дугах ограничениям сверху и снизу.

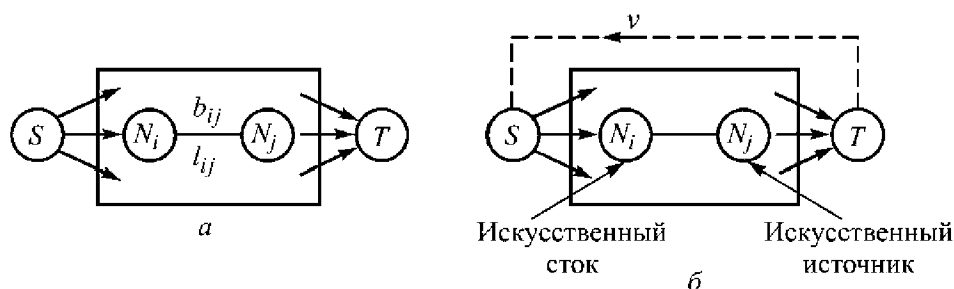


Рис. 3.2. Исходная (а) и расширенная (б) сети

Пусть в сети имеется только одна ориентированная дуга A_{ij} с ограниченным снизу дуговым потоком l_{ij} . Расширим сеть, добавив два новых узла — искусственный источник N_j с предложением l_{ij} и искусственный сток N_i с таким же спросом l_{ij} . Пропускную способность дуги A_{ij} при этом изменим: если она была равна b_{ij} , то в новой сети она станет равной $b_{ij} - l_{ij}$. Добавим, кроме того, ориентированную дугу из N_t в N_s с бесконечной пропускной способностью. Будем искать в расширенной сети максимальный поток из источника N_j в сток N_i . Если величина этого потока в расширенной сети больше или равна l_{ij} , а поток по дуге A_{ts} равен x_{ts} , то в исходной сети существует такой поток из N_s в N_t величиной $v = x_{ts}$, что $l_{ij} \leq x_{ij}$. Исходная и расширенная сети приведены на рис. 3.2, а, б. Если в сети имеется несколько дуг, обладающих нижними границами для дуговых потоков, то следует:

1) ввести несколько искусственных источников и стоков (см. рис. 3.2, б);

2) задачу с несколькими источниками и стоками свести к задаче с одним источником и одним стоком введением дополнительного источника и дополнительного стока, как это было сделано в предыдущей задаче (см. рис. 3.1).

§ 3.2. Задача о покупке автомобиля

Одним из примеров задачи о *наискратчайшей сети*, или о *сети наименьшей стоимости*, является задача об оптимизации расходов на приобретение и эксплуатацию автомобиля.

Со временем эксплуатационные расходы на содержание автомобиля заметно возрастают, да и сам автомобиль устаревает морально и технически. Возникают вопросы: когда следует заменить автомобиль и что принять за критерий, определяющий необходимость его замены? Выберем в качестве критерия общие затраты на покупку и содержание автомобиля за некоторый период времени t . Для определенности будем полагать следующее: $t = 8$ лет; в начальный момент автомобиля нет; решение о покупке автомобиля может приниматься в начале каждого года исходя из затрат на его приобретение, эксплуатационных расходов за период, в течение которого автомобиль будет использоваться, и ликвидационной стоимости автомобиля в момент его замены на новый. Предположим, что замена автомобиля должна совершаться по крайней мере каждые 4 года. На рис. 3.3 изображена сеть для рассматриваемой задачи.

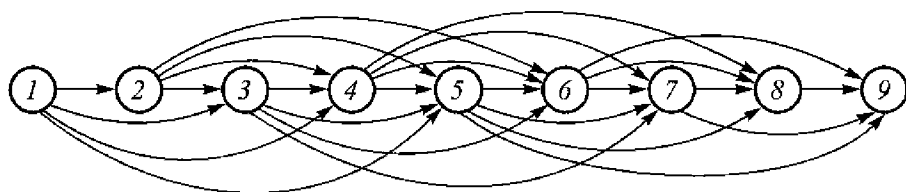


Рис. 3.3. Сеть для задачи о покупке автомобиля

Началу каждого года соответствует узел. Если автомобиль куплен в начале i -го года, а заменен в начале j -го года, то такому варианту соответствует дуга (i, j) . Общие затраты в течение этого периода (на дуге (i, j)) выразятся формулой

$$c_{ij} = P_i + \sum_{k=1}^{j-1} m_k - s_j,$$

где P_i — стоимость автомобиля в начале i -го года, m_k — эксплуатационные расходы в течение k -го года, s_j — ликвидационная стоимость автомобиля в начале j -го года. Значения c_{ij} для рассматриваемого случая приведены в табл. 3.1.

Таблица 3.1

Общие затраты по дугам

Шаг	2	3	4	5	6	7	8	9
1	1450	2750	3850	4750				
2		1700	3200	4450	5450			
3			1850	3425	4750	5800		
4				1950	3650	5050	6150	
5					2075	3875	5350	6500
6						2200	4100	5650
7							2325	4325
8								2450

Вследствие инфляции и модернизации автомобилей возрастает стоимость автомобиля и увеличиваются расходы на его содержание. Оптимальному решению данной задачи соответствует кратчайшая цепь из источника $s=1$ в сток $t=9$. Поэтому цепь можно рассматривать как цепь, минимизирующую стоимость единицы потока из узла 1 в узел 9 при условии, что стоимость единицы потока по дуге (i, j) равна c_{ij} , $i=1, \dots, 8$, $j=2, \dots, 9$. Для решения этой задачи воспользуемся алгоритмом Дейкстры (Дикстры).

В алгоритме Дейкстры минимизируется либо стоимость, либо время прохождения единицы потока по данной цепи. Каждой дуге ориентированной сети ставится в соответствие обобщенная стоимость дуги c_{ij} . Фиктивным («бесплатным») дугам приписывается стоимость $c_{ij}=0$, а каждой паре узлов i, j , для которых не существует дуги, соединяющей их, — стоимость $c_{ij}=\infty$.

Математическая постановка этой задачи следующая:

$$\sum_i \sum_j c_{ij} x_{ij} \rightarrow \min$$

при условиях

$$\sum_j x_{sj} - \sum_j x_{js} = 1, \quad \sum_j x_{ij} - \sum_j x_{ji} = 0, \quad i \neq s, \quad i \neq t,$$

$$\sum_j x_{tj} - \sum_j x_{jt} = -1, \quad f_{ij} \geq 0, \quad c_{ij} \geq 0.$$

Получили задачу ЛПП с неизвестными x_{ij} — величинами потока между i -м и j -м узлами. Согласно первому равенству единица потока вытекает из источника s , а согласно третьему — единица потока втекает в сток t . Второе равенство гарантирует сохранение потока при протекании по сети (естественно, не рассматриваются источник s и сток t). В качестве кратчайшей цепи может быть взята последовательность смежных дуг (i, j) , для которых $x_{ij} = 1$.

В алгоритме Дейкстры узлам приписывают либо временные, либо постоянные пометки. Первоначально каждому узлу, исключая источник, приписывают пометку, соответствующую длине кратчайшей дуги, ведущей из источника в данный узел. Источнику приписывают постоянную пометку, значение которой равно нулю. Каждому узлу, в который нельзя попасть непосредственно из источника, приписывают временную пометку ∞ , а остальным узлам — временные пометки c_{sj} , $j \neq s$. Если определено, что узел принадлежит кратчайшей цепи, его пометка становится постоянной. Алгоритм Дейкстры основан на следующем простом факте: если известна кратчайшая цепь из узла s (источника) в узел j и узел k принадлежит этой цепи, то кратчайшая цепь из s в k является частью первоначальной цепи, оканчивающейся в узле k . Алгоритм начинает работать при $j = s$. Затем величина j увеличивается на единицу; при $j = t$ алгоритм завершает свою работу. Итерационная процедура алгоритма состоит в следующем. Для заданного узла j обозначим «длину» кратчайшей цепи c_{sj} из источника s в узел j . Если эта «длина» не может быть «улучшена», то соответствующее значение называют *постоянной пометкой*. В противном случае «длину» называют *временной пометкой*. Сначала постоянную пометку присваивают только источнику. Каждая другая пометка является временной и ее величина равна длине дуги, ведущей из источника в соответствующий узел. Для определения «ближайшего» к источнику узла выберем временную пометку с минимальным значением и объявим ее постоянной пометкой. Для получения этой постоянной пометки необходимо следующее.

1. Рассмотреть оставшиеся узлы с временной пометкой. Сравнить величину каждой временной пометки с суммой величины последней из постоянных пометок и «длины» дуги, ведущей из соответствующего постоянно помеченного узла в рассматриваемый узел. Минимальная из двух сравниваемых величин определяется

как новая временная пометка рассматриваемого узла. Если величина старой временной пометки меньше второй из сравниваемых величин, то пометка остается прежней.

2. Среди временных пометок выбрать ту, значение которой минимально, и объявить ее постоянной пометкой. Если при этом постоянную пометку приписывают узлу t (стоку), то завершаем работу, в противном случае переходим к п. 1.

Для решения примера рассмотрим следующий алгоритм.

Шаг 0. Припишем источнику (узлу 1) постоянную пометку 0, а узлам 2, ..., 9 — временные пометки $c_{sj} = \infty$. Постоянные пометки заключены в квадратные скобки.

Шаг 1. Для источника остается постоянная пометка 0, для узла 2 временная пометка $c_{12} = 1450$, для узла 3 $c_{13} = 2750$, для узла 4 $c_{14} = 3850$, для узла 5 $c_{15} = 4750$. Для остальных узлов временные пометки равны ∞ . Последним из постоянно помеченных узлов является узел 1 (единственный, иначе надо было взять узел с минимальной пометкой).

Шаг 2. С узлом 1 непосредственно связаны узлы 2, ..., 5. Им припишем новые временные пометки, равные $\delta_2 = 0 + c_{12} = 1450$, $\delta_3 = 0 + c_{13} = 2750$, $\delta_4 = 0 + c_{14} = 3850$, $\delta_5 = 0 + c_{15} = 4750$ соответственно. Поскольку величина δ_2 минимальна из всех, то узлу 2 припишем постоянную пометку $\delta_2 = 1450$.

Шаг 3. Рассмотрим движение из узла 2; оставляем постоянные пометки 0 для узла 1 и 1450 для узла 2. Узлам 3, ..., 6, непосредственно связанным с узлом 2, припишем соответственно временные пометки $\delta_3 = c_{23} = 2750$, $\delta_4 = c_{24} = 3850$, $\delta_5 = c_{25} = 4750$, $\delta_6 = c_{26} = 6900$, минимальные из возможных временных пометок при движении из узла 1 в узлы 3, ..., 6. Например,

$$\delta_3 = \delta_2 + c_{23} = 1450 + 1700 = 3150 > 2750,$$

$$\delta_4 = 1450 + 3200 = 4650 > 3850, \dots$$

Выбираем $\delta_3 = 2750$, $\delta_4 = 3850$, ... Остальным узлам припишем временные пометки, равные ∞ .

Шаг 4. Поскольку минимальное значение временных пометок δ_3 , δ_4 , δ_5 , δ_6 равно 2750, узлу 3 припишем постоянную пометку $\delta_3 = 2750$.

Дальнейшие шаги повторяем аналогично шагам 3 и 4. Результаты вычислений сведем в табл. 3.2.

Таблица 3.2

Результаты вычислений по алгоритму Дейкстры

Шаг	1	2	3	4	5	6	7	8	9
0	[0]	∞	∞	∞	∞	∞	∞	∞	∞
1	[0]	1450	2750	3850	4750	∞	∞	∞	∞
2	[0]	[1450]	2750	3850	4750	∞	∞	∞	∞
3	[0]	[1450]	2750	3850	4750	6900	∞	∞	∞
4	[0]	[1450]	[2750]	3850	4750	6900	∞	∞	∞
5	[0]	[1450]	[2750]	3850	4750	6900	8550	∞	∞
6	[0]	[1450]	[2750]	[3850]	4750	6900	8550	∞	∞
7	[0]	[1450]	[2750]	[3850]	4750	6900	8550	10 000	∞
8	[0]	[1450]	[2750]	[3850]	[4750]	6900	8550	10 000	∞
9	[0]	[1450]	[2750]	[3850]	[4750]	6825	8550	10 000	11 250
10	[0]	[1450]	[2750]	[3850]	[4750]	[6825]	8550	10 000	11 250
11	[0]	[1450]	[2750]	[3850]	[4750]	[6825]	8550	10 000	11 250
12	[0]	[1450]	[2750]	[3850]	[4750]	[6825]	[8550]	10 000	11 250
13	[0]	[1450]	[2750]	[3850]	[4750]	[6825]	[8550]	10 000	11 250
14	[0]	[1450]	[2750]	[3850]	[4750]	[6825]	[8550]	[10 000]	11 250
15	[0]	[1450]	[2750]	[3850]	[4750]	[6825]	[8550]	[10 000]	11 250
16	[0]	[1450]	[2750]	[3850]	[4750]	[6825]	[8550]	[10 000]	[11 250]

Обратим внимание на выбор временной пометки δ_6 на шаге 9 (см. табл. 3.2). Последняя постоянная пометка $\delta_5 = 4750$ присвоена узлу 5, тогда для узла 6 временная пометка будет $\delta_6 = \delta_5 + c_{56} = 4750 + 2075 = 6825 < 6900$ (временная пометка δ_6 на шаге 8). Поэтому на шаге 9 берут минимальное значение $\delta_6 = 6825$.

Из решения задачи следует, что минимальные общие затраты на покупку и содержание автомобиля составляют за рассмотренный период 11 250 усл. ед. Кратчайшая цепь состоит из дуг, для каждой из которых разность между значениями постоянных пометок ее концевых узлов равна длине этой дуги: $[\delta_j] = [\delta_i] + c_{ij}$. Последнее соотношение можно использовать рекурсивно, двигаясь от стока t к источнику s . Определив узел, непосредственно предшествующий t в кратчайшей цепи, будем повторять данную процедуру до тех пор, пока не достигнем узла s . В нашем случае первое значение c_{ij} , совпадающее с разностью между значениями посто-

янных пометок, равно 6500, т. е. узел 5 непосредственно предшествует стоку t (узлу 9) в кратчайшей цепи. Следующим узлом кратчайшей цепи (и исходным) будет узел 1. Отсюда для минимизации общих затрат автомобиль следует заменять в начале первого, пятого и девятого годов.

§ 3.3. Задача о многополюсной кратчайшей цепи

Рассмотрим задачу нахождения *кратчайших цепей между всеми парами узлов сети*. Кратчайшей цепью между двумя произвольными узлами является цепь, стоимость единицы потока по которой минимальна. Поскольку направление потока в неориентированных дугах нельзя определить заранее, то каждую такую дугу следует заменить двумя ориентированными дугами с противоположными направлениями и длинами (стоимостями), равными длине неориентированной дуги. Предполагается, что длины дуг могут быть как положительными, так и отрицательными. Однако длина, или стоимость, каждого цикла или контура должна быть неотрицательной. Алгоритм решения данной задачи разработан Флойдом. Пусть $N = \{1, 2, \dots, n\}$ — множество узлов, а c_{ij} — количественный параметр (длина, стоимость) дуги (i, j) , направленной от узла i к узлу j . Обозначим через d_{ik}^* длину кратчайшей цепи из узла i в узел k .

Алгоритм Флойда заключается в следующем. Первоначально за длину d_{ik} кратчайшей цепи между двумя произвольными узлами i и k (между которыми могут быть и промежуточные узлы) принимают длину дуги (i, k) , соединяющей эти узлы. Затем последовательно проверяют всевозможные промежуточные узлы, расположенные между i и k . Если длина цепи, проходящей через некоторый промежуточный узел, меньше текущего значения d_{ik} , то переменной d_{ik} присваивают новое значение; если $d_{ik} > d_{ij} + d_{jk}$, то значение d_{ik} заменяют значением $d_{ij} + d_{jk}$. Такую процедуру повторяют для всевозможных пар узлов, пока не будут получены все значения d_{ik}^* .

В алгоритме Флойда начальным значением переменной d_{ik} является величина c_{ij} , а затем данная оценка последовательно улучшается до тех пор, пока не будет найдена кратчайшая цепь между

узлами i и k . Алгоритм Флойда позволяет решать задачу о многополюсной кратчайшей цепи (пути) для сети из n узлов за n итераций. Обозначим через d_{ik}^j оценку длины кратчайшей цепи из узла i в узел k , полученную на j -й итерации, и рассмотрим следующую задачу.

Задача. Необходимо соединить восемь объектов многополюсной цепью кратчайшей длины, причем один из объектов (узел 8) может быть только направляющим информацию, а остальные могут и направлять, и получать информацию без каких-либо ограничений (рис. 3.4). На рис. 3.4 каждый объект представлен узлом, а каждая линия — дугой. Ориентированные дуги соответствуют распределительным звеньям, которые могут быть использованы для передачи информации только в указанном направлении. Числа, приписанные дугам, соответствуют расстоянию между объектами и образуют матрицу $\|c_{ij}\|$. Требуется найти для каждого объекта кратчайшие пути, связывающие его с другими объектами.

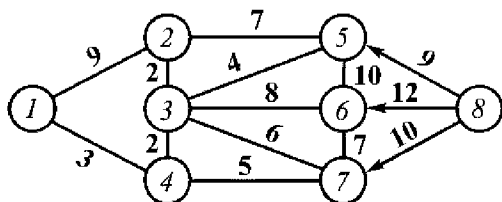


Рис. 3.4. Пример задачи о многополюсной кратчайшей сети

Решение. Воспользуемся алгоритмом Флойда. Поскольку $n = 8$, то число итераций в алгоритме будет равно восьми. На каждой итерации строят матрицу $D^j = \|d_{ik}^j\|$ длин кратчайших путей, которые содержат текущие оценки длин кратчайших цепей, где $D^0 = \|c_{ij}\|$, и матрицу маршрутов R^j , служащую для нахождения промежуточных узлов (если таковые имеются) кратчайших цепей. На j -й итерации имеем матрицу $R^j = \|r_{ik}^j\|$, где r_{ik}^j — первый промежуточный узел кратчайшей цепи из i в k , выбираемый из множества $\{1, 2, \dots, j\}$, $i \neq j \neq k$, и матрицу $R^0 = \|r_{ik}^0\|$, где $r_{ik}^0 = k$. Узел r_{ik}^j может быть получен из следующего соотношения:

$$r_{ik}^j = \begin{cases} j, & \text{если } d_{ik}^{j-1} > d_{ij}^{j-1} + d_{jk}^{j-1}, \\ r_{ik}^{j-1} & \text{— в противном случае.} \end{cases}$$

Строим матрицу длин кратчайших цепей D^0 и матрицу маршрутов R^0 , отсутствие связи помечаем знаком ∞ , нулем обозначаем связи внутри одного узла. Получаем

$$D^0 = \begin{pmatrix} 0 & 9 & \infty & 3 & \infty & \infty & \infty & \infty \\ 9 & 0 & 2 & \infty & 7 & \infty & \infty & \infty \\ \infty & 2 & 0 & 2 & 4 & 8 & 6 & \infty \\ 3 & \infty & 2 & 0 & \infty & \infty & 5 & \infty \\ \infty & 7 & 4 & \infty & 0 & 10 & \infty & \infty \\ \infty & \infty & 8 & \infty & 10 & 0 & 7 & \infty \\ \infty & \infty & 6 & 5 & \infty & 7 & 0 & \infty \\ \infty & \infty & \infty & \infty & 9 & 12 & 10 & 0 \end{pmatrix}, \quad R^0 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}.$$

Итерация 1. Выбираем базовый узел $j=1$. В матрице D^0 вычеркиваем первую (базовую) строку и первый (базовый) столбец. Чтобы определить, приведет ли использование узла l к более коротким цепям, необходимо исследовать элементы матрицы D^0 с помощью *трехместной операции*:

$$d_{ik}^j = \min\{d_{ik}^{j-1}; d_{ij}^{j-1} + d_{jk}^{j-1}\}, \quad i \neq j \neq k.$$

Если $d_{ik}^j = \infty$, т. е. j -й элемент базового столбца равен ∞ , то $d_{ik}^j = d_{jk}^{j-1}$. Если $d_{jk}^{j-1} = \infty$, т. е. j -й элемент базовой строки равен ∞ , то $d_{ik}^{j-1} = d_{ik}^j$. Если $d_{ik}^{j-1} \neq \infty$ и одно из двух значений d_{ij}^{j-1} или d_{jk}^{j-1} превышает значения d_{ik}^{j-1} , то замену также производить не следует. Столбцы 3, 5, 7 и 8 содержат элементы, равные ∞ и принадлежащие базовой строке. Строки 3, 5, ..., 8 также содержат элементы, равные ∞ и принадлежащие базовому столбцу. Значит, исследовать необходимо элементы d_{22}^0 , d_{24}^0 , d_{42}^0 , d_{44}^0 . Поскольку диагональные элементы можно не рассматривать, необходимо исследовать лишь оценки d_{24}^0 и d_{42}^0 . С помощью трехместной операции получим

$$d_{24}^1 = \min\{d_{24}^0; d_{21}^0 + d_{14}^0\} = \min\{\infty; 9 + 3\} = 12,$$

$$d_{42}^1 = \min\{d_{42}^0; d_{41}^0 + d_{12}^0\} = \min\{\infty; 3 + 9\} = 12.$$

Оценки d_{24}^1 и d_{42}^1 лучше оценок d_{24}^0 и d_{42}^0 и должны быть внесены в матрицу D^1 , а в матрице маршрутов надо положить

$r_{24}^j = j$, $r_{24}^1 = 1$ и $r_{42}^1 = 1$. Остальные элементы матриц D^0 и R^0 остаются без изменений. Получим

$$D^1 = \begin{pmatrix} 0 & 9 & \infty & 3 & \infty & \infty & \infty & \infty \\ 9 & 0 & 2 & 12 & 7 & \infty & \infty & \infty \\ \infty & 2 & 0 & 2 & 4 & 8 & 6 & \infty \\ 3 & 12 & 2 & 0 & \infty & \infty & 5 & \infty \\ \infty & 7 & 4 & \infty & 0 & 10 & \infty & \infty \\ \infty & \infty & 8 & \infty & 10 & 0 & 7 & \infty \\ \infty & \infty & 6 & 5 & \infty & 7 & 0 & \infty \\ \infty & \infty & \infty & \infty & 9 & 12 & 10 & 0 \end{pmatrix}, \quad R^1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 1 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 1 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}.$$

Итерация 2. Определим узел 2 как базовый, т. е. проверим, приведет ли его использование к более коротким цепям, и выделим в матрице D^1 вторую строку и второй столбец. Здесь столбцы 6, 7, 8 содержат элементы, равные ∞ и принадлежащие базовой строке, а строки 6, 7, 8 — элементы, равные ∞ и принадлежащие базовому столбцу. Столбцы и строки 6, 7, 8 не рассматриваем. Исключаем и диагональные элементы. Остается исследовать лишь элементы d_{13}^1 , d_{14}^1 , d_{15}^1 , d_{31}^1 , d_{34}^1 , d_{35}^1 , d_{41}^1 , d_{43}^1 , d_{45}^1 , d_{51}^1 , d_{53}^1 , d_{54}^1 . Нетрудно проверить, что здесь улучшены могут быть только оценки d_{13}^1 , d_{15}^1 , d_{31}^1 , d_{45}^1 , d_{51}^1 , d_{54}^1 , равные ∞ :

$$d_{13}^2 = \min\{d_{13}^1; d_{12}^1 + d_{23}^1\} = \min\{\infty; 9 + 2\} = 11,$$

$$d_{15}^2 = \min\{d_{15}^1; d_{12}^1 + d_{25}^1\} = \min\{\infty; 9 + 7\} = 16,$$

$$d_{31}^2 = \min\{d_{31}^1; d_{32}^1 + d_{21}^1\} = \min\{\infty; 2 + 9\} = 11,$$

$$d_{45}^2 = \min\{d_{45}^1; d_{42}^1 + d_{25}^1\} = \min\{\infty; 12 + 7\} = 19,$$

$$d_{51}^2 = \min\{d_{51}^1; d_{52}^1 + d_{21}^1\} = \min\{\infty; 7 + 9\} = 16,$$

$$d_{54}^2 = \min\{d_{54}^1; d_{52}^1 + d_{24}^1\} = \min\{\infty; 7 + 12\} = 19.$$

Таким образом, $r_{13}^2 = r_{15}^2 = r_{31}^2 = r_{45}^2 = r_{51}^2 = r_{54}^2 = 2$; остальные элементы матрицы R^2 остаются без изменения. Новые матрицы имеют вид

$$D^2 = \begin{pmatrix} 0 & 9 & 11 & 3 & 16 & \infty & \infty & \infty \\ 9 & 0 & 2 & 12 & 7 & \infty & \infty & \infty \\ 11 & 2 & 0 & 2 & 4 & 8 & 6 & \infty \\ 3 & 12 & 2 & 0 & 19 & \infty & 5 & \infty \\ 16 & 7 & 4 & 19 & 0 & 10 & \infty & \infty \\ \infty & \infty & 8 & \infty & 10 & 0 & 7 & \infty \\ \infty & \infty & 6 & 5 & \infty & 7 & 0 & \infty \\ \infty & \infty & \infty & \infty & 9 & 12 & 10 & 0 \end{pmatrix}, R^2 = \begin{pmatrix} 1 & 2 & 2 & 4 & 2 & 6 & 7 & 8 \\ 1 & 2 & 3 & 1 & 5 & 6 & 7 & 8 \\ 2 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 1 & 3 & 4 & 2 & 6 & 7 & 8 \\ 2 & 2 & 3 & 2 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}.$$

Итерация 3. Определяем, приведет ли использование узла 3 к более коротким цепям. Берем узел 3 в качестве базового и выделяем 3-ю строку и 3-й столбец в матрице D^2 . Исключаем диагональные элементы, элементы 8-го столбца и 8-й строки, исследуем оставшиеся элементы матрицы D^2 , получаем новые матрицы D^3 и R^3 .

Аналогично получаем матрицы D^j и R^j , $j = 4, \dots, 8$. Матрицы D^j , R^j , $j = 5, \dots, 8$, остаются без изменений. Следовательно, оптимальное решение соответствует матрицам D^5 и R^5 , определяющим и оптимальное расстояние для передачи между объектами, и последовательность передачи информации:

$$D^5 = \begin{pmatrix} 0 & 7 & 5 & 3 & 9 & 13 & 8 & \infty \\ 7 & 0 & 2 & 4 & 6 & 10 & 8 & \infty \\ 5 & 2 & 0 & 2 & 4 & 8 & 6 & \infty \\ 3 & 4 & 2 & 0 & 6 & 10 & 5 & \infty \\ 9 & 6 & 4 & 6 & 0 & 10 & 10 & \infty \\ 13 & 10 & 8 & 10 & 10 & 0 & 7 & \infty \\ 8 & 8 & 6 & 5 & 10 & 7 & 0 & \infty \\ 18 & 5 & 13 & 15 & 9 & 12 & 10 & 0 \end{pmatrix}, R^5 = \begin{pmatrix} 1 & 4 & 4 & 4 & 4 & 4 & 4 & 8 \\ 4 & 2 & 3 & 3 & 3 & 3 & 3 & 8 \\ 4 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 3 & 4 & 3 & 3 & 7 & 8 \\ 4 & 3 & 3 & 3 & 5 & 6 & 7 & 8 \\ 4 & 3 & 3 & 3 & 5 & 6 & 7 & 8 \\ 4 & 3 & 3 & 4 & 3 & 6 & 7 & 8 \\ 5 & 5 & 5 & 5 & 5 & 6 & 7 & 8 \end{pmatrix}.$$

Например, определим кратчайшую цепь из узла 1 в узел 5. По матрице D^5 находим, что длина этой цепи $d_{15}^5 = 9$. Чтобы найти соответствующую последовательность узлов, рассмотрим матрицу R^5 . Имеем $r_{15}^5 = 4$, т. е. узел 4 является первым промежуточным узлом в кратчайшей цепи из узла 1 в узел 5. Теперь определим, какой узел следует за узлом 4 в кратчайшей цепи из узла 4 в узел 5. Поскольку $r_{45}^5 = 3$, то за узлом 4 следует узел 3. Аналогично за уз-

лом 3 следует узел 5, так как $r_{35}^5 = 5$. Значит, кратчайшая цепь из узла 1 в узел 5 определяется последовательностью узлов 1, 4, 3, 5.

§ 3.4. Анализ сложности алгоритмов поиска кратчайших путей

В алгоритмах Дейкстры (§ 3.2) и Флойда (§ 3.3) выполняются только две элементарные операции: сложение и сравнение.

Рассмотрим сеть, содержащую n узлов. В наихудшем случае в алгоритме Дейкстры конечный узел сети будет n -м по счету узлом, которому приписывают постоянную пометку. Пусть в некоторый момент работы алгоритма m узлам приписаны постоянные пометки, а $m - n$ узлам — временные. Для определения $(m + 1)$ -го узла, которому должна быть приписана постоянная пометка, необходимо вычислить $n - m$ временных пометок, выполнив при этом каждый раз операции сложения и сравнения. После вычисления новых значений временных пометок необходимо также найти минимальное среди них, чтобы определить пометку, которая должна стать постоянной. Данная процедура минимизации состоит из $n - m - 1$ сравнений. Таким образом, если имеется m узлов с постоянными пометками, то число элементарных операций, которое необходимо выполнить для того, чтобы еще одному узлу приписать постоянную пометку, равно $3(n - m) - 1 \approx 3(n - m)$. Общее число элементарных операций, выполнение которых необходимо для завершения работы алгоритма в наихудшем случае, можно определить по формуле

$$\sum_{m=1}^n 3(n - m) = 3 \sum_{m=1}^{n-1} (n - m) = 3[(n - 1) + (n - 2) + \dots + 1] = \frac{3n(n - 2)}{2}.$$

В алгоритме Флойда на каждой итерации суммарное число элементов, значение которых должно быть оценено с помощью трехместной операции, можно вычислить с учетом того, что:

- 1) общее число элементов матрицы равно n^2 ;
- 2) суммарное число элементов в базовой строке и базовом столбце равно $2n - 1$;
- 3) число нулевых элементов на главной диагонали равно n и для них не надо получать оценку;

4) один элемент базовой строки и один элемент базового столбца расположены на главной диагонали;

5) анализ каждого элемента требует выполнения одной операции сложения и одной операции сравнения;

6) максимальное число операций, выполняемых на одной итерации алгоритма, приблизительно равно $2n(n-3)$;

7) поскольку число итераций равно числу узлов n , общее число элементарных операций в алгоритме Флойда в наихудшем случае равно $2n^2(n-3)$.

§ 3.5. Венгерский алгоритм задачи о назначениях

В практических приложениях часто необходимо:

– так распределить рабочих по рабочим местам, чтобы время изготовления изделия было минимальным;

– так разместить датчики по объектам, чтобы информация о работе объектов была максимальной;

– так распределить экипажи самолетов по рейсам, чтобы время простоя техники было минимальным, и т. д.

Особенность этой задачи заключается в том, что каждый ресурс (рабочий, датчик, экипаж) используется ровно один раз и каждому объекту будет приписан ровно один ресурс.

Решение задачи может быть записано в виде двумерной матрицы $X = \|x_{ij}\|$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, m$, где m — число объектов или ресурсов. Искомая переменная определяется следующим образом:

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

Критерий эффективности (целевая функция) задачи включает в себя элементы матрицы стоимостей c_{ij} — затраты, связанные с назначением i -го ресурса на j -й объект. Для любого недопустимого назначения соответствующую ему стоимость полагают равной достаточно большому числу M . Допустимое решение задачи называют *назначением*. Для заданного значения m ресурсов и объектов существует $m!$ допустимых решений. Допустимое решение строят путем выбора ровно одного элемента в каждой

строке матрицы $X = \|x_{ij}\|$ и ровно одного элемента в каждом столбце этой матрицы.

В задаче о назначениях требуется минимизировать общую стоимость назначений, т. е. необходимо минимизировать функцию

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

при ограничениях:

а) каждый ресурс используется ровно один раз, т. е.

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

б) каждому объекту будет приписан ровно один ресурс, т. е.

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

Очевидно, что задача о назначениях является частным случаем транспортной задачи при единичных значениях параметров a_i и b_i . Поэтому для решения задачи о назначениях можно воспользоваться любым алгоритмом линейного программирования или методом потенциалов, который применялся в § 2.6 для решения транспортной задачи. Однако для подобных задач разработан так называемый *венгерский алгоритм*, который наиболее эффективно использует специфику задачи о назначениях.

Задача. Разместить четыре датчика на четырех объектах таким образом, чтобы стоимость такого размещения была минимальна. Матрица стоимости назначений имеет вид

$$C = \|c_{ij}\| = \begin{pmatrix} 2 & 10 & 9 & 7 \\ 15 & 4 & 14 & 8 \\ 13 & 14 & 16 & 11 \\ 4 & 15 & 13 & 19 \end{pmatrix}.$$

Решение. В процессе решения задачи о назначениях используют тот факт, что если к каждому элементу i -й строки добавлять действительное число γ_i , а к каждому элементу j -го столбца — действительное число δ_j , то минимизация целевой функции

$$\sum_j \sum_i c_{ij} x_{ij}$$

эквивалентна минимизации функции

$$\sum_j \sum_i d_{ij} x_{ij}, \quad d_{ij} = c_{ij} + \gamma_i + \delta_j.$$

Шаг 1 (редукция строк). Данный шаг предназначен для получения в матрице стоимости назначений возможно большего числа нулевых элементов. Для этого в каждой строке из всех элементов вычитают минимальный:

$$\|c_{ij}^1\| = \begin{pmatrix} 0 & 8 & 7 & 5 \\ 11 & 0 & 10 & 4 \\ 2 & 3 & 5 & 0 \\ 0 & 11 & 9 & 15 \end{pmatrix}.$$

Шаг 2 (редукция столбцов). В каждом столбце вычитают минимальные элементы соответствующего столбца:

$$\|c_{ij}^2\| = \begin{pmatrix} \boxed{0} & 8 & 2 & 5 \\ 11 & \boxed{0} & 5 & 4 \\ 3 & 4 & 0 & 0 \\ \boxed{0} & 11 & 4 & 15 \end{pmatrix}.$$

Шаг 3 (определение назначений). Если в полученной матрице стоимостей можно выбрать по *одному* нулевому элементу так, что соответствующее этим элементам решение будет допустимым, то данное назначение оптимально.

Рассмотрим сначала строки матрицы стоимости $\|c_{ij}^2\|$. Строки 1, 2 и 4 содержат по одному нулю. Взяв эти строки в порядке возрастания их номеров, произведем вначале назначение, соответствующее элементу c_{11}^2 , и вычеркнем нулевой элемент c_{41}^2 , так как в первом столбце может быть только один нулевой элемент. Затем проведем назначение, соответствующее элементу c_{22}^2 . Элемент c_{41}^2 был уже вычеркнут.

Рассмотрим столбцы матрицы стоимости назначений с учетом вычеркнутых нулевых элементов по строкам. Третий и четвертый столбцы содержат по одному нулевому элементу. Проведем третье назначение, соответствующее элементу c_{33}^2 . В четвертом столбце назначение невозможно, так как нулевой элемент стоит в третьей строке, а назначение в ней выполнено. В результате матрица стоимостей приняла следующий вид:

$$\|c_{ij}^3\| = \begin{pmatrix} \boxed{0} & 8 & 2 & 5 \\ 11 & \boxed{0} & 5 & 4 \\ 2 & 3 & \boxed{0} & 0 \\ 0 & 11 & 4 & 15 \end{pmatrix}.$$

Поскольку полного назначения нулевой стоимости не может быть получено, необходимо провести дальнейшую модификацию редуцированной матрицы стоимости.

Шаг 4 (модификация редуцированной матрицы). Получим новые нулевые элементы. Определим для редуцированной матрицы стоимостей минимальное множество строк и столбцов, содержащих нулевые элементы, и найдем минимальный элемент вне данного множества по следующему плану.

1) Определим количество нулей в строках и столбцах матрицы стоимости $\|c_{ij}^3\|$; соответственно имеем: 1, 1, 2 и 1 в строках и 2, 1, 1 и 1 в столбцах;

2) Максимальное число нулей (по два) содержат строка 3 и столбец 1. Удаляем элементы строки 3.

3) Число оставшихся нулей после удаления строки 3 равно 1, 1 и 1 в строках 1, 2 и 4 соответственно. Для столбцов число оставшихся нулей равно 2, 1, 0 и 0. Выбираем столбец 1 и удаляем его элементы. Остался только один невычеркнутый нуль — элемент c_{22}^3 , поэтому можно удалить либо строку 2, либо столбец 2. Удаляем элементы строки 2 и получаем следующую матрицу-таблицу:

$$\begin{pmatrix} | & 8 & 2 & 5 \\ | & - & - & - \\ | & - & - & - \\ | & 11 & 4 & 15 \end{pmatrix}.$$

Если число линий (горизонтальных и вертикальных), необходимое для того, чтобы вычеркнуть нулевые элементы, равно числу строк или столбцов матрицы стоимости, то существует назначение нулевой стоимости.

4) Минимальным элементом полученной матрицы-таблицы является 2. Вычитая его из всех оставшихся элементов, в результате получаем новую матрицу-таблицу

$$\begin{pmatrix} | & 6 & 0 & 3 \\ | & - & - & - \\ | & - & - & - \\ | & 9 & 2 & 13 \end{pmatrix}.$$

Прибавляя значение 2 ко всем элементам матрицы $\|c_{ij}^3\|$, расположенными на пересечении вертикальной и двух горизонтальных штриховых линий (столбца 1 со строками 2 и 3), т. е. к элементам c_{21}^3 и c_{31}^3 , получаем новую редуцированную матрицу стоимостей:

$$\|c_{ij}^4\| = \begin{pmatrix} 0 & 6 & 0 & 3 \\ 13 & 0 & 5 & 4 \\ 4 & 3 & 0 & 0 \\ 0 & 9 & 2 & 13 \end{pmatrix},$$

оставив остальные элементы матрицы стоимости $\|c_{ij}^3\|$ без изменений.

Объяснение последней операции следующее. Если значение этого минимального элемента вычесть из всех остальных элементов матрицы $\|c_{ij}^3\|$, то на месте нулей будут стоять отрицательные величины и по крайней мере один элемент, не принадлежащий выделенному (вычеркнутому) множеству строк и столбцов, станет равным нулю. Однако теперь назначение нулевой стоимости может не быть оптимальным, поскольку матрица содержит отрицательные элементы. Чтобы матрица не содержала отрицательных элементов, прибавим абсолютное значение наименьшего отрицательного элемента ко всем элементам выделенных строк и столбцов. Отметим, что к элементам, расположенным на пересечении столбца 1 со строками 2 и 3, данная величина прибавляется дважды. Кроме того, все отрицательные элементы преобразуют в нулевые или положительные.

Новая редуцированная матрица стоимостей $\|c_{ij}^4\|$ содержит еще один нуль — элемент c_{13}^4 . Возвращаемся к шагу 3. Шаги 3 и 4 выполняют до тех пор, пока не будет получено оптимальное решение.

Шаг 5 (проведение назначений). Проведем назначение согласно $\|c_{ij}^5\| = \|c_{ij}^4\|$, соответствующее элементу c_{13}^4 , и зачеркнем нуль c_{11}^4 . Затем проведем назначения по элементам c_{22}^4 , c_{41}^4 , c_{34}^4 . Получили оптимальное решение, поскольку проведено полное назначение:

$$\bar{C} = \|c_{ij}^5\| = \begin{pmatrix} 0 & 6 & 0 & 3 \\ 13 & 0 & 5 & 4 \\ 4 & 3 & 0 & 0 \\ 0 & 9 & 2 & 13 \end{pmatrix}, \quad \bar{X} = \|x_{ij}^5\| = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Вычисляем минимальное значение целевой функции (c_{ij} берем из исходной матрицы $\|c_{ij}\|$ стоимости назначений):

$$f(x) = 9 + 4 + 11 + 4 = 28.$$

Замечание. Если в задаче о назначениях необходимо найти максимум критерия эффективности (целевой функции), то все элементы матрицы стоимостей следует умножить на -1 (тем самым перейти к минимизации исходной целевой функции) и прибавить к ним достаточно большое число M такое, чтобы вновь получившаяся матрица стоимостей не содержала отрицательных элементов (M достаточно выбрать равным по модулю наименьшему отрицательному элементу в матрице стоимостей). Теперь задачу можно решать как задачу минимизации.

§ 3.6. Задача размещения производства

Пусть намечается выпуск m видов продукции, которые могли бы производиться на n предприятиях, $n > m$. Следует из n предприятий выбрать такие m , каждое из которых будет производить один вид продукции. Издержки производства и сбыта единицы продукции, плановый объем годового производства продукции и плановая стоимость единицы продукции каждого вида известны.

Эту задачу путем введения фиктивных $n - m$ видов продукции с нулевыми стоимостями сводим к задаче о назначениях.

Издержки производства и сбыта заданы в табл. 3.3 и 3.4.

Таблица 3.3

Издержки производства на единицу продукции (усл. ед.)

Вид продукции	Предприятие				
	1	2	3	4	5
1	20	23	38	15	35
2	8	29	6	35	35
3	5	8	3	4	7

Таблица 3.4

Издержки сбыта на единицу продукции (усл. ед.)

Вид продукции	Предприятие				
	1	2	3	4	5
1	20	50	20	10	13
2	7	90	8	35	60
3	5	5	4	15	6

Плановый объем производства видов продукции составляет соответственно 35 000, 16 000, 54 000 ед.; плановая стоимость — соответственно 55, 50, 30 усл. ед.

Решение. Вычислим прибыль на единицу продукции (разность цены и суммарных издержек на производство и сбыт) (табл. 3.5).

Таблица 3.5

Прибыль на единицу продукции

Вид продукции	Предприятие				
	1	2	3	4	5
1	15	−18	−3	30	7
2	35	−69	36	−20	−45
3	20	17	23	11	17

Умножив прибыль, приходящуюся на единицу продукции, на годовой объем сбыта, получим общую годовую прибыль, соответствующую каждой паре: вид продукции — предприятие (табл. 3.6).

Таблица 3.6

Прибыль предприятий

Вид продукции	Предприятие				
	1	2	3	4	5
1	525	−630	−105	1050	245
2	5600	−11 040	5760	−3200	−7200
3	1080	918	1242	594	918

Чтобы свести исходную задачу к задаче о назначениях, введем два вида фиктивной продукции (4 и 5), которым соответствует нулевая прибыль. Поскольку это задача максимизации прибыли, то умножим все элементы матрицы на -1 , после чего наименьшим

отрицательным числом -5760 получившейся матрицы $\|a_{ij}\|$ станет элемент a_{23} . Затем ко всем элементам матрицы $\|a_{ij}\|$ (кроме нулевых) прибавим число 5760 . В результате получим матрицу стоимости назначений (табл. 3.7). Далее редуцируем матрицу. Модифицируем редуцированную матрицу, вычеркивая в ней третий и четвертый столбцы, четвертую и пятую строки. Получим оптимальное решение (табл. 3.8).

Таблица 3.7

Матрица стоимости

Вид продукции	Предприятие				
	1	2	3	4	5
1	5235	6390	5865	4710	5515
2	160	16 800	0	8960	12 960
3	4680	4842	4518	5166	4842
4	0	0	0	0	0
5	0	0	0	0	0

Таблица 3.8

Оптимальное решение

Вид продукции	Предприятие				
	1	2	3	4	5
1	365	1520	1155	[0]	645
2	[0]	16 640	0	8960	12 800
3	2	164	[0]	648	164
4	0	[0]	160	160	0
5	0	0	160	160	[0]

Отсюда следует, что производство первого вида продукции назначается 4-му предприятию, второго вида — 1-му, третьего вида — 3-му. Суммарная годовая прибыль, соответствующая данному решению, равна 7 892 000 усл. ед.

Модифицированный алгоритм решения задачи о назначениях может быть использован и при решении задачи коммивояжера (см. § 3.9).

§ 3.7. Задача о максимальном потоке

Пусть задана ориентированная сеть с одним источником s и одним стоком t , и пусть дуги (i, j) имеют ограниченную пропускную способность. Задача о максимальном потоке заключается в поиске таких потоков по дугам, что результирующий поток, протекающий из источника s в сток t , является максимальным. Задача о максимальном потоке может быть сформулирована как задача ЛП и поэтому может быть решена симплекс-методом.

Здесь для ее решения будет использован более эффективный метод расстановки пометок, разработанный Л. Р. Фордом и Д. Р. Фалкерсоном. Алгоритм начинает работу с некоторого допустимого решения, узлы рассматривают как промежуточные пункты передачи потока, а дуги — как распределительные каналы. Для описания алгоритма вводят два понятия: *пометки* и *увеличивающие (аугментальные) пути потока*. Пометку узла используют для указания как величины потока, так и источника потока, вызывающего изменение текущей величины потока по дуге, соединяющей этот источник с рассматриваемым узлом. Если q_i единиц потока посылают из узла i в узел j и это вызывает увеличение потока по данной дуге, то будем говорить, что узел j *помечен из узла i символом $(+q_j)$* ; в данном случае узлу j приписывают пометку $[+q_j; i]$. Если посылка q_j единиц потока вызывает уменьшение потока по дуге, то будем говорить, что узел j *помечен из узла i символом $(-q_j)$* . В таком случае узлу j приписывают пометку $[-q_j; i]$.

Текущий поток из узла i в узел j увеличивается, когда q_j единиц дополнительного потока послано в узел j по ориентированной дуге (i, j) в направлении, совпадающем с ее ориентацией. В таком случае дугу (i, j) называют *прямой*. Если q_i единиц дополнительного потока послано в узел j по ориентированной дуге (j, i) , т. е. в направлении, противоположном ее ориентации, то текущий поток из узла j в узел i уменьшается, а дугу (j, i) называют *обратной*. Узел j может быть помечен из узла i только после того, как узлу i приписана пометка. Если узел j помечен из узла i и дуга (i, j) *прямая*, то поток по данной дуге *увеличивается* и величина, соответствующая оставшейся неиспользованной пропускной способности дуги, должна быть нужным образом скор-

ректирована. Эту величину обычно называют *остаточной пропускной способностью*. Если некоторому узлу приписана пометка и при этом используется прямая ветвь, то она может иметь только остаточную пропускную способность.

Увеличивающий (аугментальный) путь потока из источника s в сток t определяют как связную последовательность прямых и обратных дуг, по которым из s в t можно послать несколько единиц потока. Поток по каждой прямой дуге увеличивается, не превышая при этом ее пропускной способности, а поток по каждой обратной дуге уменьшается, оставаясь при этом неотрицательным. Аугментальный путь потока используют для выбора такого способа изменения потока, при котором поток в узле t увеличивается и для каждого внутреннего узла сети не будет нарушено условие сохранения потока.

Как конкретно можно увеличить поток на q_i единиц? Предположим, что дуге (i, j) уже приписан поток $f_{ij} \geq 0$, $f_{ij} \leq u_{ij}$, где u_{ij} — пропускная способность дуги (i, j) . Величина q_i не может превосходить остаточной пропускной способности $u_{ij} - f_{ij}$. Однако из узла i не всегда можно получить $u_{ij} - f_{ij}$ единиц потока. Отметим, что в узел j можно послать столько единиц потока, сколько их добавлено в узел i , т. е. самое большое — это q_i . Следовательно, поток по прямой дуге (i, j) можно увеличить на величину $q_j = \min \{q_i; u_{ij} - f_{ij}\}$. Аналогично помечают узел j , если дуга (j, i) является обратной. Уменьшение потока по дуге (j, i) возможно только в том случае, когда $f_{ij} > 0$. Поток может быть уменьшен, самое большее, на число единиц потока, которые можно взять из узла i , т. е. на величину q_i . Следовательно, поток по обратной дуге (j, i) может быть уменьшен на величину $q_j = \min \{q_i; f_{ij}\}$.

В начале работы алгоритма источнику приписывают пометку $[\infty; -]$, указывающую на то, что из данного узла может вытекать поток бесконечно большой величины. Далее мы ищем аугментальный путь потока от источника к стоку, проходящий через помеченные узлы. Все узлы (за исключением источника) в начальный момент не помечены. Пытаясь достичь стока, мы проходим, выбирая дуги с наибольшими пропускными способностями, по прямым и обратным дугам и последовательно помечаем принадлежащие им узлы. Возможны два случая.

1. Стоку t приписывается пометка $[+q_t; k]$, k — номер узла, связанного со стоком. В этом случае аугментальный путь потока найден и поток по каждой дуге этого пути может быть увеличен или уменьшен на величину q_t . После изменения дуговых потоков текущие пометки далее не используются и всю описанную процедуру выполняют заново.

2. Сток t не может быть помечен. Это означает, что аугментальный путь потока не может быть найден. Следовательно, построенные дуговые потоки образуют оптимальное решение (максимальный поток).

Задача. Найти максимальное количество информации, которое может быть передано из источника s в сток t по сети, представленной на рис. 3.5. Пропускные способности u_{ij} дуг отмечены полужирным шрифтом на рис. 3.5.

Решение. Решим задачу методом Форда и Фалкерсона.

Итерация 1. Каждой дуге (i, j) приписывают пометку $[f_{ij}; u_{ij}]$, где f_{ij} — текущий дуговой поток, u_{ij} — пропускная способность дуги. Приписываем узлу s пометку $[\infty; -]$. Из источника s максимальный поток идет в узел 2; узлу 2 приписываем пометку $[+3; s]$, так как $q_2 = \min\{q_s, u_{s2}\} = \min\{\infty, 3\} = +3$. Из узла 2 максимальную пропускную способность имеет дуга $(2, t)$, ведущая в сток t . Приписываем стоку пометку: $[+2; 2]$, так как $q_t = \min\{q_s, u_{2t}\} = \min\{3, 2\} = 2$, т. е. изменение дуговых потоков возможно на 2 единицы: $f_{s2} = 2$, $f_{2t} = 2$. Дуги получают пометки, представленные на рис. 3.6. Суммарный поток F для этой итерации равен 2 ед.

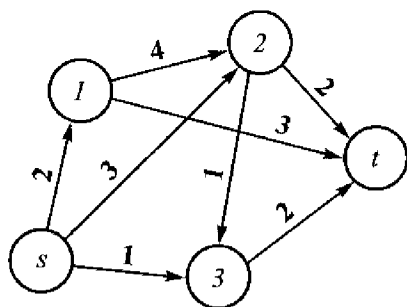


Рис. 3.5. Исходные данные для задачи о максимальном потоке (о максимальном количестве информации)

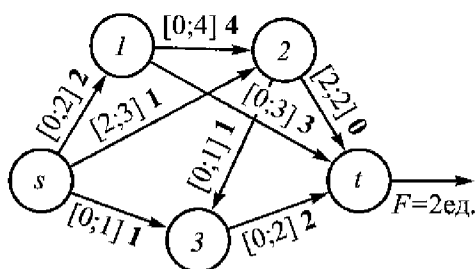


Рис. 3.6. Первая итерация

Итерация 2. Теперь просматриваем остаточные пропускные способности $u_{ij} - f_{ij}$ (на рис. 3.6 выделены жирным шрифтом). Источнику s приписываем пометку $[\infty; -]$. Из источника s предпочтительнее двигаться в узел 1; узлу 1 приписываем пометку $[+2; s]$. Из узла 1 идем в узел 2; узлу 2 приписываем пометку $[+2; 1]$, так как $q_2 = \min\{q_1; u_{12} - f_{12}\} = 2$. Из узла 2 идем в узел 3. Для узла 3 $q_3 = \min\{q_2; u_{23} - f_{23}\} = \min\{2; 1\} = 1$, узлу 3 приписываем пометку $[+1; 2]$. Из узла 3 идем в сток. Стоку приписываем пометку $[+1; 3]$. Минимальное изменение дуговых потоков равно 1, т. е. $f_{s1} = 1$, $f_{12} = 1$, $f_{23} = 1$, $f_{3t} = 1$. Максимальный поток F равен $2 + 1 = 3$ (ед.). Дуги получают пометки, представленные на рис. 3.7.

Итерация 3. Приписываем узлу s пометку $[\infty; -]$. Из узла s движемся в узел 1: $q_1 = \min\{q_s; u_{s1} - f_{s1}\} = \min\{\infty; 1\} = +1$. Узлу 1 приписываем пометку $[+1; s]$. Из узла 1 идем в сток t : $q_t = \min\{q_1; u_{1t} - f_{1t}\} = \min\{1; 3\} = 1$. Стоку t приписываем пометку $[+1; 1]$. Изменение дуговых потоков для f_{s1} равно 1; для f_{1t} равно 1, изменятся только значения потоков на дугах $(s, 1)$ и $(1, t)$. Полные потоки станут равными: $f_{s1} = 2$, $f_{1t} = 1$. Дуги получают пометки, представленные на рис. 3.8. Максимальный поток F станет равным $3 + 1 = 4$ (ед.).

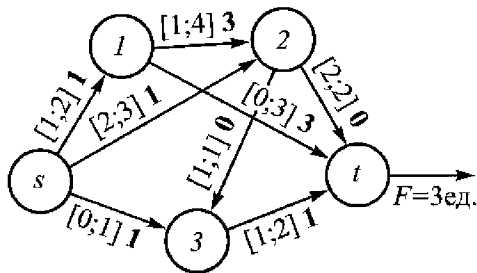


Рис. 3.7. Вторая итерация

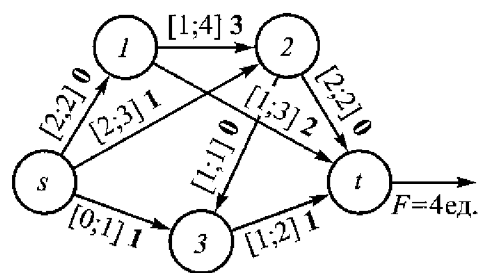


Рис. 3.8. Третья итерация

Итерация 4. Приписываем узлу s пометку $[\infty; -]$. Из узла s направляемся в узел 2: $q_2 = \min\{q_s; u_{s2} - f_{s2}\} = \min\{\infty; 1\} = +1$; узлу 2 приписываем пометку $[+1; s]$. Из узла 2 мы можем попасть в сток t , только пройдя по обратной дуге в узел 1, а затем в сток t . Определим пометку узла 1 в данном случае: $q_1 = \min\{q_2; f_{12}\} = \min\{+1; 4\} = +1$, т. е. узлу 1 приписываем пометку $[-1; 2]$. Для узла t : $q_t = \min\{q_1; u_{1t} - f_{1t}\} = \min\{+1; 1\} = +1$; узлу t приписываем пометку $[+1; 1]$. Изменение дуговых потоков для f_{s2} равно 1, для

f_{12} равно 1, для f_{1t} равно 1. Поток по этим дугам будут соответственно $f_{s2} = 2$, $f_{12} = 0$, $f_{1t} = 2$; максимальный поток F из s в t равен $4 + 1 = 5$ (ед.). Дуги получают пометки (изменяют пометки только на дугах $(s, 2)$, $(1, 2)$ и $(1, t)$), представленные на рис. 3.9.

Итерация 5. Приписываем источнику s пометку $[\infty; -]$. Из узла s мы можем попасть только в узел 3. Для узла 3 имеем $q_3 = \min\{q_s; u_{s3} - f_{s3}\} = \min\{\infty; +1\} = +1$; узел 3 получает пометку $[+1; s]$. Из узла 3 попадаем в сток t : $q_t = \min\{q_3; u_{3t} - f_{3t}\} = \min\{1; 1\} = +1$. Сток t получает пометку $[+1; 3]$. Изменение дуговых потоков для f_{s3} и f_{3t} равно 1; соответственно получим $f_{s3} = 1$, $f_{3t} = 2$ (на второй итерации $f_{3t} = 1$). Максимальный поток F равен $5 + 1 = 6$ (ед.). Дуги получают пометки, представленные на рис. 3.10.

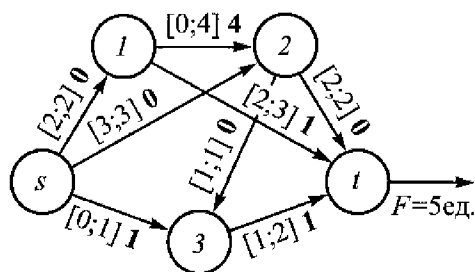


Рис. 3.9. Четвертая итерация

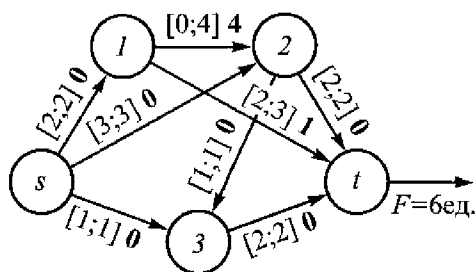


Рис. 3.10. Оптимальное решение

Итерация 6. Приписываем узлу s пометку $[\infty; -]$. Аугментальный поток не может быть найден, т. е. построенные дуговые потоки образуют оптимальное решение. Максимальный поток (максимальное количество информации) из источника s в сток t равен 6 ед.: 2 ед. потока идут из источника s через узел 1 в сток t ; 1 ед. потока идет из источника s через узел 3 в сток t ; 3 ед. потока идут из источника s в узел 2, затем 2 ед. потока поступают из узла 2 в сток t , а 1 ед. потока поступает в сток t через узел 3. Через дугу $(1, 2)$ информация не идет.

В приложении П4 приводится описание компьютерной программы (инструкция пользователю) для решения задачи о максимальном потоке.

§ 3.8. Задача о многополюсном максимальном потоке

Мы рассмотрели задачу о максимальном потоке из единственного узла s в единственный узел t . В целом ряде технических и экономических приложений возникают задачи, в которых надо определить максимальный поток между любыми выбранными парами узлов в сети. Примерами таких систем являются: сети автомобильных дорог, где автостреды изображаются дугами, пропускные способности которых соответствуют максимально допустимой интенсивности движения; телефонная или информационная сеть, где линии представляются дугами, а их пропускные способности соответствуют максимальному числу вызовов (или объему информации), которые могут обслуживаться в каждый момент времени; электрические и электроэнергетические системы, где линии электропередачи представлены дугами, а пропускные способности соответствуют максимальному объему электроэнергии, который может передаваться по линиям в единицу времени, и т. п.

Задачи о многополюсных максимальных потоках бывают двух видов: анализа и синтеза сети.

1. *Задача анализа сети.* Задана сеть с ограниченными пропускными способностями дуг. Следует определить, каковы величины максимальных потоков, которые можно пропустить между всеми парами узлов в заданной сети.

2. *Задача синтеза сети.* Требуется построить сеть, в которой величины максимальных потоков f_{ij} между всеми парами узлов удовлетворяют заданным ограничениям снизу и в которой общая пропускная способность всех дуг максимальна.

Задачу о многополюсном максимальном потоке можно решить, опираясь на рассмотренную задачу о максимальном потоке между единственным источником и единственным стоком. Если пропускная способность каждой дуги не зависит от направления движения потока по этой дуге и если каждую пару узлов можно рассматривать как пару источник — сток, то общее число задач о максимальном потоке, которое должно быть решено, равно $n(n-1)/2$, где n — число узлов в сети. Рассмотрим здесь алгоритм Гомори — Ху, согласно которому максимальный поток в задаче о многополюсном максимальном потоке определяют только $n-1$ раз. Основная идея алгоритма состоит в итеративном построении

максимального остова дерева, ветви которого соответствуют разрезам, а параметры ветвей — величинам разрезов.

Пусть $G = (N, A)$ — неориентированная сеть, где N — множество узлов, A — множество дуг, и пусть c_{ij} — пропускная способность дуги (i, j) из множества A , причем $c_{ij} = c_{ji}$. Максимальный поток между узлами i и j равен v_{ij} ; $(X, \bar{X})_{ij}$ — минимальный разрез, отделяющий узел i от узла j ($i \in X$; $j \in \bar{X}$); $c(X, \bar{X})_{ij}$ — пропускная способность минимального разреза, отделяющего узел i от узла j . Согласно теореме о максимальном потоке и минимальном разрезе $v_{ij} = c(X, \bar{X})_{ij}$. Если некоторый узел k принадлежит \bar{X} , то $v_{ik} \leq c(X, \bar{X})_{ij}$, а если $k \in X$, то $v_{kj} \leq c(X, \bar{X})_{ij}$. Следовательно, $v_{ij} \geq v_{ik}$ и $v_{ij} \geq v_{kj}$. Поэтому $v_{ij} \geq \min\{v_{ik}; v_{kj}\}$. Если рассмотреть в общем случае связанное подмножество узлов $\{t, p, k, q, j\}$, то

$$v_{ij} \geq \min\{v_{ip}; v_{pk}; v_{kq}; v_{qj}\}.$$

А для максимального остовного дерева

$$v_{ij} \leq \min\{v_{ip}; v_{pk}; v_{kq}; v_{qj}\},$$

где (i, j) — произвольная дуга, не принадлежащая данному дереву. Если это не так, то вместо любой дуги пути из узла i в узел j можно взять дугу (i, j) , в результате чего будет построено дерево с большим весом. Отсюда для любой дуги, не принадлежащей дереву, имеем

$$v_{ij} = \min\{v_{ip}; v_{pk}; v_{kq}; v_{qj}\}.$$

Максимальное остовное дерево, удовлетворяющее последнему равенству, называют *деревом разрезов* потому, что каждая его ветвь соответствует разрезу, а вес ветви равен пропускной способности разреза. Если требуется определить величину максимального потока между двумя противоположными узлами, необходимо в дереве найти путь, соединяющий эти два узла, и выбрать на этом пути дугу с минимальным весом. Ее вес равен величине максимального потока между рассматриваемыми узлами.

Если некоторый узел рассматривать как источник s , а другой узел как сток t , то максимальным потоком между ними является $v_{st} = c(X, \bar{X})_{st}$. Если затем в качестве источника и стока выбирают другую пару узлов i, j , удовлетворяющих такому условию, что они оба принадлежат подмножеству X (или \bar{X}), то подмножество уз-

лов \bar{X} (или X , если узлы i, j принадлежат \bar{X}) может быть объединено в один *конденсированный узел*. При этом величина максимального потока из узла i в узел j будет одной и той же для исходной и конденсированной сетей и в алгоритме Гомори — Ху при определении величины v_{ij} используют решение задачи о максимальном потоке, найденное на предыдущем шаге.

Пусть \bar{N}_{ij} — множество узлов, образуемое в результате конденсации всех узлов, лежащих по ту сторону разреза, где не содержатся узлы i и j ; \bar{A}_{ij} — множество дуг, соединяющих узлы из множества \bar{N}_{ij} . Если известны пропускные способности дуг, принадлежащих \bar{A}_{ij} , то для нахождения величины максимального потока между узлами i, j можно воспользоваться процедурой *расстановки пометок*.

В свою очередь, пропускные способности дуг из \bar{A}_{ij} определяют следующим образом. Пусть j_1, j_2, \dots, j_r — узлы из подмножества \bar{X} , непосредственно связанные с узлом $i \in X$. При конденсации подмножества \bar{X} дуги $(i, j_1), (i, j_2), \dots, (i, j_r)$ заменяют одной дугой, соединяющей узел i и конденсированный узел подмножества \bar{X} . Пропускная способность такой дуги вычисляется по формуле

$$c_{i\bar{X}} = c_{ij_1} + c_{ij_2} + \dots + c_{ij_r} = \sum_{m=1}^r c_{ij_m}.$$

Для определения величины v_{ij} вновь надо найти минимальный разрез $(X, \bar{X})_{ij}$ с минимальной пропускной способностью, отделяющий узел i от узла j . Теперь можно выбрать другую пару узлов, принадлежащих либо X , либо \bar{X} , и построить конденсированную сеть. В результате выполнения процедуры расстановки пометок можно будет определить другой разрез и построить новую конденсированную сеть и т. д.

Задача. В сети (рис. 3.11) для каждой пары узлов определить величину максимального потока информации между ними.

Решение. Для простоты сначала найдем максимальные потоки между узлами N_1, N_3, N_4 и N_5 .

Итерация 1. Выберем два произвольных узла, скажем N_1 и N_3 , и найдем максимальный поток между ними (путем процедуры расстановки пометок). Получим минимальный разрез $(N_1, N_2, N_6 \mid N_3, N_4, N_5)$. Пропускная способность дуги, соединяющей

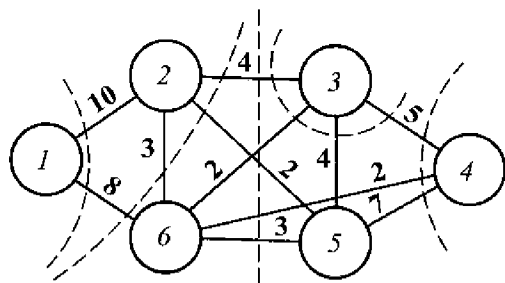


Рис. 3.11. Исходные данные для задачи о многополюсном максимальном потоке информации

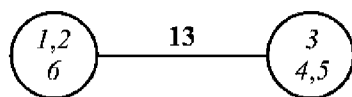


Рис. 3.12. Начало построения дерева разрезов

конденсированные узлы, равна $4 + 2 + 2 + 2 + 3 = 13$. Начало построения дерева разрезов представлено на рис. 3.12.

Итерация 2. Найдем максимальный поток между узлами N_3 и N_1 (рис. 3.13). Здесь узлы N_1 , N_2 и N_6 объединены в конденсированный узел. Узел N_3 соединен с узлами N_2 и N_6 , т. е. пропускная способность дуги от конденсированного узла к узлу N_3 будет равна $4 + 2 = 6$. Узел N_5 соединен с узлами N_2 и N_6 , т. е. пропускная способность дуги, соединяющей конденсированный узел с узлом N_5 , равна $2 + 3 = 5$. Величина максимального потока равна $5 + 2 + 7 = 14$, минимальный разрез имеет вид $(N_1, N_2, N_6, N_3, N_5 \mid N_4)$. По минимальному разрезу определим, что узлы (N_3, N_5) и (N_1, N_2, N_6) лежат по одну сторону в минимальном разрезе, разделяющем узлы N_3 и N_4 . Получим дерево, представленное на рис. 3.14.

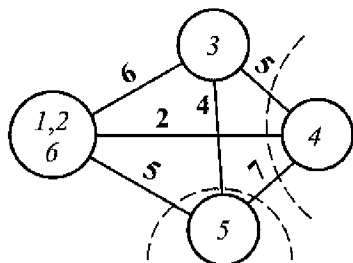


Рис. 3.13. К определению максимального потока между узлами N_3 и N_1

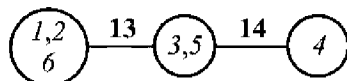


Рис. 3.14. Построение дерева разрезов

Итерация 3. Найдем максимальный поток в той же сети (см. рис. 3.13) между узлами N_3 и N_5 . Получим минимальный разрез

$(N_1, N_2, N_6, N_5, N_4 \mid N_3)$. Величина максимального потока равна $5 + 4 + 7 = 16$. Символическое изображение сети представлено на рис. 3.15. Теперь каждая интересующая вершина в полученном дереве (см. рис. 3.15) содержит только по одному полюсу и максимальные потоки будут $f_{13} = f_{14} = f_{15} = 13$ (они равны минимальной пропускной способности дуги на данном пути), $f_{34} = f_{45} = 13$, $f_{35} = 16$. Эти потоки равны соответствующим максимальным потокам в исходной сети (см. рис. 3.15).

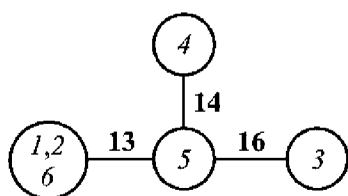


Рис. 3.15. Символическое изображение сети с помощью дерева разрезів

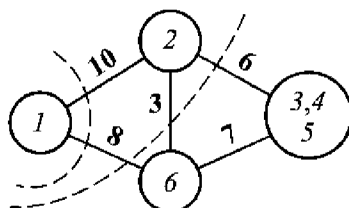


Рис. 3.16. Нахождение максимального потока между узлами N_1 и N_6

Если необходимо найти величины максимальных потоков между всеми парами узлов, то следует процесс продолжить.

Итерация 4. Найдем максимальный поток между узлами N_1 и N_6 в сети, изображенной на рис. 3.16. Минимальный разрез $(N_1, N_2 \mid N_6, N_3, N_4, N_5)$, имеющий пропускную способность $6 + 3 + 8 = 17$, изображен на рис. 3.17.

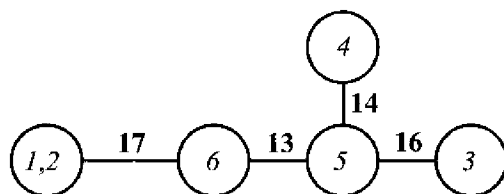


Рис. 3.17. Построение дерева разрезів

Итерация 5. Найдем максимальный поток между узлами N_1 и N_2 . Минимальный разрез $(N_1 \mid N_2, N_3, N_4, N_5, N_6)$, имеющий пропускную способность $10 + 8 = 18$, приведен на рис. 3.18. С помощью дерева, представленного на рис. 3.18, можно определить величины максимальных потоков между всеми парами узлов (табл. 3.9).

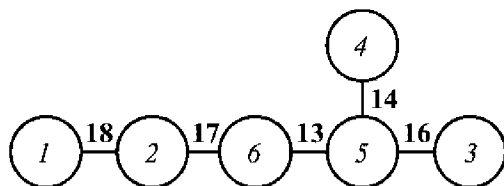


Рис. 3.18. Дерево разрез

Таблица 3.9

Величины максимальных потоков между узлами сети

Узел	1	2	3	4	5	6
1	∞	18	13	13	13	17
2	18	∞	13	13	13	17
3	13	13	∞	14	16	13
4	13	13	14	∞	14	13
5	13	13	16	14	∞	13
6	17	17	13	13	13	∞

Если в двух сетях равны величины максимальных потоков между парами узлов, принадлежащих некоторому заданному множеству, то эти сети называют *потоко-эквивалентными по отношению к заданному множеству узлов*. Заметим, что существует много деревьев, которые потоко-эквивалентны некоторой заданной сети. Потоко-эквивалентное дерево (см. рис. 3.18) обладает следующей особенностью: каждая ветвь этого дерева соответствует некоторому минимальному разрезу в исходной сети. Поэтому его называют *деревом разрез*. На дереве разрез для сети, содержащей n узлов, показано $n - 1$ минимальных разрезов исходной сети, не пересекающихся друг с другом (см. рис. 3.11).

§ 3.9. Методы ветвей и границ. Задача коммивояжера

Методы ветвей и границ предназначены для решения широкого круга дискретных оптимизационных задач. Различные методы ветвей и границ существенно используют специфику конкретных задач и поэтому заметно отличаются друг от друга, но все они основаны на последовательном разбиении допустимого множества решений D на подмножества (ветвления) и вычислении оценок (границ), позволяющих не рассматривать подмножества, заведомо не содержащие решения задачи.

Пусть требуется найти точку минимума функции $f(x)$ при $x \in D$. В зависимости от специфики задачи выбирается способ вычисления оценок снизу $d(D')$ функции $f(x)$ на подмножествах $D' \subset D$ (может быть, что $D' = D$):

$$f(x) \geq d(D'), \quad x \in D'.$$

Оценка снизу часто вычисляется путем релаксации, т. е. замены задачи минимизации функции $f(x)$ на множестве D' задачей минимизации по некоторому более расширенному множеству. Например, в целочисленных задачах не учитывают требование целочисленности.

Выбирается также правило ветвления, состоящее в выборе разветвляемого подмножества D' из числа подмножеств, на которые к данному шагу разбито множество D , и выборе способа разбиения D' на непересекающиеся подмножества: ветвлению подвергается подзадача минимизации функции $f(x)$ на подмножестве D' . Обычно для ветвления выбирается подмножество D' с наименьшей оценкой значения целевой функции, поскольку в таком множестве естественно искать минимум в первую очередь. При этом рассматриваются только такие способы вычисления оценок снизу, в которых оценки для подмножеств, получившихся в результате разветвления подмножества D' , не меньше $d(D')$.

При решении релаксированной задачи может оказаться, что:

а) допустимое множество этой задачи пусто и, стало быть, $D' = \emptyset$;

б) значение минимума $d(D')$ для релаксированной задачи больше или равно наименьшему из уже вычисленных значений функции $f(x)$ (текущему значению рекорда), и потому $\min_{x \in D} f(x)$ достигается вне множества D' ;

в) точка минимума для релаксированной задачи принадлежит множеству D' и, стало быть, является точкой минимума $f(x)$ на D' .

Во всех трех случаях подмножество D' далее не разветвляется. В случае а) текущее значение рекорда полагается равным минимуму из предыдущего текущего значения и вычисленного значения $\min_{x \in D'} f(x)$.

В методе ветвей и границ на каждом шаге искомое значение не больше текущего значения рекорда (верхней границы) и не мень-

ше наименьшей из оценок снизу для подзадач, входящих на данном шаге в число кандидатов на ветвление (нижней границы).

Существуют варианты метода ветвей и границ, разработанные специально для нахождения приближенного решения различных задач.

Пример. Рассмотрим решение целочисленной задачи линейного программирования:

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

при

$$\sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, 2, \dots, m, \quad x_j \geq 0, \quad j = 1, 2, \dots, n,$$

$$x_j \in \{0, 1, 2, \dots\}, \quad j \in J \subset \{1, 2, \dots, n\}.$$

Допустимое множество D задачи предполагается ограниченным, оценки снизу вычисляются с помощью релаксации — без условия целочисленности переменных. Оценку снизу получают с помощью симплекс-метода.

Если после применения симплекс-метода решение не является целочисленным, то на первом шаге алгоритма выбирается любая нецелочисленная компонента $x_{r_1}^1$, $r_1 \in J$, полученного решения и исходная задача разветвляется на две подзадачи: первая — с дополнительным ограничением $x_{r_1} \leq [x_{r_1}^1]$, вторая — с дополнительным ограничением $x_{r_1} \geq [x_{r_1}^1] + 1$, где $[\cdot]$ — целая часть числа. Вычисляются оценки снизу, и если обе подзадачи остаются в числе кандидатов на дальнейшее ветвление, то для ветвления на втором шаге выбирается подзадача с минимальной оценкой.

На k -м шаге выбранная на $(k-1)$ -м шаге подзадача разветвляется на две новые с дополнительными ограничениями $x_{r_k} \leq [x_{r_k}^k]$ и $[x_{r_k} \geq [x_{r_k}^k] + 1]$ соответственно, где x_{r_k} , $r_k \in J$, — любая нецелочисленная компонента решения x^k задачи ЛП, получающейся релаксацией подзадачи, выбранной на $(k-1)$ -м шаге. Для новых подзадач вычисляются оценки снизу. Формируется список кандидатов на ветвление. Для ветвления на $(k+1)$ -м шаге из числа кандидатов на ветвление выбирается подзадача с минимальной оценкой.

Конечность алгоритма следует из ограниченности множества D .

Процесс ветвления существенно упрощается, если в задаче имеются ограничения

$$x_j \in \{0, 1\}, \quad j \in J \subset \{1, 2, \dots, n\}.$$

Задача коммивояжера. В гл. 1 мы упоминали задачу о Кёнигсбергских мостах, в которой требовалось обойти все мосты, не проходя ни один из них дважды, т. е. пройти все мосты по кратчайшему пути.

Можно привести и другую постановку этой задачи в более общем виде. Некто решил, что настало время посмотреть жизнь в других странах. Он выбрал семь мест, которые намеревался обязательно посетить, получил информацию о стоимости билета на самолет в каждый из выбранных городов и стоимости проезда из одного города в другой. Учел все льготы, предоставляемые авиакомпаниями, и составил матрицу стоимостей (табл. 3.10) проезда в выбранные города и обратно. Зная матрицу стоимостей, путешественнику надо так составить маршрут, чтобы затраты на это путешествие были минимальными и чтобы выполнялось естественное требование: каждый пункт посещается только один раз. Матрица стоимостей проезда из-за льгот авиакомпаний оказалась несимметричной. При симметричной матрице стоимостей процедура решения задачи не изменяется.

Таблица 3.10

Матрица стоимости проезда (усл. ед.)

№	Пункт	Исходный	Токио	Гонконг	Лондон	Сидней	Рим
		1	2	3	4	5	6
1	Исходный	—	270	430	160	300	260
2	Токио	70	—	160	10	300	300
3	Гонконг	200	130	—	350	50	0
4	Лондон	210	160	250	—	180	180
5	Сидней	120	460	270	480	—	50
6	Рим	230	50	50	90	50	—

Примечание. В процессе проведения расчетов все стоимости будут уменьшены в 10 раз.

Подобные задачи в математическом программировании называются *задачей коммивояжера*. Ее формулируют следующим образом. Коммивояжер должен выехать из исходного пункта и побывать

в каждом из остальных $n - 1$ пунктов ровно один раз и вернуться в исходный пункт. Задача заключается в определении последовательности посещаемых пунктов, при которой коммивояжеру требуется минимизировать некоторый критерий эффективности: стоимость проезда, время в пути, суммарное расстояние и т. д. Здесь требуется выбрать один или несколько оптимальных маршрутов из $(n - 1)!$ возможных. Если некоторые города для коммивояжера недоступны, то минимальное значение целевой функции должно быть бесконечно большим.

Рассмотрим решение задачи коммивояжера методом ветвей и границ. Вначале определяют некоторое *допустимое решение* (*допустимый маршрут*). После этого множество всех оставшихся маршрутов разбивают на все более мелкие подмножества и при каждом разбиении вычисляют нижнюю границу целевой функции текущего наилучшего (в смысле наименьшей стоимости проезда) маршрута. С помощью найденных границ проводят дальнейшее разбиение подмножеств допустимых маршрутов и в конечном итоге определяют оптимальный маршрут. Это разбиение подмножеств маршрутов можно рассматривать как узлы дерева. Поэтому данный метод называют *методом поиска по дереву решений*, или *методом ветвей и границ*.

Матрица стоимостей содержит неотрицательные элементы c_{ij} . Маршрут T можно представить как множество упорядоченных пар пунктов $T = \{(i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n), (i_n, i_1)\}$. Каждый допустимый маршрут представляет собой цикл, проходя по которому коммивояжер посещает каждый пункт ровно один раз и потом возвращается в исходную точку. Каждая упорядоченная пара (i, j) является *дугой*, или *звеном маршрута*. Стоимость маршрута T равна сумме элементов матрицы стоимостей, но только тех, которые соответствуют пунктам, лежащим на маршруте T :

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

Величина $z(T)$ определена для любого допустимого маршрута и не может быть меньше длины оптимального маршрута, т. е. текущее значение $z(T)$ является *верхней границей* $z_{\text{в}}(T)$ стоимости оптимального маршрута T .

Для вычисления нижних границ стоимости маршрута используют операции *редукции строк и столбцов* матрицы стоимости.

Процедуру вычитания из каждого элемента строки наименьшего элемента этой же строки и из каждого элемента столбца наименьшего элемента этого же столбца называют соответственно *редукцией строк* и *редукцией столбцов*. Матрицу с неотрицательными элементами, в каждой строке и каждом столбце которой содержится по крайней мере один нулевой элемент, называют *редуцированной*. Она может быть получена в результате последовательной редукции ее строк и столбцов. Если вычесть из каждого элемента некоторой строки матрицы стоимости постоянную величину c , то стоимость любого маршрута, определяемая новой матрицей, меньше стоимости того же маршрута, определяемого старой матрицей, на величину c , так как для любого допустимого маршрута каждая строка и каждый столбец матрицы стоимости содержит по одному элементу, соответствующему этому маршруту. При редукции относительные стоимости всех маршрутов останутся неизменными. Следовательно, останутся неизменными и все оптимальные маршруты.

Если $z(T)$ — стоимость маршрута T , определяемая матрицей стоимости до выполнения редукции, $z_1(T)$ — стоимость того же маршрута, определяемая редуцированной матрицей, H — сумма всех констант, используемых при вычислении редуцированной матрицы, то $z(T) = z_1(T) + H$. Поскольку редуцированная матрица содержит только неотрицательные элементы, то H является *нижней границей* стоимости маршрута T для нередуцированной матрицы стоимости.

В алгоритме метода ветвей и границ диагональные элементы исходной матрицы стоимости полагают равными ∞ , т. е. $c_{ii} = \infty$.

1. Выберем произвольный допустимый маршрут, состоящий, например, из звеньев $(1, 4)$, $(4, 5)$, $(5, 3)$, $(3, 6)$, $(6, 2)$, $(2, 1)$. Вычислим стоимость данного маршрута:

$$z_b(T) = 16 + 18 + 27 + 0 + 5 + 7 = 73,$$

т. е. для оптимального маршрута стоимость поездки коммивояжера не может превосходить значения $z_b(T) = 73$.

2. Выполним редукцию строк; для этого в каждой строке i определим минимальный элемент и найденное значение c_i вычтем из элементов соответствующей строки. Результаты запишем в виде табл. 3.11.

Таблица 3.11

Редукция строк

Узел	1	2	3	4	5	6	c_i
1	∞	11	27	0	14	10	16
2	6	∞	15	0	29	29	1
3	20	13	∞	35	5	0	0
4	5	0	9	∞	2	2	16
5	7	41	22	43	∞	0	5
6	18	0	0	4	0	∞	5

3. В полученной табл. 3.11 проведем редукцию столбцов. Из табл. 3.11 видно, что проводить следует только редукцию первого столбца, так как остальные столбцы содержат нулевые элементы. Редукция столбцов представлена в табл. 3.12.

Таблица 3.12

Редукция столбцов

Узел	1	2	3	4	5	6	c_i
1	∞	11	27	0	14	10	16
2	1	∞	15	0	29	29	1
3	15	13	∞	35	5	0	0
4	0	0	9	∞	2	2	16
5	2	41	22	43	∞	0	5
6	13	0	0	4	0	∞	5
Q_j	5	0	0	0	0	0	$H = 48$

Строка Q_j содержит вычитаемые константы для каждого столбца при редукции столбцов. Значение нижней границы для всех маршрутов в рассматриваемой задаче равно H — сумме всех вычитаемых констант:

$$H = \sum_{i=1}^6 c_i + \sum_{j=1}^6 Q_j = 48.$$

4. Теперь следует выбрать оптимальный маршрут. Если бы в каждой строке и каждом столбце было ровно по одному нулевому элементу, то эти элементы образовали бы оптимальный маршрут и оптимальная стоимость проезда была равна H .

Однако нулевые элементы не единственны в строках и столбцах. Вместо того чтобы одновременно определять все звенья оптимального маршрута с помощью текущей матрицы стоимости, воспользуемся алгоритмом, на каждом шаге которого по матрице стоимости строится одно звено оптимального маршрута. Очевидно сначала выбрать звено нулевой длины, а затем последовательно добавлять звенья нулевой или минимальной длины. Если выбрать звено (i, j) , то решение не должно содержать других звеньев, соответствующих элементам i -й строки и j -го столбца; если звено (i, j) можно исключить из окончательного решения, то его можно не рассматривать при выполнении последующих операций. Следовательно, для каждого звена достаточно рассмотреть следующие два случая: в первом случае звено включают в текущее и все последующие решения до определения оптимального решения; во втором — звено исключают из дальнейшего рассмотрения. В нашем примере мы уже получили начальный узел дерева ветвления, соответствующий множеству всех маршрутов с нижней границей стоимости всех маршрутов, равной $H = 48$, и верхней, равной $z_v(T) = 73$, т. е. диапазон (отрезок) стоимости всех маршрутов $z(T)$ равен $[48, 73]$.

5. Следующим шагом процедуры является выбор звена, на котором будет базироваться ветвление. Поскольку в каждой строке и в каждом столбце несколько элементов c_{ij} равны нулю, то надо рассмотреть маршруты, не содержащие звено (i, j) . Пункт должен быть связан с некоторым другим пунктом, и поэтому каждый маршрут, не содержащий звено (i, j) , должен содержать звено A , стоимость которого не меньше минимального элемента i -й строки, кроме $c_{ij} = 0$. Стоимость звена A обозначим A_i . Таким образом, чтобы A_i было равно нулю, в строке должно быть не менее двух нулевых элементов. Аналогично, чтобы в пункт j можно было попасть из некоторого другого города, маршрут, не содержащий звено (i, j) , должен содержать звено B , у которого стоимость не меньше минимального элемента j -го столбца, кроме $c_{ij} = 0$. Обозначим стоимость проезда по звену B через B_j , а сумму величин A_i и B_j через Φ_{ij} . Величину Φ_{ij} называют *вторичным штрафом*, и она равна минимальному штрафу, которому мы подвергаемся, если не включаем звено (i, j) в оптимальный маршрут. Если штраф за неиспользование звена вычислить для всех звеньев, у которых $c_{ij} = 0$, то можно сравнить соответствующие значения Φ_{ij} и

включить в текущий маршрут звено (i, j) , за неиспользование которого мы заплатили бы максимальный штраф, т. е., включая звено (i, j) , мы получаем выигрыш в стоимости, равный максимальному значению Φ_{ij} . Нижняя граница для соответствующей ветви должна быть выбрана таким образом, чтобы она не превосходила длины ни одного из маршрутов, не содержащих звено (i, j) . Данное требование будет выполнено, если значение новой нижней границы положить равным сумме значений текущей нижней границы и максимального штрафа за неиспользование звена (i, j) . Для определения максимального значения Φ_{ij} будем исследовать все элементы $c_{ij} = 0$; при $c_{ij} \neq 0$ величина $\Phi_{ij} = 0$. Данное утверждение справедливо в силу того, что если положить $c_{ij} = \infty$, а затем провести редукцию i -й строки и j -го столбца, то сумма вычитаемых констант будет равна Φ_{ij} . Для рассматриваемого случая значения A_i и B_j приведены в табл. 3.13, а значения Φ_{ij} (вторичный штраф) для узлов, соответствующих $c_{ij} = 0$, — в табл. 3.14. Максимальное значение $\Phi_{ij} = 10$ соответствует звену $(1, 4)$. Следовательно, в качестве базового звена ветвления выбираем звено $(1, 4)$.

Таблица 3.13

Значения A_i и B_j

Узел	1	2	3	4	5	6	c_i	A_i
1	∞	11	27	0	14	10	16	10
2	1	∞	15	0	29	29	1	1
3	15	13	∞	35	5	0	0	5
4	0	0	9	∞	2	2	16	0
5	2	41	22	43	∞	0	5	2
6	13	0	0	4	0	∞	5	0
Q_i	5	0	0	0	0	0	$H = 48$	
B_j	1	0	9	0	2	0		

Таблица 3.14

Значения Φ_{ij}

Звено (i, j)	(1, 4)	(2, 4)	(3, 6)	(4, 1)	(4, 2)	(5, 6)	(6, 2)	(6, 3)	(6, 5)
$\Phi_{ij} = A_i + B_j$	10	1	5	1	0	2	0	9	2

6. Нижняя граница для маршрутов, не включающих звено $(1, 4)$, вычисляется по формуле $H + \Phi_{14} = 48 + 10 = 58$. Чтобы

определить новую нижнюю границу для маршрутов, включающих звено $(1, 4)$, необходимо преобразовать матрицу стоимости. Если мы включили в маршрут некоторое звено (k, l) , то в дальнейшем мы не рассматриваем k -ю строку и l -й столбец. Кроме того, звено (k, l) является тогда звеном некоторого ориентированного цикла и не может принадлежать этому же маршруту. Последнее условие можно выполнить, положив $c_{lk} = \infty$.

Из рассмотрения следует исключить и так называемые *запрещенные звенья* — звенья, с помощью которых в дальнейшем могут быть образованы циклы, включающие в себя неполное множество пунктов (могут быть образованы подмаршруты). Элементы матрицы стоимости, соответствующие этим звеньям, полагают равными ∞ . Преобразованная матрица стоимости представлена в табл. 3.15.

Таблица 3.15

Преобразованная матрица стоимости

Узел	1	2	3	5	6
2	1	∞	15	29	29
3	15	13	∞	5	0
4	∞	0	9	2	2
5	2	41	22	∞	0
6	13	0	0	0	∞

Таблица 3.16

Вторая матрица решений

Узел	1	2	3	5	6	c_i	A_i
2	0	∞	14	28	28	1	14
3	15	13	∞	5	0	0	5
4	∞	0	9	2	2	0	2
5	2	41	22	∞	0	0	2
6	13	0	0	0	∞	0	0
Q_i	0	0	0	0	0	$H_1 = 1$	—
B_i	2	0	9	2	0		—

Запрещенных звеньев в данном случае не существует. Матрица решений после редукции строк и столбцов, а также с указанием значений c_i , A_i и B_j для нее приведена в табл. 3.16. Нижняя граница для маршрута, включающего звено $(1, 4)$, может быть вы-

числена как сумма всех новых вычитаемых констант H_1 и старой нижней границы, т. е. нижняя граница равна $48 + 1 = 49$.

Дерево решений для рассмотренных двух этапов имеет вид, представленный на рис. 3.19, где показаны маршруты, включающие звено $(1, 4)$ и не включающие звено $(1, 4)$.

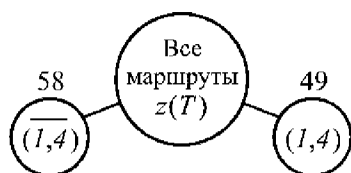


Рис. 3.19. Дерево решений для первых двух этапов

В узлах дерева указаны нижние границы для каждого варианта маршрута, $48 \leq z(T) \leq 73$.

7. С вычисления второй матрицы решений начинается вторая итерация решения. После второй редукции имеем $\Phi_{21} = 16$, $\Phi_{36} = 5$, $\Phi_{42} = 2$, $\Phi_{56} = 2$, $\Phi_{62} = 0$, $\Phi_{63} = 9$ и $\Phi_{65} = 2$. Макси-

мальным среди них является значение $\Phi_{21} = 16$, т. е. выбирают звено $(2, 1)$. Новая нижняя граница для маршрута, не включающего звено $(2, 1)$, равна $49 + 16 = 65$.

Чтобы определить множество маршрутов, содержащих звено $(2, 1)$, вычеркнем во второй матрице стоимости (см. табл. 3.16) вторую строку и первый столбец. Стоимость звена $(2, 1)$ равна теперь ∞ , но в третью матрицу решений этот элемент не входит. Однако звено $(4, 2)$ теперь является запрещенным, поскольку оно могло бы образовать подмаршрут. Поэтому полагаем $c_{42} = \infty$. В табл. 3.17 приведена третья матрица решений после выполнения редукции.

Таблица 3.17

Третья матрица решений

Узел	2	3	5	6	c_i	A_i
3	13	∞	5	0	0	5
4	∞	7	0	0	2	0
5	41	22	∞	0	0	22
6	0	0	0	∞	0	0
Q_i	0	0	0	0	$H_1 = 2$	—
B_i	13	7	0	0	—	—

Новая нижняя граница для маршрутов, содержащих звено $(2, 1)$, равна $49 + 2 = 51$. Ветвление на следующей итерации будет

осуществляться от звена $(2, 1)$. Полный маршрут приведен на рис. 3.20. Этот маршрут — промежуточное решение.

8. Построенный полный маршрут будет оптимальным, если его длина не превосходит длины любого маршрута, соответствующего другим звеньям дерева. Длина построенного полного маршрута равна 63, а нижняя граница для звена $(1, 4)$, равная 58, меньше 63. Необходимо исследовать и подмножество маршрутов, которые не содержат звено $(1, 4)$, т. е. начать рассмотрение с исходной матрицы стоимости (см. табл. 3.10). Но для того, чтобы исключить

все маршруты, содержащие звено $(1, 4)$, значение элемента c_{14} матрицы стоимости примем равным ∞ . Получим табл. 3.18. Процедуру анализа предыдущих промежуточных точек ветвления, которые могли бы определить более короткий маршрут, называют *возвратом*. Поэтому матрицу стоимости называют в данном случае *матрицей стоимости возврата*.

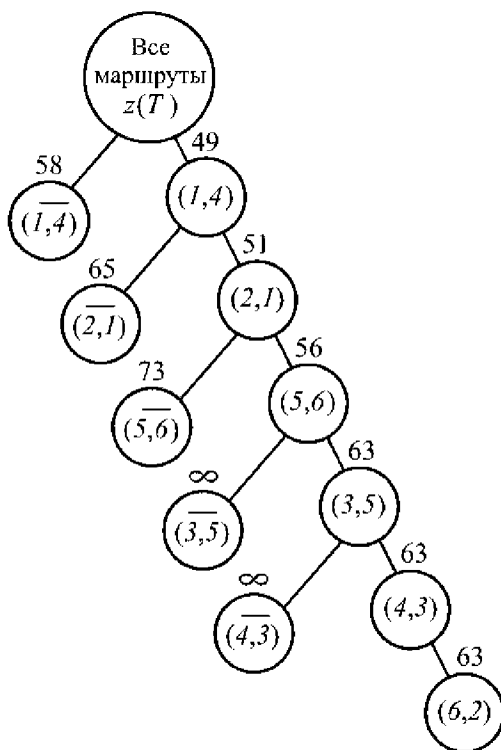


Рис. 3.20. Дерево промежуточных решений

Таблица 3.18

Матрица стоимости возврата

Узел	1	2	3	4	5	6
1	∞	27	43	∞	30	26
2	7	∞	16	1	30	30
3	20	13	∞	35	5	0
4	21	16	25	∞	18	18
5	12	46	27	48	∞	5
6	23	5	5	9	5	∞

9. С новой матрицей, матрицей стоимости возврата, выполняют описанные процедуры ветвления и построения границ. Полученное при этом дерево изображено на рис. 3.21. Нахождение верхних границ не обязательно, но эта операция иногда позволяет сократить проводимые вычисления. Из рис. 3.21 следует, что нижняя граница даже неполного маршрута, не содержащего звено $(1, 4)$, превышает 63. Таким образом, маршрут, содержащий звено $(1, 4)$, является оптимальным. Оптимальный маршрут состоит из следующих звеньев или пар пунктов: $(6, 2)$, $(4, 3)$, $(3, 5)$, $(5, 6)$, $(2, 1)$, $(1, 4)$. Он является ориентированным циклом, стоимость проезда по которому равна 63, т. е. 630 усл. ед.

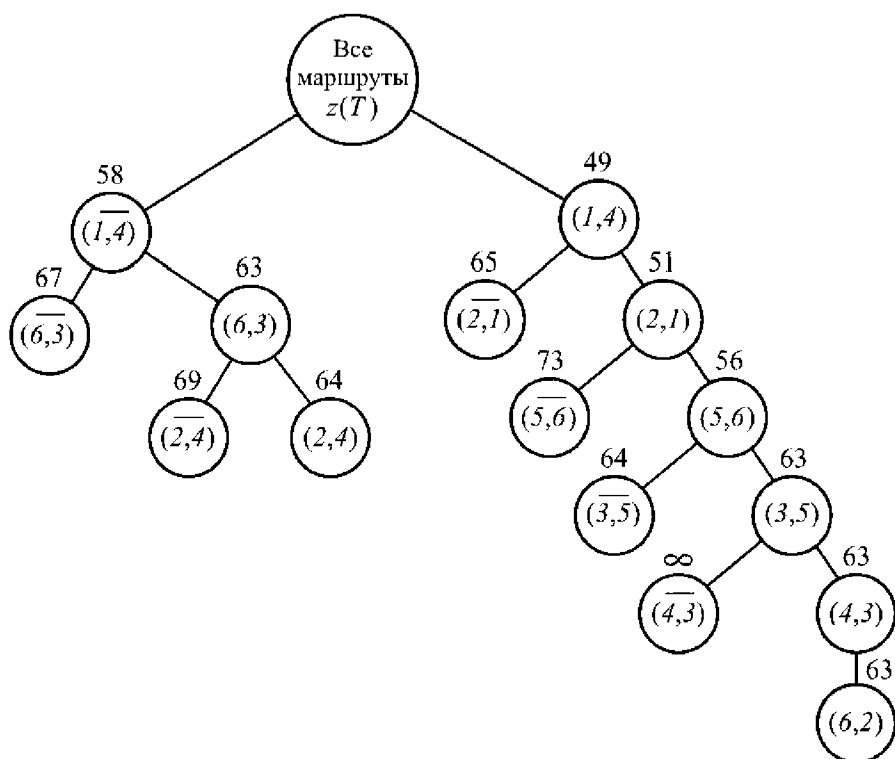


Рис. 3.21. Оптимальное решение

Задача коммивояжера не может быть непосредственно сформулирована и решена, как задача линейного программирования. Основная особенность задачи коммивояжера заключается в том, что в ней требуется существование ориентированного цикла, в который ровно один раз входят все узлы сети.

§ 3.10. Задача о многополюсной цепи с максимальной пропускной способностью

С задачей о многополюсном максимальном потоке тесно связана задача о *многополюсной цепи с максимальной пропускной способностью*. Алгоритм, описанный в § 3.8, позволяет находить максимальный поток между каждой парой узлов. Очевидно, максимальному потоку между каждой парой узлов могло соответствовать множество путей или цепей из источника в сток. В действительности в задаче о максимальном потоке рассматривают лишь те пути или цепи, с помощью которых можно увеличить поток из одного узла в другой. Рассмотрим простую сеть, изображенную на рис. 3.22. (Числа, приписанные дугам, соответствуют верхним границам потоков по ним.) Величина максимального потока между узлами A и D равна 40, а соответствующие потоки по дугам следующие: $x_{AB} = 20$, $x_{AC} = 25$, $x_{CB} = 5$, $x_{BD} = 25$, $x_{CD} = 15$. Узлы A и D соединены тремя цепями, как показано на рис. 3.22.

Рассмотрим задачу, относящуюся к приведенной на рис. 3.22 сети: какая цепь, ведущая из узла A в узел D , имеет максимальную пропускную способность? Очевидно, цепь с максимальной пропускной способностью определяют последовательностью узлов $A \rightarrow B \rightarrow D$. Величина максимального потока по этой цепи есть $F_{\max} = 20$. Задача, которую мы хотим рассмотреть в настоящем параграфе, это задача о многополюсной цепи с максимальной пропускной способностью, т. е. задача о цепи с максимальным потоком между всеми парами узлов.

Эффективную вычислительную процедуру, которая представляет собой модификацию трехместной операции, используемой при решении задачи о многополюсной кратчайшей цепи (пути), разработал Т. Ху. Данный алгоритм состоит в следующем.

Шаг 1. Построить матрицу пропускных способностей $\|d_{ij}\|$ размером $n \times n$, элементы которой соответствуют пропускным способностям дуг между узлами i и j , $i, j = 1, 2, \dots, n$, и матрицу маршрутов $\|r_{ik}\|$, заменив в ней все элементы столбцов одинаковыми числами, равными номеру столбца.

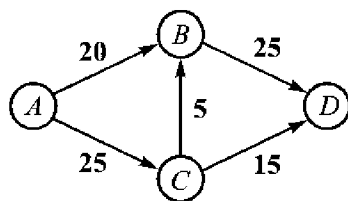


Рис. 3.22. Сеть для задачи о цепи с максимальной пропускной способностью

Шаг 2. Произвести над матрицей пропускных способностей следующие операции.

1) Для любых $i, j = 1, 2, \dots, n$ исключить j -ю строку и i -й столбец матрицы и над каждым оставшимся элементом d_{ik} (диагональные элементы также исключаются) выполнить трехместную операцию $d_{ik} = \max \{d_{ik}; \min(d_{ij}; d_{jk})\}$ для всех $i, k \neq j, j = 1, 2, \dots, n$.

Еще одна матрица, называемая *матрицей маршрутов*, необходима для определения внутренних узлов каждой цепи. Матрица маршрутов также имеет размер $n \times n$, где k -й элемент i -й строки в ней первоначально равен k .

2) Одновременно с заменами элементов в матрице пропускных способностей выполнить замены элементов в матрице маршрутов по следующему правилу:

$$r_{ik} = \begin{cases} j, & \text{если } d_{ik} < \min \{d_{ij}; d_{jk}\}, \\ \text{остается неизменным,} & \text{если } d_{ik} \geq \min \{d_{ij}; d_{jk}\}. \end{cases}$$

Если узлы i и j не соединены дугой (или связь между ними недопустима), то значение соответствующего элемента d_{ij} матрицы пропускных способностей полагается равным $-\infty$.

Для иллюстрации данного алгоритма рассмотрим следующую задачу.

Задача. Производятся перевозки автотранспортом крупногабаритного груза. Требуется перевозить несколько видов оборудования с очень большим вертикальным габаритным размером. Это оборудование можно перевозить только по семи утвержденным маршрутам. Задача заключается в выборе маршрутов с максимально допустимыми подмостовыми зазорами.

Таблица 3.19

Транспортировка крупногабаритного груза

Пункт погрузки	Пункт выгрузки						
	1	2	3	4	5	6	7
1	0	11	30	—	—	—	—
2	11	0	—	12	2	—	—
3	30	—	0	19	—	4	—
4	—	12	19	0	11	9	—
5	—	2	—	11	0	—	—
6	—	—	4	9	—	0	—
7	—	—	—	20	1	1	0

Каждый элемент матрицы, приведенной в табл. 3.19, равен высоте соответствующего моста. Необходимо получить информацию о максимально допустимых вертикальных габаритных размерах грузов, которые можно транспортировать из каждого пункта погрузки в каждый пункт выгрузки.

Решение. Для каждой пары узлов определяют соединяющий их маршрут с максимальной пропускной способностью. Ниже приводятся результаты вычислений.

Итерация 0. Выпишем исходные матрицы пропускных способностей и маршрутов:

$$\begin{pmatrix} 0 & 11 & 30 & -\infty & -\infty & -\infty & -\infty \\ 11 & 0 & -\infty & 12 & 2 & -\infty & -\infty \\ 30 & -\infty & 0 & 19 & -\infty & 4 & -\infty \\ -\infty & 12 & 19 & 0 & 11 & 9 & -\infty \\ -\infty & 2 & -\infty & 11 & 0 & -\infty & -\infty \\ -\infty & -\infty & 4 & 9 & -\infty & 0 & -\infty \\ -\infty & -\infty & -\infty & 20 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}.$$

Итерация 1. Согласно шагу 2 алгоритма Т. Ху изменим матрицы пропускных способностей и маршрутов:

$$\begin{pmatrix} 0 & 11 & 30 & -\infty & -\infty & -\infty & -\infty \\ 11 & 0 & 11 & 12 & 2 & -\infty & -\infty \\ 30 & 11 & 0 & 19 & -\infty & 4 & -\infty \\ -\infty & 12 & 19 & 0 & 11 & 9 & -\infty \\ -\infty & 2 & -\infty & 11 & 0 & -\infty & -\infty \\ -\infty & -\infty & 4 & 9 & -\infty & 0 & -\infty \\ -\infty & -\infty & -\infty & 20 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 1 & 4 & 5 & 6 & 7 \\ 1 & 1 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}.$$

Выполним аналогичные семь итераций. Приведем сразу результаты последней итерации.

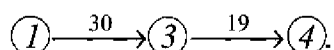
Итерация 7. В результате преобразований получим оптимальное решение:

$$\|d^*\| = \begin{pmatrix} 0 & 12 & 30 & 19 & 11 & 9 & -\infty \\ 12 & 0 & 12 & 12 & 11 & 9 & -\infty \\ 30 & 12 & 0 & 19 & 11 & 9 & -\infty \\ 19 & 12 & 19 & 0 & 11 & 9 & -\infty \\ 11 & 11 & 11 & 11 & 0 & 9 & -\infty \\ 9 & 9 & 9 & 9 & 9 & 0 & -\infty \\ 19 & 12 & 19 & 20 & 11 & 9 & 0 \end{pmatrix}, \|r^*\| = \begin{pmatrix} 1 & 4 & 3 & 3 & 4 & 4 & 7 \\ 4 & 2 & 4 & 4 & 4 & 4 & 7 \\ 1 & 4 & 3 & 4 & 4 & 4 & 7 \\ 3 & 2 & 3 & 4 & 5 & 6 & 7 \\ 4 & 4 & 4 & 4 & 5 & 4 & 7 \\ 4 & 4 & 4 & 4 & 4 & 6 & 7 \\ 4 & 4 & 4 & 4 & 4 & 4 & 7 \end{pmatrix}.$$

Для каждой пары узлов максимально допустимый вертикальный габаритный размер перевозимого груза можно определить непосредственно из последней матрицы пропускных способностей. Оптимальный маршрут также может быть построен с помощью последней матрицы маршрутов. Например, максимальная пропускная способность цепи из пункта 1 в пункт 4 определяется значением элемента $d_{14}^* = 19$, которому соответствует элемент $r_{14}^* = 3$ в матрице маршрутов. Теперь надо связать пункт 3 с пунктом 4 и пунктом 1. Поскольку $r_{34}^* = 4$ и $r_{13}^* = 3$, то соответствующий маршрут строят следующим образом:

- а) $r_{14}^* = 3$: двигаться из узла 1 в узел 4 через узел 3;
- б) $r_{34}^* = 4$: двигаться из узла 3 непосредственно в узел 4;
- в) $r_{13}^* = 3$: двигаться из узла 1 непосредственно в узел 3.

Отсюда получим путь из пункта 1 в пункт 4:



Действительно, $d_{14}^* = \min \{30; 19\} = 19$.

Отметим, что дуги, соединяющей узлы 1 и 4, не существует.

ОСНОВЫ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ И ТЕОРИИ ИГР

§ 4.1. Условия применимости динамического программирования

Как было показано в гл. 1, для нахождения оптимального решения непосредственное применение только необходимых признаков, как правило, не дает в задачах математического программирования нужных результатов. Во-первых, система уравнений, вытекающая из необходимых признаков, оказывается разрешимой только в простейших случаях. Бывает легче непосредственно предсказать максимум (минимум) целевой функции, чем решить такую систему уравнений. Во-вторых, указанный способ вовсе не гарантирует нахождение решения во всех случаях. Даже если составленная система уравнений может быть решена, отыскание абсолютного экстремума целевой функции требует проведения проверок, тем более сложных, чем больше аргументов имеет функция.

Наконец, в ряде практических случаев целевую функцию вообще нельзя дифференцировать, например когда аргументы $x = (x_1, x_2, \dots, x_m)$ представляют собой не непрерывно изменяющиеся величины, а дискретные.

Все эти обстоятельства приводят к тому, что применение классических методов математического анализа или вариационного исчисления в большинстве задач планирования оказывается неэффективным, при этом первоначально поставленная задача отыскания экстремума приводит к таким вторичным задачам, которые оказываются не проще исходной, а зачастую сложнее.

В гл. 1 было также показано, что все задачи математического оптимального программирования в зависимости от вида целевой функции и ограничений могут быть разбиты на классы, характеризуемые своими методами решения. К примеру, в линейном программировании изучают класс задач, в которых и целевые функции, и ограничения линейны.

Вместе с тем решение многих таких задач может быть упрощено, если решать их поэтапно, т. е. использовать *метод динамического программирования*. Идея метода состоит в том, что отыскание точек экстремума целевой функции многих переменных заменяется последовательным нахождением точек экстремума функции одного или небольшого числа переменных.

Итак, динамическое программирование есть поэтапное планирование многошагового процесса, при котором управление разбивается на ряд последовательных этапов, соответствующих, как правило, различным моментам времени или разным координатам.

В задачах, решаемых методом динамического программирования, значение целевой функции (оптимизируемого критерия), достигнутое в течение всего процесса, получают простым суммированием частных значений $f_i(x)$ того же критерия, полученных на отдельных шагах, т. е.

$$f(x) = \sum_{i=1}^m f_i(x).$$

Если критерий оптимальности $f(x)$ обладает этим свойством, то он называется *аддитивным*.

Во многих практических задачах критерий $f(x)$ аддитивен; если в первоначальной постановке задачи критерий не аддитивен, то постановку задачи надо видоизменить так, чтобы он стал аддитивным. К примеру [4, 5], если рассматривают критерий $f(x)$, представленный в виде произведения частных значений, достигаемых на отдельных этапах: $f(x) = f_1(x)f_2(x)\dots f_m(x)$ (такой критерий называют *мультиплексным*), то можно преобразовать его к аддитивному, прологарифмировав:

$$\lg f(x) = \sum_{i=1}^m \lg f_i(x).$$

Обозначим $V = \lg f(x)$, $V_i = \lg f_i(x)$. Получим новую целевую функцию (критерий) $V = \sum_{i=1}^m V_i$, обладающую свойством аддитивности и достигающую максимум (минимум) одновременно с функцией $f(x)$.

Рассмотрим общую схему решения задач, имеющих аддитивный критерий.

Пусть процесс управления некоторой системой состоит из m шагов. На каждом i -м шаге управление x_i переводит систему из состояния S_{i-1} , достигнутого в результате $(i-1)$ -го шага, в новое состояние S_i , которое зависит от состояния S_{i-1} и выбранного управления x_i :

$$S_i = S_i(S_{i-1}, x_i).$$

Здесь существенно то, что новое состояние S_i зависит только от состояния S_{i-1} и управления x_i и не зависит от того, каким образом система пришла в состояние S_{i-1} . В крайнем случае это условие достигается увеличением числа состояний системы. Понятием «состояние системы» характеризуются только те параметры, от которых зависит будущий результат.

В теории динамического программирования, когда рассматривают процессы, зависящие только от текущего состояния, оптимальную стратегию формулируют следующим образом.

Принцип оптимальности. *Оптимальная стратегия обладает тем свойством, что, каковы бы ни были первоначальные состояния системы S_{i-1} и решение x_i , последующее решение должно определять оптимальную стратегию относительно состояния, полученного в результате начального решения.*

Другими словами, каково бы ни было состояние системы, перед очередным шагом необходимо выбрать управление на этом шаге так, чтобы выигрыш на данном шаге и оптимальный выигрыш на всех последующих шагах были максимальными.

Рассмотрим задачу о нахождении максимума функции дохода $f(x)$ на m -шаговом процессе.

Под влиянием управлений x_1, x_2, \dots, x_m система переходит из начального состояния S_0 в конечное $S_{\text{кон}}$. За m шагов получают целевую функцию (выигрыш)

$$f(x) = \sum_{i=1}^m f_i(S_{i-1}, x_i),$$

где $f_i(S_{i-1}, x_i)$ — выигрыш на i -м шаге, зависящий от исходного состояния системы S_{i-1} и выбранного управления x_i .

Принцип оптимальности позволяет заключить, что при любом начальном управлении x_1

$$f(x) = f_1(S_0, x_1) + (f_2(S_1, x_2) + \dots + f_m(S_{m-1}, x_m)) =$$

$$= f_1(S_0, x_1) + \sum_{i=2}^m f_i(S_{i-1}, x_i).$$

Поскольку данное соотношение справедливо для любых начальных решений x_1 , то, чтобы найти максимальный доход $f_{\max}(S_0)$, необходимо найти максимум по x_1 значения $f(x)$:

$$f_{\max}(S_0) = \max_{x_1} f(x) = \max_{x_1} \{f_1(S_0, x_1) + f_{\max}(S_{m-1}(S_0, x_1))\},$$

где $f_{\max}(S_{m-1}(S_0, x_1))$ — оптимальный выигрыш на всех последующих шагах.

Получили основное функциональное уравнение динамического программирования. Его можно вывести, используя формулу

$$\max_{x_1, \dots, x_m} f(x) = \max_{x_1} \left\{ \max_{x_2, \dots, x_m} f(x) \right\}.$$

Следовательно,

$$\begin{aligned} f_{\max}(S_0) &= \max_{x_1, \dots, x_m} \{f_1(S_0, x_1) + f_2(S_1, x_2) + \dots + f_m(S_{m-1}, x_m)\} = \\ &= \max_{x_1} \left\{ \max_{x_2, \dots, x_m} \{f_1(S_0, x_1) + f_2(S_1, x_2) + \dots + f_m(S_{m-1}, x_m)\} \right\} = \\ &= \max_{x_1} \{f_1(S_0, x_1)\} + \max_{x_2, \dots, x_m} \{f_2(S_1, x_2) + \dots + f_m(S_{m-1}, x_m)\} = \\ &= \max_{x_1} \{f_1(S_0, x_1) + f_{\max}(S_{m-1}(S_0, x_1))\}. \end{aligned}$$

Согласно этому выражению алгоритм получения решения задач динамического программирования состоит в оптимизации последовательности функций дохода $\{f_n(S_{n-1})\}$ или последовательности стратегий $\{x_n(S_{n-1})\}$. Эти последовательности определяют одна другую. Причем имеется только одна последовательность оптимальных функций дохода, хотя может быть много последовательностей оптимальных стратегий, которые приведут к тому же максимальному доходу.

В динамическом программировании, планируя многоэтапную процедуру, управление на каждом шаге, кроме последнего, выбирают с учетом будущего. Последний шаг можно спланировать так, чтобы он привел к максимуму целевой функции.

Возникает вопрос: как спланировать последний шаг, если мы не знаем, чем кончился предпоследний? Для этого делают разные

предположения о том, чем кончится предпоследний шаг, и для каждого допущения выбирают управление на последнем шаге и запоминают его до конца решения задачи.

Спланировав оптимальным образом последний шаг, к нему присоединяют предпоследний, при этом формулируют гипотезу о том, чем закончился предыдущий шаг, и находят согласно основному функциональному уравнению наибольшее значение целевой функции на этих двух шагах и т. д. Поэтому в динамическом программировании процесс решения задачи происходит от конца к началу. Этот процесс повторяют на каждом шаге. Такое оптимальное управление, выбранное в предположении, чем кончится предыдущий шаг, называют *условным*. Зная исходные данные для первого шага, можно получить значения оптимального управления от первого до n -го шага.

Приведем далее примеры решения задач методом динамического программирования.

§ 4.2. Задача об оптимальной загрузке транспортного средства неделимыми предметами

В § 1.1 мы сформулировали задачу о бабе, отправляющейся на рынок и решающей, сколько ей надо взять *живых* гусей, кур и уток, чтобы получить наибольшую выручку. Мы знаем и то, что это задача целочисленного линейного программирования. Теперь рассмотрим, как подобные задачи решаются с помощью методов динамического программирования.

Задача. Пусть в самолет требуется погрузить четыре вида предметов, чтобы эффект от этих предметов (например, стоимость) был максимальным. Грузоподъемность самолета равна \bar{w} ; P_i — вес единицы i -го вида предметов; v_i — стоимость единицы i -го вида; x_i — количество i -го вида предметов, взятых на борт самолета.

Решение. Найдем максимум целевой функции задачи:

$$f(x) = \sum_{i=1}^4 x_i v_i \equiv x_1 v_1 + x_2 v_2 + x_3 v_3 + x_4 v_4 \rightarrow \max.$$

Ограничение записывается следующим образом:

$$\sum_{i=1}^4 x_i P_i = x_1 P_1 + x_2 P_2 + x_3 P_3 + x_4 P_4 \leq \bar{w}.$$

Эту задачу называют *задачей о рюкзаке* вследствие следующей гипотетической ситуации. Турист, собираясь в поход, должен выбрать, какие предметы ему взять с собой. Каждый предмет имеет свой вес и свою ценность. Требуется, чтобы общий вес рюкзака с выбранными предметами не превышал некоторого заданного веса, а их общая ценность была максимальной.

Для численного решения предположим, что $\bar{w} = 83$, $P_1 = 24$, $P_2 = 22$, $P_3 = 16$, $P_4 = 10$, $v_1 = 96$, $v_2 = 85$, $v_3 = 50$, $v_4 = 20$ соответствующих единиц.

Очевидно, что это задача целочисленного программирования.

Применим для ее решения многошаговую процедуру принятия решения. Согласно общей схеме решения задач динамического программирования будем рассматривать два этапа. На первом этапе найдем возможные оптимальные варианты при последовательной оптимизации по одной переменной, а на втором этапе из этих «заготовок» выберем оптимальное решение нашей задачи.

Первый этап будет содержать четыре шага.

Шаг 1. Сначала найдем возможные оптимальные варианты загрузки самолета только предметами первого вида. Необходимо найти и запомнить значения x_1 и соответствующую им максимальную стоимость груза $f_1(w)$ при различных возможных значениях w . В этом случае максимальная стоимость груза определится следующим образом:

$$f_1(w) = \max_{x_1} \{x_1 v_1\}$$

при условии $x_1 P_1 \leq w$, $x_1 = 0, 1, 2, \dots$. Поскольку $x_1 \leq w/P_1$, то для нахождения максимума $f_1(w)$ необходимо взять x_1 возможно большим, т. е. $x_1 = [w/P_1]$ — наибольшее целое число, не превосходящее w/P_1 , и, таким образом, $f_1(w) = [w/P_1] v_1$.

Зададим некоторые значения w и найдем для них x_1 и $f_1(w)$. Очевидно, если грузоподъемность самолета меньше 24 ед., то ни одного предмета первого вида погрузить нельзя. При грузоподъемности от 24 до 47 ед. можно погрузить 1 ед. предмета первого вида и т. д. Результаты представим в виде таблиц. Значения w , $f_1(w)$ и x_1 приведены в табл. 4.1. В таблицах принят более широкий предел значений для w — от 0 до 87 ед.

Таблица 4.1

Первый шаг оптимизации

w	$f_1(w)$	x_1	w	$f_1(w)$	x_1
0–23	0	0	48–71	192	2
24–47	96	1	72–87	288	3

Шаг 2. Проведем оптимизацию целевой функции при условии, что самолет загружают предметами первого и второго видов. Необходимо найти стоимость груза в этом случае, максимальную эффективность загрузки обозначим через $f_2(w)$. Если предметов второго вида взято x_2 , то вес предметов первого вида должен быть не больше чем $w - P_2 x_2$. Максимальная стоимость предметов первого вида выразится формулой

$$f_1(w - x_2 P_2) = \max_{x_1} \{x_1 v_1\}, \quad x_1 = \frac{w - x_2 P_2}{P_1},$$

а общая стоимость груза — формулой

$$f_{1,2} = x_2 v_2 + f_1(w - x_2 P_2).$$

Тогда

$$f_2(w) = \max_{x_2} \{x_2 v_2 + f_1(w - x_2 P_2)\}, \quad 0 \leq x_2 \leq \left\lfloor \frac{w}{P_2} \right\rfloor. \quad (4.1)$$

Максимум этого выражения определяют только по x_2 . Однако мы не знаем, какую часть веса w могут занимать предметы второго вида. Поэтому мы должны рассмотреть выражение (4.1) для всевозможных значений w от 0 до 87 ед. При вычислении $f_1(w - x_2 P_2)$ воспользуемся полученными результатами (см. табл. 4.1).

Например, пусть $w = 46$ ед. Величина x_2 может принимать значения 0, 1, 2; соответствующая стоимость предметов второго вида равна 0, 85, 170 ед., а для предметов первого вида составит 46, 24, 2 ед. Из табл. 4.1 для $w = 46$; 24; 2 ед. находим $f_1(w) = 96$; 96; 0 ед. Суммируем соответствующие значения: $0 + 96 = 96$ ед., $85 + 96 = 181$ ед., $170 + 0 = 170$ ед., и выбираем наибольшее из них: 181 ед. Оно получено для $x_2 = 1$. Для $w = 46$ заносим в табл. 4.2 $x_2 = 1$, $f_2(w) = 181$. Оптимальные результаты вычисления $f_2(w)$ и x_2 приведены в табл. 4.2.

Таблица 4.2

Второй шаг оптимизации

w	$f_2(w)$	x_2	w	$f_2(w)$	x_2
0–21	0	0	48–65	192	0
22–23	85	1	66–67	255	3
24–43	96	0	68–69	266	2
44–45	170	2	70–71	277	1
46–47	181	1	72–87	288	0

Из табл. 4.2 следует, что при грузоподъемности транспортного средства до 21 ед. ничего в него погрузить нельзя, при грузоподъемности 22...23 ед. можно погрузить только один предмет второго вида, при грузоподъемности 24...43 ед. — либо один предмет первого вида, либо один предмет второго вида. Максимальной стоимость груза будет при погрузке предмета первого вида. При грузоподъемности 44...45 ед. можно погрузить либо один предмет первого вида, либо два предмета второго вида. Бóльшая стоимость груза будет в последнем случае: $f_2(w) = 170$. При грузоподъемности 46...47 ед. можно погрузить один предмет первого вида, два предмета второго вида или по одному предмету первого и второго видов. Стоимость груза в последнем варианте — максимальная.

Далее аналогично.

Шаг 3. Проведем оптимизацию целевой функции при условии, что самолет загружают предметами первых трех видов. Требуется максимизировать по x_3 функцию

$$f_{1,2,3} \equiv x_3 v_3 + f_2(w - x_3 R_3); \quad f_3(w) = \max_{x_3} f_{1,2,3}, \quad 0 \leq x_3 \leq \left\lfloor \frac{w}{R_3} \right\rfloor.$$

Задаем значение w и для каждого такого значения получаем максимум $f_{1,2,3}$ по x_3 . Значения $f_2(w)$ берут в табл. 4.2. Например, пусть $w = 38$ ед. Возможное значение $x_3 = 0, 1, 2$ ед., и соответствующая стоимость предметов третьего вида равна 0; 50; 100 ед. На предметы первого и второго видов остается соответственно вес 38; 22; 6 ед. По табл. 4.2 находим $f_2(w) = 96; 85; 0$ ед. Суммарная стоимость будет равна 96; 135; 100 ед. Максимальное значение стоимости при $w = 38$ ед. равно 135 ед. при $x_3 = 1$. Далее аналогично. Результаты расчетов помещены в табл. 4.3.

Таблица 4.3

Третий шаг оптимизации

w	$f_3(w)$	x_3	w	$f_3(w)$	x_3
0–15	0	0	60–61	220	1
16–21	50	1	62–63	231	1
22–23	85	0	64–65	242	1
24–31	96	0	66–67	255	0
32–37	100	2	68–69	266	0
38–39	135	1	70–71	277	0
40–43	146	1	72–79	288	0
44–45	170	0	80–81	292	2
46–47	181	0	82–83	305	1
48–55	192	0	84–85	316	1
56–59	196	2	86–87	327	1

Шаг 4. Получим оптимальный вес груза w^* , который может принять самолет.

Зададим значение w , которое может быть занято четвертым грузом, и вычислим x_4 , $f_4(w)$ (табл. 4.4).

Таблица 4.4

Четвертый шаг оптимизации

w	$f_4(w)$	x_4	w	$f_4(w)$	x_4
0–9	0	0	58–59	212	1
10–15	20	1	60–61	220	0
16–21	50	0	62–63	231	0
22–23	85	0	64–65	242	0
24–31	96	0	66–67	255	0
32–33	105	1	68–69	266	0
34–37	116	1	70–71	277	0
38–39	135	0	72–79	288	0
40–43	146	0	80–81	297	1
44–45	170	0	82–83	308	1
46–47	181	0	84–85	316	0
48–55	192	0	86–87	327	0
56–57	201	1			

Второй этап решения задачи состоит в нахождении оптимального решения. Максимальное значение $f_4(w)$ определяет соответствующее значение аргумента x_4 — количество груза четвертого вида, который берет самолет (x_4^*). При $w = 83$ имеем

$$f_4(w^*) = \max f_4(w) = 308, \quad x_4^* = 1.$$

Вес, занятый предметами четвертого вида, будет $x_4^* P_4 = 10$ ед. На остальные три вида груза остается $w - 10 = 73$ ед. Входим с этим значением в табл. 4.3 и получаем $x_3^* = 0$, а затем $x_2^* = 0$, $x_1^* = 3$.

Строго говоря, нет необходимости заполнять всю табл. 4.4, достаточно заполнить только одну строку для $w = 83$.

Из табл. 4.1–4.4 видим, что они содержат «заготовки» для оптимальной загрузки самолета любой грузоподъемности до $w = 87$ ед. указанными предметами. Таким образом, мы решили не только поставленную задачу, но и ряд родственных задач. При этом в памяти компьютера необходимо держать табл. 4.1–4.4: столбцы $f_1(w)$, $f_2(w)$, $f_3(w)$, $f_4(w)$ на протяжении следующего шага, а столбцы x_1 , x_2 , x_3 , x_4 и соответствующие им w — на протяжении всего решения. Для сложных задач последнее обстоятельство имеет существенное значение и ограничивает возможности решения задач рассмотренным методом.

В экономических приложениях это задача о распределении выделенных средств S между предприятиями для обеспечения максимального количества выпускаемой продукции.

В приложении П5 представлено описание компьютерной программы (инструкция пользователю) решения задачи о рюкзаке.

§ 4.3. Задача о вкладе средств в производство

Задача. Планируется организация деятельности двух производств (например, швейного цеха и цеха по ремонту обуви) сроком на 5 лет ($m = 5$). Функции вклада средств в соответствующее производство имеют вид $\varphi(x) = 0,75x$, $\psi(y) = 0,3y$, т. е. если средства z вкладывать только в первую отрасль, то по годам они будут вкладываться $z \cdot 0,75$; $z \cdot 0,75^2$; $z \cdot 0,75^3$; ... Соответственно, вкладывая средства z только во вторую отрасль, получаем $z \cdot 0,3$; $z \cdot 0,3^2$; $z \cdot 0,3^3$; ...

Функции дохода как функции объема вкладываемых средств x и y имеют вид

$$f(x) = 1 - e^{-x}, \quad g(y) = 1 - e^{-2y}.$$

Требуется распределить ресурсы $z = 2$ ед. между отраслями (цехами) по годам так, чтобы получить максимальный доход.

Решение. Решение задачи не удастся получить в аналитическом виде, решим ее графически. Пусть к началу пятого года количество средств равно z_4 и на пятый год потребуется средств первому производству x_5 , тогда второму производству нужно $y_5 = z_4 - x_5$. Чтобы найти *условное оптимальное управление* на пятом шаге $x_5^*(z_4)$, нужно для каждого z_4 найти максимум функции

$$f_5 = 1 - e^{-x_5} + 1 - e^{-2(z_4 - x_5)} = 2 - (e^{-x_5} + e^{-2(z_4 - x_5)}).$$

При фиксированном z_4 максимум функции f_5 достигается либо при $x_5 = 0$, либо внутри отрезка $[0, z_4]$. Максимум внутри отрезка будет в точке x_5^* , которую находим из условия

$$\begin{aligned} \frac{\partial f_5}{\partial x_5} &= 0, \quad e^{-x_5} - 2e^{-2(z_4 - x_5)} = 0, \\ x_5^* &= \frac{1}{3}(2z_4 - \ln 2), \quad x_5 \geq 0. \end{aligned}$$

Из условия $x_5 \geq 0$ следует $z_4 \geq (\ln 2)/2$, т. е. при $z_4 \geq (\ln 2)/2 \approx 0,347$ максимум достигается в точке

$$x_5^*(z_4) = \frac{1}{3}(2z_4 - \ln 2).$$

При $z_4 \leq (\ln 2)/2$ максимум будет в точке $x_5^*(z_4) = 0$. Итак, *условное оптимальное управление* на пятом шаге можно записать в следующем виде:

$$x_5^*(z_4) = \begin{cases} 0 & \text{при } z_4 \leq \frac{\ln 2}{2}, \\ \frac{1}{3}(2z_4 - \ln 2) & \text{при } z_4 > \frac{\ln 2}{2}. \end{cases} \quad (4.2)$$

Построим (рис. 4.1) зависимости $x_5^* = x_5^*(z_4)$ и $f_5^* = f_5^*(z_4)$. При построении зависимости $f_5^*(z_4)$ пользуемся соотношением (4.2).

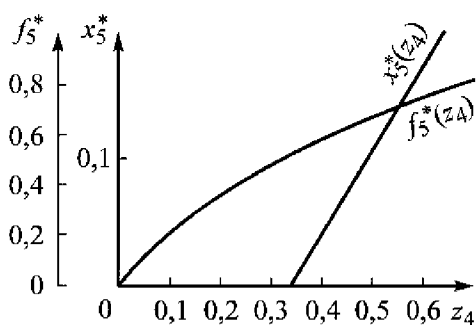


Рис. 4.1. Зависимости f_5^* и x_5^* от аргумента z_4

Перейдем к оптимизации на четвертом шаге. Запас средств после третьего шага обозначим z_3 . Определим, в каких пределах может изменяться z_3 . Наибольшее значение z_3 будет достигнуто, если все средства вложить в первую отрасль:

$$\begin{aligned} z_{3\max} &= z \cdot 0,75^3 = \\ &= 2 \cdot 0,75^3 = 0,844. \end{aligned}$$

Наименьший запас средств соответствует случаю, когда все средства на трех шагах вложены во вторую отрасль:

$$z_{3\min} = z \cdot 0,3^3 = 0,054.$$

Будем рассматривать теперь набор значений z_3 от 0,1 до 0,8 с шагом 0,1. Для каждого значения z_3 найдем *условное оптимальное управление* $x_4^*(z_3)$ на четвертом шаге и *условный оптимальный доход* $f_4^*(z_3)$ на двух последних шагах.

Для этого построим серию кривых, показывающих выигрыш на двух последних шагах при любом управлении на четвертом и при оптимальном на пятом:

$$\begin{aligned} f_{4,5} &= f_4(z_3, x_4) + f_5^*(z_4) = f_4(z_3, x_4) + f_5^*(0,75x_4 + 0,3(z_3 - x_4)), \\ f_4(z_3, x_4) &= 2 - (e^{-x_4} + e^{-2(z_3 - x_4)}), \end{aligned}$$

значение $f_5^*(z_4)$ берем из рис. 4.1.

Значения $f_{4,5}(x_4)$ приведены на рис. 4.2. Для каждого значения z_3 максимальная ордината определяет *условный максимальный выигрыш* $f_{4,5}^*(z_3)$ на двух последних шагах, а абсцисса — *условное оптимальное управление* $x_4^*(z_3)$.

На рис. 4.3 построены зависимости $f_{4,5}^*(z_3)$ и $x_4^*(z_3)$.

Аналогично решают задачу для третьего и второго шагов. Результаты приведены на рис. 4.4–4.7.

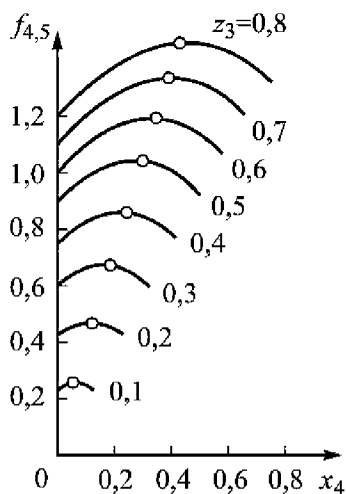


Рис. 4.2. Зависимости $f_{4,5}$ от аргумента x_4 и параметра z_3

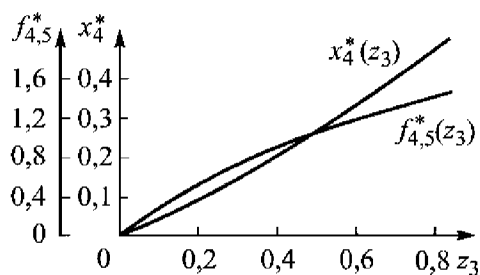


Рис. 4.3. Зависимости $f_{4,5}^*$ и x_4^* от аргумента z_3

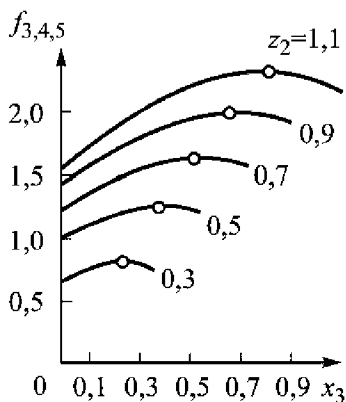


Рис. 4.4. Зависимости $f_{3,4,5}$ от аргумента x_3 и параметра z_2

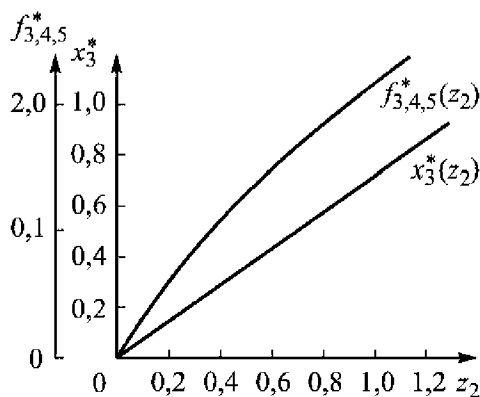


Рис. 4.5. Зависимости $f_{3,4,5}^*$ и x_3^* от аргумента z_2

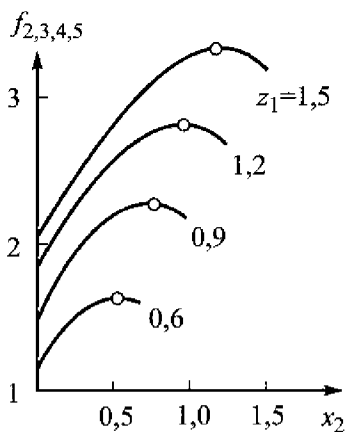


Рис. 4.6. Зависимости $f_{2,3,4,5}$ от аргумента x_2 и параметра z_1

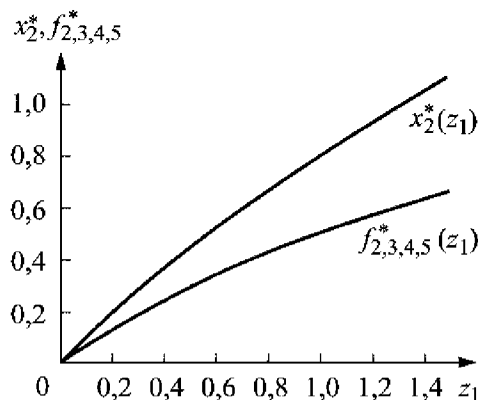


Рис. 4.7. Зависимости $f_{2,3,4,5}^*$ и x_2^* от аргумента z_1

Спланируем первый шаг. Доход на пяти шагах (при любом управлении на первом шаге и при оптимальном на последующих) определяем по формулам

$$f_{1,2,3,4,5}(x_1) = f_1(z, x_1) + f_{2,3,4,5}^*(z_1) = \\ = 2 - (e^{-x_1} + e^{-2(z-x_1)}) + f_{2,3,4,5}^*(z_1), \quad z_1 = 0,75x_1 + 0,3(z - x_1).$$

На первом шаге имеющиеся ресурсы известны: $z = 2$. Поэтому функция $f_{1,2,3,4,5}(x_1)$ опишется одной линией (рис. 4.8). На рис. 4.8

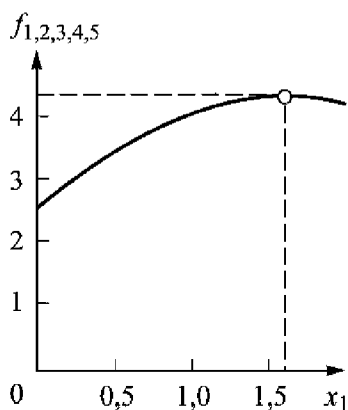


Рис. 4.8. Зависимость $f_{1,2,3,4,5}$ от аргумента x_1

видно, что максимальный доход будет равен $f_{1,2,3,4,5}^* = 4,35$; оптимальное управление на первом шаге — $x_1^* = 1,60$. По известному оптимальному управлению на первом шаге можно определить запас средств к концу этого шага, что, в свою очередь, определит оптимальное управление на следующем шаге.

Вычислим запас средств к концу первого шага:

$$z_1^* = 0,75x_1^* + 0,3(z - x_1^*) = 1,32.$$

Оптимальное управление на втором шаге тогда (см. рис. 4.7) будет иметь значение $x_2^* = 1,02$. Остаток средств к концу второго шага

$$z_2^* = 0,75x_2^* + 0,3(z_1^* - x_2^*) = 0,86;$$

и, далее, $x_3^* = 0,62$, $z_3^* = 0,54$, $x_4^* = 0,30$, $z_4^* = 0,30$, $x_5^* = 0$.

Получим оптимальное управление, показывающее, сколько средств надо вкладывать в первую отрасль:

$$x^* = (x_1^*; x_2^*; x_3^*; x_4^*; x_5^*) = (1,60; 1,02; 0,62; 0,30; 0).$$

Автоматически находим количество средств, вкладываемых во вторую отрасль:

$$y_1^* = z - x_1^* = 0,40, \quad y_2^* = z_1^* - x_2^* = 0,30, \quad y_3^* = z_2^* - x_3^* = 0,24, \\ y_4^* = z_3^* - x_4^* = 0,24, \quad y_5^* = z_4^* - x_5^* = 0,30.$$

Определим остаток средств: $0,30 \cdot 0,3 = 0,09$.

§ 4.4. Задача о распределении средств поражения

Рассмотрим задачу о распределении средств поражения по обороняющимся целям, которая аналогична задаче о резервировании средств.

Задача. Планируются боевые действия самолетами по обороняющимся целям. Цели эшелонированы по глубине территории на четырех параллельных рубежах. Перед тем как выйти на данный рубеж, самолеты проходят зону действия огневых средств этого рубежа, где подвергаются обстрелу со стороны последних. Огневые средства каждого рубежа могут вести огонь не только по средствам поражения, направляющимся непосредственно на цели данного рубежа, но и по тем самолетам, которые проходят через зону действия, направляясь на следующие рубежи.

Вероятность поражения одного самолета, пролетающего зону действия орудий i -го рубежа, определяется формулой

$$v_i = 1 - e^{-\alpha_i \bar{N}_i}, \quad i = 1, 2, 3, 4,$$

где \bar{N}_i — среднее число орудий, сохранивших боеспособность на i -м рубеже; α_i — эффективность стрельбы орудий по самолетам.

Среднее число орудий i -го рубежа, пораженных j -й волной вылетов самолетов, направленных на цели этого рубежа, вычисляется по формуле

$$Q_i = N_i \left(1 - e^{(-\bar{v}_j / N_i) P_i} \right),$$

где N_i — число орудий на i -м рубеже; \bar{v}_j — среднее число самолетов в j -й волне вылетов, сохранивших боевые свойства до i -го рубежа; P_i — средняя вероятность поражения одного орудия i -го рубежа атакующим его самолетом.

Имеется N орудий на всех рубежах:

$$N = \sum_{i=1}^4 N_i,$$

где N_i — число орудий на i -м рубеже: $N_1 = 10$, $N_2 = 12$, $N_3 = 15$, $N_4 = 10$, отсюда $N = 47$.

Имеется $z_0 = 80$ самолетов, которые необходимо распределить на четыре волны вылетов так, чтобы сделать максимальным среднее число поражений на всех рубежах:

$$f(x) = \sum_{i=1}^4 f_i(x),$$

где $f_i(x)$ — среднее число поражений на i -м рубеже.

Волны вылетов распределены по времени так, чтобы к моменту подлета следующей волны предшествующая ей волна самолетов успела выполнить боевое задание.

Пусть

$$\begin{aligned} P_1 &= 0,4, & P_2 &= 0,5, & P_3 &= 0,4, & P_4 &= 1,0, \\ \alpha_1 &= 0,05, & \alpha_2 &= 0,04, & \alpha_3 &= 0,04, & \alpha_4 &= 0,05. \end{aligned}$$

Решение. Будем пользоваться общей схемой решения задач методом динамического программирования. Проведем оптимизацию на четвертом (последнем) шаге. К зоне действия орудий 4-го рубежа подойдет z_3 самолетов. Очевидно, что все они должны быть направлены на поражение орудий четвертого рубежа. Условное оптимальное управление на 4-м шаге определяется по формуле

$$x_4^*(z_3) = z_3.$$

На четвертом рубеже установлено N_4 орудий, этот рубеж не подвергался бомбежке, т. е.

$$\bar{N}_4 = N_4.$$

Подлетевшие z_3 самолетов поразят на четвертом рубеже $f_4^*(z_3)$ орудий:

$$Q_4(x_4^*) \equiv f_4^*(z_3) = N_4(1 - e^{(-\bar{v}_4/N_4)P_4}),$$

где $\bar{v}_4(z_3) = z_3(1 - v_4)$; v_4 — вероятность поражения одного самолета на четвертом рубеже; $v_4 = 1 - e^{-\alpha_4 N_4}$, т. е. $\bar{v}_4(z_3) = z_3 e^{-\alpha_4 N_4}$.

Задавая различные значения z_3 , получим график функции $f_4^*(z_3)$ (рис. 4.9). Вообще-то следовало бы брать значения z_3 в интервале $(0, 80)$, но в данной задаче в этом нет необходимости.

Проведем оптимизацию на третьем шаге. Зададим число z_2 самолетов, преодолевших первый и второй рубежи: $z_2 \in (10, 40)$. Для каждого значения z_2 вычислим суммарный выигрыш: на третьем шаге при любом управлении, на четвертом — при оптимальном, т. е.

$$f_{3,4} = Q_3(x_3) + f_4^*(z_3).$$

Из z_2 самолетов надо выделить x_3 на подавление орудий третьего рубежа, а $z_2 - x_3$ самолетов направить на четвертый рубеж через зону огня третьего рубежа. Условное оптимальное управление $x_3^*(z_2)$ найдем из условия максимального выигрыша на двух последних шагах.

Здесь $Q_3(x_3)$ — среднее число орудий, пораженных на третьем рубеже выделенными для этой цели x_3 самолетами. При таком управлении до четвертого рубежа дойдут z_3 самолетов. Имеем

$$Q_3(x_3) = N_3(1 - e^{(-\bar{v}_3 / N_3)P_3}),$$

где $\bar{v}_3 = x_3(1 - v_3) = x_3(1 - 1 + e^{-\alpha_3 N_3}) = x_3 e^{-\alpha_3 N_3}$.

Вычислим среднее число самолетов из оставшихся $z_2 - x_3$ самолетов, прошедших через огонь третьего рубежа и дошедших до четвертого рубежа. На третьем рубеже осталось в действии $\bar{N}_3 = N_3 - Q_3(x_3)$ орудий. Тогда среднее число дошедших самолетов до четвертого рубежа из полного числа $z_2 - x_3$ будет

$$z_3 = (z_2 - x_3)e^{-\alpha_3 \bar{N}_3},$$

где величина \bar{N}_3 зависит от x_3 . Следовательно,

$$f_{3,4}^*(z_2) = \max_{0 \leq x_3 \leq z_2} \left\{ N_3 \left(1 - \exp \left(-\frac{x_3 e^{-\alpha_3 N_3}}{N_3} P_3 \right) \right) + f_4^* \left((z_2 - x_3) e^{-\alpha_3 \bar{N}_3(x_3)} \right) \right\}.$$

Задав значения z_2 и взяв значения $f_4^*(z_3)$ согласно графику, приведенному на рис. 4.9, для каждого из них получим зависимости $f_{3,4}(x_3)$ (рис. 4.10). Отмечая точки максимума функции $f_{3,4}(x_3)$, для каждого z_2 строим по значениям, приведенным на рис. 4.10, графики зависимостей $f_{3,4}^*$ и x_3^* от z_2 (рис. 4.11).

Оптимизируем решение на втором шаге. Процедура аналогична оптимизации, проведенной на третьем шаге.

Для разных значений z_1 (число самолетов, преодолевших первый рубеж) вычислим (рис. 4.12) целевую функцию

$$f_{2,3,4} = Q_2(x_2) + f_{3,4}^*(z_2).$$

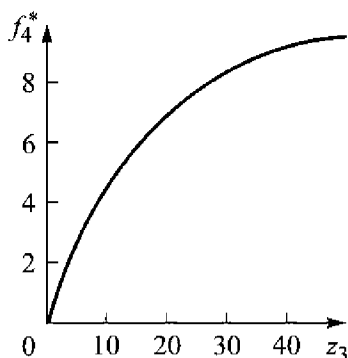


Рис. 4.9. Зависимость f_4^* от аргумента z_3

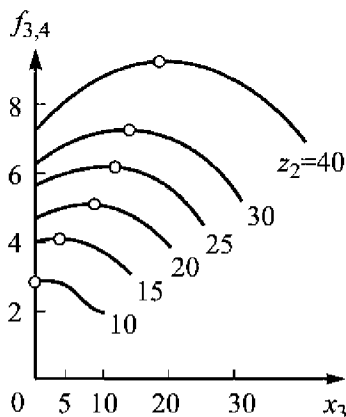


Рис. 4.10. Зависимости $f_{3,4}$ от аргумента x_3 и параметра z_2

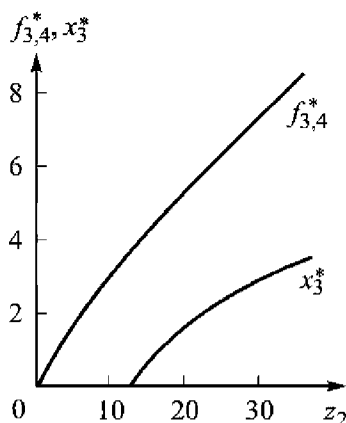


Рис. 4.11. Зависимости $f_{3,4}^*$ и x_3^* от аргумента z_2

Здесь

$$Q_2(x_2) = N_2 \left(1 - e^{(-\bar{v}_2 / N_2) P_2} \right), \quad \bar{v}_2 = x_2 e^{-\alpha_2 N_2},$$

$$z_2 = (z_1 - x_2) e^{-\alpha_2 \bar{N}_2}, \quad \bar{N}_2 = N_2 - Q_2(x_2).$$

По значениям $f_{3,4}^*(z_2)$, приведенным на рис. 4.11, строим зависимости $f_{2,3,4}^*$ и x_2^* от z_1 (рис. 4.13).

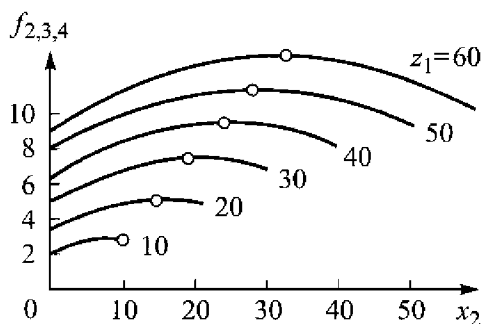


Рис. 4.12. Зависимости $f_{2,3,4}$ от аргумента x_2 и параметра z_1

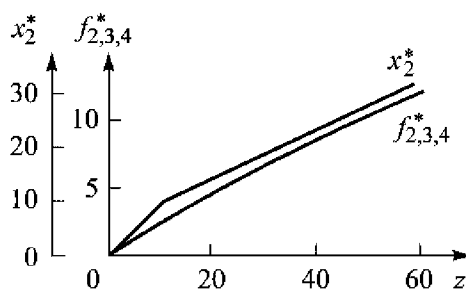


Рис. 4.13. Зависимости $f_{2,3,4}^*$ и x_2^* от аргумента z_1

Проведем оптимизацию решения на первом шаге. Число самолетов, которые совершают налет на первый рубеж, задано: $z_0 = 80$. Выигрыш составит

$$f_{1,2,3,4}(x_1) = Q_1(x_1) + f_{2,3,4}^*(z_1), \quad Q_1(x_1) = N_1 \left(1 - e^{(-\bar{v}_1 / N_1) P_1} \right),$$

где $\bar{v}_1 = x_1 e^{-\alpha_1 N_1}$, $z_1 = (z_0 - x_1) e^{-\alpha_1 \bar{N}_1}$, $\bar{N}_1 = N_1 - Q_1(x_1)$.

По значениям $f_{2,3,4}^*$, приведенным на рис. 4.13, строим график зависимости $f_{1,2,3,4}$ от x_1 при $z_0 = 80$ (рис. 4.14). По точке максимума получаем, что $x_1^* = 34$, а полное число пораженных орудий равно 14,1.

Итак, 34 самолета бомбят первый рубеж, 46 самолетов направлено дальше. К зоне действия орудий второго рубежа дошло z_1^* самолетов:

$$\begin{aligned} z_1^* &= 46e^{-\alpha_1 \bar{N}_1}, \quad \bar{N}_1^* = N_1 - Q_1(x_1^*), \\ Q_1(x_1^*) &= N_1 \left(1 - e^{(-\bar{v}_1 / N_1) P_1} \right), \quad \bar{v}_1^* = x_1^* e^{-\alpha_1 N_1}. \end{aligned} \quad (4.3)$$

Получили $z_1^* = 37$, и на первом рубеже поражено $Q_1(x_1^*) = 5,6$ орудий. При $z_1^* = 37$ по значениям, приведенным на рис. 4.13, получили оптимальное управление на втором шаге: $x_2^* = 23$, т. е. дальше следует направить 14 самолетов. Пользуясь формулами, аналогичными формулам (3), получим

$$\begin{aligned} Q_2(x_1^*) &= 5,4, & z_2^* &= 10,7, & x_3^* &= 0, \\ Q_3(x_3^*) &= 0, & z_3^* &= 5,9, & x_4^* &= 5,9, & Q_4(x_4^*) &= 3,1. \end{aligned}$$

Итак, в первую волну вылетов войдут 34 самолета: $n_1 = 34$. Ко второму рубежу подойдут 37 самолетов, из них полетят дальше только 14, а так как всего самолетов осталось 46, то во вторую волну следует включить $n_2 = 46(14/23) = 28$ самолетов, $n_3 = 0$ (так как $x_3^* = 0$) и в четвертую волну — оставшиеся 18 самолетов.

Анализ полученных результатов показывает, что эффективнее осуществлять планирование на подступах к каждому рубежу, т. е. каждый раз решать задачу, аналогичную рассмотренной.

Продолжая решение, убеждаемся, что эта задача является вырожденной задачей динамического программирования — планировать нужно каждый шаг отдельно, распределяя самолеты перед рубежом так, чтобы получать максимальное число самолетов, преодолевающих данный рубеж.

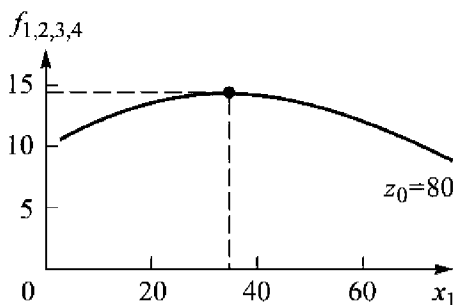


Рис. 4.14. Зависимости $f_{1,2,3,4}$ от аргумента x_1

§ 4.5. Вычислительные аспекты решения задач методом динамического программирования

Мы рассмотрели алгоритм решения задач, опирающийся на основное функциональное уравнение метода динамического программирования

$$f_{\max}(S_0) = \max_{x_1} \{f_1(S_0, x_1) + f_{\max}(S_{m-1}(S_0, x_1))\}, \quad m \geq 1. \quad (4.4)$$

Если $f_1(S_0, x_1)$ известно, то уравнение (4) представляет типичное рекуррентное соотношение. Чтобы вычислить $f_{\max}(S_0)$, необходимо в памяти компьютера иметь значения функций $f_1(S_0, x_1), \dots, f_{m-1}(S_{m-2}, x_{m-1}), S_{m-1}(S_0, x_1)$ или алгоритмы для образования этих значений.

Поскольку невозможно запомнить все значения функции $f_{m-1}(S_{m-2}, x_{m-1})$, запоминают лишь дискретный набор значений $f_{m-1}((S_{m-2}, x_{m-1})_i)$. Другие значения могут быть получены интерполяцией и экстраполяцией записанных данных. Чтобы найти максимум целевой функции по x_1 , рассмотрим дискретный набор значений $\{x_1\}_j, j = 1, 2, \dots$. Одно или несколько значений x_1 , дающих максимум, также запомним.

Повторяя эту процедуру на каждом шаге, получаем функцию дохода и функцию стратегии для данного шага. Рекуррентно вычислим последовательности $\{f_i(S_{i-1}, x_i)\}$ и $\{x_i(S_{i-1})\}$. Эта процедура является итеративной, что облегчает ее программирование. Отметим, что как только функция $f_1(S_0, x_1)$ использована для вычисления $f_2(S_1, x_2)$ и $x_2(S_1)$, ее можно стереть из памяти, и т. д. Чем меньше область возможных значений x_i , тем меньше требуется объем памяти, тем легче применить метод динамического программирования.

Для определенности рассмотрим случай, когда S_0 — вектор размерности 2, $S_0 = f(x, y)$, и мы запоминаем значения S_0 в 100 точках по x и по y , т. е. имеем 10^4 чисел. Этот случай трудностей не вызывает. Если размерность вектора более высокая, скажем 4, то получим 10^8 чисел.

Для изучения таких процессов на современных компьютерах применяют более тонкие алгоритмы. К примеру, вместо простого запоминания набора значений $\{f_i(S_{i-1}, x_i)\}, i = 1, 2, \dots$, в памяти хранят метод воспроизведения функции $f_i(S_{i-1}, x_i)$. Наиболее

простой прием такого воспроизведения функции — это представление ее в виде полинома.

Методы динамического программирования эффективнее применять к решению задач с дискретными переменными, чем с непрерывными. Основное функциональное уравнение имеет один и тот же вид и для дискретных, и для непрерывных переменных. В отдельных случаях при наличии непрерывных переменных удастся сократить расчеты, используя дифференциальное исчисление; однако в большинстве случаев решаемая задача сводится к задаче с дискретными переменными.

Метод динамического программирования позволяет найти все оптимальные решения, сколько бы их ни было, для любой *аддитивной* функции многих переменных, которая определена на любом множестве D , заданном условиями-ограничениями.

§ 4.6. Теория игр. Игры в чистых стратегиях

На практике часто имеют место ситуации, когда нашим целям противостоят цели противника. Подобные ситуации называются *конфликтными*. Примеры подобных ситуаций можно встретить в экономике, военных задачах, в разрешении политических проблем и т. п. Принятие решений каждой из сторон связано с *преодолением конфликта* и затруднено в силу *неопределенности поведения* противника.

Естественно, что противник стремится предпринять наименее выгодные для нас действия, чтобы обеспечить себе наибольший успех. Мы не можем точно предсказать действия противника, равно как и он не может точно предсказать наши действия. И, тем не менее, как нам, так и ему приходится принимать вполне определенные решения.

Необходимость обосновывать *оптимальные* решения, принимаемые в тех или иных конфликтных ситуациях, привела к возникновению специального направления в современной математике — *теории игр*. Под термином «игра» здесь понимается упрощенная математическая модель рассматриваемой конфликтной ситуации. В отличие от реального конфликта игра ведется по *определенным правилам*, которые четко определяют права и обязанности участников игры, объем информации каждой стороны о другой, а также исход игры (выигрыш и проигрыш каждого участника).

Задача теории игр состоит в выработке игроками стратегии, которая обеспечит одной стороне максимальный выигрыш, а другой — минимальный проигрыш. Теория игр используется в конфликтных ситуациях, которые повторяются многократно.

Задолго до появления теории игр широко использовались подобные упрощенные модели конфликтов — игры в буквальном смысле слова: шахматы, шашки, домино, карточные игры и т. д. Отсюда происходит как название самой теории, так и различные термины, используемые в ней. Так, конфликтующие стороны называют *игроками*, одну реализацию игры — *партией*, выбор игроком того или иного действия из возможных (в пределах правил) — *ходом* и т. д. Различают два вида ходов — личные и случайные. *Личный* ход предполагает сознательный выбор игроком того или иного действия, разрешенного правилами игры. *Случайный* ход не зависит от воли игрока — он может быть определен с помощью датчика случайных чисел.

Игры, состоящие только из случайных ходов, называют *азартными*. Характерный пример — игра в лото. Игры, в которых имеются личные ходы, называются *стратегическими*. Существуют стратегические игры, состоящие только из личных ходов (например, шахматы). Существуют также стратегические игры, состоящие как из личных, так и из случайных ходов (например, карточные игры). Заметим, что в играх с личными и случайными ходами неопределенность имеет два вида: неопределенность результата случайных ходов и неопределенность поведения противника в его личных ходах.

В теории игр не исследуют азартные игры, занимаются только стратегическими играми. Задача теории игр — определить такую стратегию игрока, при которой его шансы на выигрыш оказались бы наибольшими. В основе поиска оптимальных стратегий лежит следующее основное положение: считается, что противник также разумен и активен, как и сам игрок, и предпринимает все меры для того, чтобы достичь успеха.

Разумеется, на практике это не всегда выполняется. Часто наши действия в реальном конфликте оказываются оптимальными не тогда, когда мы исходим из наиболее разумного поведения противника, а тогда, когда удастся предвидеть, в чем противник может ошибаться, и воспользоваться его «глупостью». При этом мы рискуем. Известно, как рискованно рассчитывать на «глупость» противника. Теория игр не учитывает элементы риска. Она выявляет лишь наиболее осторожные, «перестраховочные» вари-

анты поведения в данной ситуации. Риски учитываются в другом разделе математики — в *теории статистических решений*. Можно сказать, что теория игр дает нам мудрые советы. Учитывая эти советы, мы принимаем на практике те или иные решения, часто выбирая сознательно некоторый риск. Теория игр ценна прежде всего самой постановкой задач, которая учит не забывать о том, что противник тоже мыслит, и учитывать его возможные хитрости и уловки. Пусть рекомендации, вытекающие из игрового подхода, не всегда определены и не всегда осуществимы, однако полезно, выбирая решение, ориентироваться, в числе других рекомендаций, и на игровую модель. Не стоит только выводы, вытекающие из этой модели, считать окончательными и непререкаемыми.

В теории игр наиболее исследованы *конечные парные игры с нулевой суммой*. Игра называется *парной*, если в ней участвуют два игрока. Участников игр может быть более двух, и они могут быть объединены в группы. Тогда игра называется *коалиционной*. Игра называется *конечной*, если каждый игрок имеет конечное число стратегий, т. е. конечное число вариантов поведения, и *бесконечной*, если хотя бы один из игроков имеет бесконечное число стратегий. Делая личный ход, игрок следует одной из стратегий. Игрой с *нулевой суммой* является игра, в которой выигрыш одного игрока равен проигрышу другого.

Оптимальная стратегия игрока — это такая стратегия, которая при многократном повторении игры обеспечивает игроку максимально возможный средний выигрыш, а другому — минимальный средний проигрыш. Предположим, что рассматривается некоторая конечная парная игра с нулевой суммой, когда игрок A имеет m стратегий, а игрок B имеет n стратегий (игра $m \times n$). Обозначим стратегии игрока A через A_1, A_2, \dots, A_m , а стратегии игрока B через B_1, B_2, \dots, B_n . Пусть игрок A , делая личный ход, выбирает некоторую стратегию A_i , $i = 1, 2, \dots, m$, а игрок B — стратегию B_j , $j = 1, 2, \dots, n$. Обозначим через a_{ij} реализуемый в этом случае выигрыш игрока A . Для определенности мы будем отождествлять себя с игроком A и рассматривать каждый ход с позиции выигрыша игрока A . При этом под выигрышем a_{ij} можно понимать как действительный выигрыш, так и проигрыш (например, проигрыш можно рассматривать как отрицательный выигрыш). Набор выигрышей a_{ij} для разных значений i и j располагают в виде матрицы, строки которой отвечают стратегиям игрока A , а столбцы — стратегиям игрока B (табл. 4.5). Это есть *платежная матрица* игры.

Принцип минимакса

Таким образом, задать игру — это задать m стратегий игрока A , n стратегий игрока B и для пар (A_i, B_j) — задать выигрыш a_{ij} . Все элементы a_{ij} можно увеличить на одно и то же число, чтобы все они стали неотрицательными. Полученная при этом матрица будет эквивалентна исходной. Выбор оптимальной стратегии для игрока A — это выбор стратегии A_i , которая обеспечивает ему максимально возможный выигрыш, независимо от стратегии игрока B . Игрок A предполагает, что минимальный выигрыш, который он получит при стратегии A_i , — это минимальное число в i -й строке. Игрок B отвечает стратегией, которая минимизирует его собственный проигрыш. С точки зрения игрока A , ему необходимо выбрать такую стратегию A_i , которая *максимизирует* минимально возможный выигрыш $\alpha = \max_i \min_j a_{ij}$ — *максиминная стратегия* (*максимин*). Величину α называют еще *нижней ценой игры*. Меньше, чем α , игрок A выиграть не может. Для игрока B оптимальной стратегией является та, при которой максимально возможный выигрыш игрока A оказывается наименьшим и в любом случае не превосходит $\beta = \min_j \max_i a_{ij}$. Величина β называется *минимаксом*, или *верхней ценой игры*. Все параметры игры заносятся в платежную матрицу игры (см. табл. 4.5).

Таблица 4.5

Платежная матрица игры

Стратегия	B_1	B_2	...	B_n	$\alpha_i = \min_j a_{ij}$
A_1	a_{11}	a_{12}	...	a_{1n}	α_1
A_2	a_{21}	a_{22}	...	a_{2n}	α_2
...
A_m	a_{m1}	a_{m2}	...	a_{mn}	α_m
$\beta_j = \max_i a_{ij}$	β_1	β_2	...	β_n	

Максиминная стратегия игрока A так же, как и минимаксная стратегия игрока B , является наиболее осторожной, перестрахо-

вочной стратегией, и она гарантируют игроку B , что максимально возможный выигрыш игрока A оказывается наименьшим и в любом случае не превосходит минимакса, или, иначе, верхней цены игры. Точно так же при любом поведении игрока B игроку A гарантирован минимально возможный выигрыш (наибольший по сравнению с остальными стратегиями) не меньше нижней цены игры (максимина). Принцип осторожности, диктующий игрокам выбор таких стратегий, называется *принципом минимакса*.

Пример. Рассмотрим следующую игру. Игроки A и B одновременно и независимо друг от друга записывают одно из трех чисел: либо 1, либо 2, либо 3. Если сумма записанных чисел оказывается *четной*, то игрок B платит игроку A эту сумму; если же сумма чисел оказывается *нечетной*, то эту сумму выплачивает игрок A игроку B . Игрок A имеет три стратегии: A_1 — записать 1, A_2 — записать 2, A_3 — записать 3. Стратегии игрока B аналогичны. Рассматриваемая игра является игрой размеренностью 3×3 , ее платежная матрица имеет 3 строки и 3 столбца. Эта матрица представлена в табл. 4.6.

Заметим, что выигрыш игрока A , равный, например, — 3, означает в действительности его проигрыш, так как в этом случае игрок A выплачивает 3 усл. ед. игроку B .

В матрице (табл. 4.6) одни элементы являются положительными, а другие отрицательными. Чтобы все элементы платежной матрицы были положительными, увеличим каждый элемент рассматриваемой матрицы на одно и то же число, например на 6. Получим матрицу, представленную в табл. 4.7. С точки зрения анализа оптимальных стратегий эта матрица эквивалентна исходной, но верхняя и нижняя цена игры увеличится на 6. Будем анализировать игру, применяя принцип минимакса и используя платежную матрицу, представленную в табл. 4.7.

Предположим, что игрок A выбирает стратегию A_1 . Тогда в зависимости от того, какую стратегию выберет противник, наш выигрыш будет равен либо 8, либо 3, либо 10 усл. ед. Итак, выбирая стратегию A_1 , мы в худшем случае получаем выигрыш 3 усл. ед. Если же мы выбираем стратегию A_2 или стратегию A_3 , то будем иметь в худшем случае выигрыш 1 усл. ед. Запишем минимально возможные выигрыши для разных стратегий A_i в крайнем правом столбце платежной матрицы (см. табл. 4.7). Ясно, что следует выбирать ту стратегию, где минимально возможный

выигрыш оказывается наибольшим (по сравнению с остальными стратегиями). В данном случае это есть стратегия A_1 . Выигрыш 3 усл. ед. является максимальным в тройке минимальных выигрышей (в тройке 3, 1, 1). Получим максиминный выигрыш (максимин) — нижнюю цену игры.

Таблица 4.6

Исходная платежная матрица

Стратегия	B_1	B_2	B_3
A_1	2	-3	4
A_2	-3	4	-5
A_3	4	-5	6

Таблица 4.7

Редуцированная платежная матрица

Стратегия	B_1	B_2	B_3	α_i
A_1	8	3	10	3
A_2	3	10	1	1
A_3	10	1	12	1
β_j	10	10	12	

Итак, если выбираем максиминную стратегию (в данном случае это стратегия A_1), то при любом поведении противника нам гарантирован выигрыш не меньше нижней цены игры (в данном случае этот выигрыш равен 3 усл. ед.). Аналогичным образом рассуждает противник. Если он выберет стратегию B_1 , то в худшем для себя случае позволит нам получить выигрыш 10 усл. ед. Это же относится и к выбору стратегии B_2 . При выборе стратегии B_3 худший (для противника) случай соответствует нашему выигрышу, равному 12 усл. ед. Числа 10, 10, 12 — максимальные значения наших выигрышей, отвечающие стратегиям противника B_1 , B_2 , B_3 соответственно. Выпишем эти значения в нижнюю строку платежной матрицы (см. табл. 4.7). Ясно, что противник должен выбрать ту стратегию, при которой максимально возможный выигрыш игрока A оказывается наименьшим. Это либо стратегия B_1 , либо B_2 . Обе стратегии являются минимаксными, обе они дают противнику гарантию того, что выигрыш игрока A в любом случае не превысит минимакса, или, иначе, верхней цены игры, равной в данном случае 10 усл. ед. Противник имеет две минимаксные стратегии: B_1 и B_2 . Какую он предпочтет?

Полагая, что мы проявили осторожность и выбрали максиминную стратегию A_1 , он, возможно, не станет выбирать стратегию B_1 , поскольку при этом мы получили бы выигрыш 8 усл. ед.

Значит, скорее всего, он выберет стратегию B_2 , тогда наш выигрыш будет равен 3 усл. ед. Однако если мы правильно поняли замыслы противника, то не следует ли нам рискнуть и выбрать стратегию A_2 ? Ведь при выборе противником стратегии B_2 наша стратегия A_2 позволит получить нам выигрыш 10 усл. ед. Однако наше отступление от принципа минимакса может дорого нам обойтись. Если противник окажется достаточно хитроумным и проведет такие же рассуждения, то он может ответить на нашу стратегию A_2 не стратегией B_2 , а стратегией B_3 . И тогда вместо выигрыша 10 усл. ед. мы получим выигрыш всего лишь 1 усл. ед.

Игра с седловой точкой

Теория игр не всегда рекомендует применять минимаксные (максиминные стратегии). Это зависит от того, имеет ли платежная матрица игры *седловую точку*. Рассмотрим некоторую игру, в которой максиминный и минимаксный выигрыши равны, т. е. нижняя и верхняя цена игры совпадают: $\alpha = \beta$. Выигрыш является одновременно и максимальным из минимальных выигрышей для игрока A , и минимальным из максимальных выигрышей для игрока B . Точку на поверхности, являющуюся одновременно точкой минимума по одной оси координат и точкой максимума по другой оси, называют *седловой точкой*. Соответствующий элемент $a_{ij} = \alpha$ (или $a_{ij} = \beta$) платежной матрицы игры называют *седловой точкой* (*седловым элементом*) матрицы. О такой игре говорят, что она *имеет седловую точку*. В этой ситуации каждый из игроков должен придерживаться максиминной (минимаксной) стратегии. Любое отклонение от этой стратегии будет невыгодно для игрока, допустившего отклонение. Если игра не имеет седловой точки (см. табл. 4.7), то ни одна из стратегий A_i или B_j не является оптимальной. Рассмотренные стратегии игроков A и B называют *чистыми*. Чистые стратегии не дают устойчивости решения.

Пример. Рассмотрим некоторую матричную игру размерностью 3×3 , платежная матрица которой приведена в табл. 4.8. Здесь максиминный и минимаксный выигрыши равны 4 (верхняя и нижняя цена игры совпадают). Выигрыш 4 является одновременно и максимальным из минимальных выигрышей для стратегий A_1 , A_2 , A_3 , и минимальным из максимальных выигрышей для стратегий B_1 , B_2 , B_3 . Элемент $a_{22} = 4$ — седловая точка матрицы. Ис-

следуя данную матрицу, становится понятно, что любое отклонение от седловой точки не выгодно для игроков. Минимаксная (максиминная) стратегия является оптимальной в данном случае.

Таблица 4.8

Платежная матрица с седловой точкой

Стратегия	B_1	B_2	B_3	α_i
A_1	2	3	7	2
A_2	5	4	6	4
A_3	6	2	1	1
β_j	6	4	7	

§ 4.7. Поиск оптимальной смешанной стратегии

Теория игр используется для исследования *многократно* повторяющихся конфликтных ситуаций. Если игроки будут от игры к игре придерживаться одной и той же стратегии, рекомендуемой принципом минимакса, то один из них может оказаться в наихудшей ситуации при внезапной смене стратегии другим игроком. Здесь проявляется общее правило для игр без седловой точки: игрок, играющий по определенной (детерминированной) стратегии, оказывается в более худшем положении по сравнению с игроком, который меняет стратегию случайным образом.

Пусть игроки A и B многократно играют в игру, платежная матрица которой приведена в табл. 4.7. Если игрок A выберет определенную стратегию, например максиминную стратегию A_1 , и будет придерживаться ее от игры к игре, то игрок B , поняв это, будет выбирать каждый раз стратегию B_2 , в результате чего выигрыш игрока A будет равен 3 усл. ед. Однако если игрок A внезапно сменит стратегию A_1 на стратегию A_2 , то получит выигрыш 10 усл. ед. Разгадав стратегию игрока A , игрок B сменит стратегию B_2 на стратегию B_3 , уменьшив выигрыш игрока A до 1 усл. ед.

Рассмотрим общий подход к разрешению данной проблемы. Пусть A_1, A_2, \dots, A_m — возможные стратегии игрока A . Для получения наибольшего эффекта игрок A должен использовать все или некоторые из этих стратегий случайным образом с разными вероятностями: стратегия A_i используется с вероятностью p_i . В этом

случае говорят, что игрок A применяет *смешанную стратегию* $S_A(p_1, p_2, \dots, p_m)$. В отличие от смешанных стратегий S_A стратегии A_i , как уже указывалось, называют чистыми. При надлежащем выборе вероятностей p_i смешанная стратегия может оказаться *оптимальной* (аналог седловой точки). При этом выигрыш игрока A будет не меньше некоторого значения v , называемого *ценой игры*. Это значение в принципе минимакса больше нижней цены игры, но меньше верхней.

Аналогичным образом должен вести себя и игрок B . Его оптимальной стратегией является $S_B(q_1, q_2, \dots, q_n)$, где q_j — специально подобранные вероятности, с которыми игрок B использует стратегии B_j , $j = 1, 2, \dots, n$.

Чтобы получить оптимальную смешанную стратегию, требуется найти соответствующие вероятности $p_i, i = 1, 2, \dots, m$, и $q_j, j = 1, 2, \dots, n$, а также определить цену игры v . Допустим, что игрок B выбирает чистую стратегию B_j . Тогда средний выигрыш игрока A будет равен $\sum_i a_{ij} p_i$. Этот выигрыш должен быть не меньше цены игры v , т. е.

$$\sum_{i=1}^m a_{ij} p_i \geq v, \quad j = 1, 2, \dots, n. \quad (4.5)$$

При этом

$$\sum_{i=1}^m p_i = 1. \quad (4.6)$$

Введем обозначения $x_i = p_i/v$, $i = 1, 2, \dots, m$, и перепишем соотношения (4.5), (4.6) в виде

$$\sum_{i=1}^m a_{ij} x_i \geq 1, \quad j = 1, 2, \dots, n, \quad (4.7)$$

$$\sum_{i=1}^m x_i = \frac{1}{v}. \quad (4.8)$$

Цена игры v должна быть как можно больше, следовательно, величина $1/v$ должна быть как можно меньше. Таким образом, поиск оптимальной смешанной стратегии свелся к решению задачи линейного программирования: найти неотрицательные величины x_i , $i = 1, 2, \dots, m$, такие, чтобы они удовлетворяли неравенст-

вам (4.7), (4.8) и при этом обращали в минимум сумму (целевую функцию) $\sum_{i=1}^m x_i$. Сформулируем следующую теорему.

Теорема. *Чтобы стратегия (p^*, q^*) была оптимальной, необходимо и достаточно выполнения неравенств*

$$\sum_{i=1}^m a_{ij} p_i^* \geq v, \quad j=1, 2, \dots, n,$$

$$\sum_{j=1}^n a_{ij} q_j^* \leq v, \quad i=1, 2, \dots, m,$$

$$\text{при } \sum_{i=1}^m p_i = 1, \quad \sum_{j=1}^n q_j = 1.$$

Чистая стратегия, которая входит в смешанную с вероятностью больше нуля, называется *активной*.

Теорема. *Если один из игроков придерживается своей оптимальной стратегии, то его выигрыш остается неизменным и равным цене игры независимо от стратегии другого игрока, если только он не выходит за пределы активных стратегий.*

§ 4.8. Решение матричных игр размерностью $m \times n$

В процедуре решения матричных игр можно выделить следующие этапы.

I. Необходимо проверить, существует ли седловая точка в платежной матрице игры. Если седловая точка есть, то оптимальное решение достигается в чистых стратегиях: решением игры является тройка чисел (A_i^*, B_j^*, v) .

II. Если седловой точки нет, то оптимальное решение достигается в смешанных стратегиях. Предварительно следует попытаться упростить задачу.

1. Убрать дублирующие и доминируемые стратегии. Если в платежной матрице для двух строк элементы a_{kj} не меньше элементов a_{sj} (т. е. $a_{kj} \geq a_{sj}$) и хотя бы один элемент k -й строки строго больше соответствующего элемента s -й строки, то для игрока A стратегия A_k предпочтительнее стратегии A_s . Стратегия A_k называется *доминирующей*, стратегия A_s — *доминируемой*, ее можно удалить (в смешанной стратегии получим вероятность, равную нулю). Доминируемая стратегия *заведомо невыгодная*.

Аналогично для столбцов: если соответствующие элементы l -го и r -го столбцов удовлетворяют неравенству $a_{il} \leq a_{ir}$ и хотя бы один элемент l -го столбца строго меньше соответствующего элемента r -го столбца, то стратегия B_l является доминирующей и r -й столбец исключаем. Столбцы и строки, соответствующие элементам которых равны, называются *дублирующими*, и в платежной матрице игры следует оставить только одну строку или один столбец из дублирующих.

2. Решить, нельзя ли уменьшить число стратегий путем замены групп чистых стратегий смешанными.

3. Упростить платежную матрицу, добавив ко всем элементам матрицы некоторое число. При этом цена игры увеличится на это число, а оптимальная стратегия не изменится.

III. Если платежная матрица игры имеет размерность $2 \times n$ или $m \times 2$, то используются методы решения матричной игры размерностью 2×2 .

По условию теоремы о минимаксе любая матричная игра размерностью $m \times n$ имеет оптимальное решение, принадлежащее области смешанных стратегий. Оптимальные стратегии (p^*, q^*) образуют седловую точку платежной матрицы и также являются минимаксными. Решить матричную игру — это значит получить оптимальное решение (p^*, q^*, v) .

Алгоритм решения определяется необходимым и достаточным условиями оптимальности.

Шаг 1. Найдем оптимальную стратегию для игрока B : вероятности гипотез равны $q = (q_1, q_2, \dots, q_n)$, заданы ограничения

$\sum_{j=1}^n a_{ij} q_j \leq v, \quad i = 1, 2, \dots, m$. Применяя такую стратегию, игрок B не

проиграет больше v при любой чистой стратегии игрока A , при оптимальной стратегии игрока A проигрыш игрока B равен v . Сформулируем данную задачу как задачу линейного программирования: введем переменные $y_j = \frac{q_j}{v}$, $\sum_{j=1}^n y_j = \frac{1}{v}$, т. е. получим задачу

$$\sum_{j=1}^n y_j \rightarrow \max;$$

$$\sum_{j=1}^n a_{ij} y_j \leq 1, \quad i = 1, 2, \dots, m, \quad y_j \geq 0, \quad j = 1, 2, \dots, n.$$

Применяя симплекс-метод, найдем значения y^* , $v = \left(\sum_{j=1}^n y_j^* \right)^{-1}$.

Шаг 2. Определим оптимальную стратегию игрока A . Эта стратегия должна обеспечить ему выигрыш не менее v при любой стратегии игрока B и равный v при оптимальной стратегии игрока B . Из необходимого и достаточного условий следует

$$\sum_{i=1}^m a_{ij} p_i \geq v, \quad j = 1, 2, \dots, n,$$

$$\sum_{i=1}^m p_i = 1, \quad p_i \geq 0, \quad i = 1, 2, \dots, m.$$

Здесь p_i — вероятность стратегии A_i . Введем новые переменные $x_i = p_i/v$ и перейдем к задаче ЛП:

$$\sum_{i=1}^m x_i \rightarrow \min;$$

$$\sum_{i=1}^m a_{ij} x_i \geq 1, \quad j = 1, 2, \dots, n, \quad x_i \geq 0, \quad i = 1, 2, \dots, m.$$

Решаем эту задачу также симплекс-методом.

Таким образом, матричные игры размерностью $m \times n$ сводятся к решению задач линейного программирования. В матричных играх существование решения задач ЛП обеспечено. Конкретные условия матричной игры определяют выбор метода решения задач ЛП.

Пример. Рассмотрим игру с платежной матрицей

$$M = \begin{pmatrix} 4 & 2 & 2 \\ 2 & 5 & 0 \\ 0 & 2 & 5 \end{pmatrix}.$$

Для игрока B имеем задачу ЛП: найти максимум целевой функции

$$f(y) = y_1 + y_2 + y_3$$

при условиях

$$4y_1 + 2y_2 + 2y_3 \leq 1,$$

$$2y_1 + 5y_2 \leq 1,$$

$$2y_2 + 5y_3 \leq 1,$$

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

Решением этой задачи является точка $y^* = \left(\frac{9}{88}, \frac{14}{88}, \frac{12}{88} \right)$. Отсюда

$$y_{\max} = \sum_{j=1}^3 y_j^* = \frac{35}{88}, \quad v = \frac{1}{y_{\max}} = \frac{88}{35},$$

$$q^* = \left(\frac{9}{35}, \frac{14}{35}, \frac{12}{35} \right).$$

Для игрока A имеем задачу ЛП: найти минимум целевой функции

$$f(x) = x_1 + x_2 + x_3$$

при ограничениях

$$\begin{aligned} 4x_1 + 2x_2 &\geq 1, \\ 2x_1 + 5x_2 + 2x_3 &\geq 1, \\ 2x_1 + 5x_3 &\geq 1, \\ x_1 \geq 0, \quad x_2 &\geq 0, \quad x_3 \geq 0. \end{aligned}$$

Решением этой задачи является точка $p^* = \left(\frac{19}{35}, \frac{6}{35}, \frac{10}{35} \right)$.

В рассмотренном примере все три стратегии игроков оказались активными. Игра, в которой все стратегии активны, называется *полностью усредненной*.

Общий метод решения матричных игр размерностью $m \times n$ — сведение их к задачам ЛП — не всегда оказывается самым простым. В ряде случаев, особенно при малой размерности задач, удастся решить игру более простыми методами, если заранее определить, какие стратегии являются активными. Например, для случая полностью усредненной игры с квадратной матрицей ($m = n$) неравенства (4.7) преобразуются в равенства, т. е. получаем системы m линейных уравнений с m неизвестными x_i или y_i соответственно, $i = 1, 2, \dots, m$. Решая их, найдем положительные значения x_i или y_i , цену игры v и вероятности p_i в оптимальной смешанной стратегии S_A^* : $p_i = vx_i$, $i = 1, 2, \dots, m$. Так, в данном примере, заменив неравенства на равенства, можно непосредственно получить $2q_1^* + 5q_2^* = v$.

В смешанных стратегиях матричных игр размерностью $m \times n$ могут присутствовать стратегии с вероятностью, равной нулю.

Число активных стратегий в таком случае не превосходит $\min\{m; n\}$. Оптимальная стратегия находится с помощью решения соответствующей системы уравнений. Пусть вектор p содержит r вероятностей, отличных от нуля, а вектор q содержит s таких вероятностей; тогда

$$\sum_{i=1}^r p_i = \sum_{j=1}^s q_j, \quad \sum_{i=1}^m a_{ij} p_i^* \geq v,$$

$$v = \sum_{i=1}^r \sum_{j=1}^s a_{ij} p_i^* q_j^* = \sum_{j=1}^s q_j^* \sum_{i=1}^r a_{ij} p_i^* \geq \sum_{j=1}^s q_j^* v.$$

Отсюда следует, что если игрок не выходит за пределы активных стратегий, то для получения оптимальной стратегии достаточно решить приведенную систему уравнений.

Наиболее просто смешанная оптимальная стратегия находится для игры с платежной матрицей размерностью 2×2 . Рассмотрим платежную матрицу (табл. 4.9), в которой нет седловой точки. Если седловая точка существует, то решение очевидно. Определим оптимальную смешанную стратегию для данного случая, т. е. найдем $S_A^* = (p_1^*, p_2^*)$, $S_B^* = (q_1^*, q_2^*)$ и цену игры v .

Таблица 4.9

Платежная матрица

Стратегия	B_1	B_2
A_1	2	3
A_2	5	4

Пусть игрок B использует стратегию B_1 , тогда цена игры будет $v = a_{11}p_1 + a_{21}p_2$; для стратегии B_2 имеем $v = a_{12}p_1 + a_{22}p_2$, $p_1 + p_2 = 1$. Для стратегий A_1 и A_2 соответственно имеем $v = a_{11}q_1 + a_{12}q_2$, $v = a_{21}q_1 + a_{22}q_2$, $q_1 + q_2 = 1$. Решив эти системы уравнений, получим

$$p_1^* = \frac{a_{22} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}, \quad p_2^* = \frac{a_{11} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}},$$

$$q_1^* = \frac{a_{22} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}}, \quad q_2^* = \frac{a_{11} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}},$$

$$v = \frac{a_{22}a_{11} - a_{12}a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}.$$

Решению матричных игр размерностями 2×2 , $2 \times n$, $m \times 2$ можно дать геометрическую интерпретацию. Рассмотрим решение игры с платежной матрицей размерностью $2 \times n$, т. е.

$$M = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \end{pmatrix}.$$

Цена игры с позиции игрока A есть

$$v(p) = \min_j \{a_{1j}p_1 + a_{2j}p_2\}, \quad p_1 = 1 - p_2,$$

$$v(p_2) = \min_j \{(a_{2j} - a_{1j})p_2 + a_{1j}\}.$$

Задача игрока A — максимизировать функцию

$$v(p) = \min\{(a_{21} - a_{11})p_2 + a_{11}; (a_{22} - a_{12})p_2 + a_{12}\}.$$

При $p_2 = 0$ имеем $v(p_2) = \min_j a_{1j}$, при $p_2 = 1$ имеем $v(p_2) = \min_j a_{2j}$, $j = 1, 2, \dots, n$.

Диаграмма для цены игры размерностью 2×2 с позиции игрока A представлена на рис. 4.15.

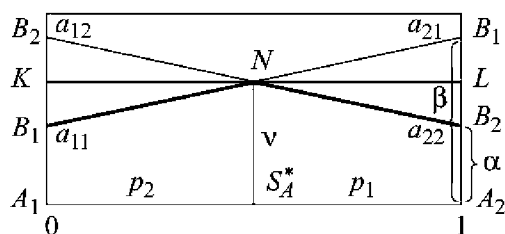


Рис. 4.15. Диаграмма матричной игры размерностью 2×2 с позиции игрока A

По оси абсцисс отложим отрезок длиной, равной единице. Левый край с координатой, равной нулю, будет отражать стратегию A_1 , правый — стратегию A_2 . На вертикальных линиях, выходящих из нуля и единицы, будем откладывать выигрыши игрока A при стратегиях B_1 и B_2 игрока B , т. е. на левой линии отложим значения a_{11} и a_{12} , на правой — a_{21} и a_{22} . Точки на вертикальных линиях, соответствующие одной из стратегий B_j , $j = 1, 2$, соединим прямыми линиями — линиями выигрыша.

В большинстве случаев эти прямые пересекаются в точке N . Абсцисса точки N делит единичный отрезок пропорционально вероятностям p_1 и p_2 , а ордината точки N — это цена игры v . На диаграмме отмечены α и β — нижняя и верхняя цены игры. Поскольку стратегия S_A^* требует, чтобы минимальный выигрыш игрока A обращался в максимум, то строим *нижнюю* границу выигрыша, отмеченную жирной линией. Линии выигрыша могут не пересекаться. Это значит, что игрок B имеет заведомо невыгодную стратегию. Однако если игрок A имеет одну заведомо невыгодную стратегию, то формальный анализ пересечения линий выигрыша может привести к ошибке. Чтобы избежать ошибки, необходимо предварительно исключить заведомо невыгодные стратегии игрока A .

Для игрока B цена игры определяется следующим образом:

$$v(q_1) = \max_i \{a_{i1}q_1 + a_{i2}q_2\}, \quad q_1 = 1 - q_2,$$

$$v(q_2) = \max_i \{(a_{i2} - a_{i1})q_2 + a_{i1}\}.$$

Игроку B необходимо минимизировать величину

$$v(q) = \max\{(a_{12} - a_{11})q_2 + a_{11}; (a_{22} - a_{21})q_2 + a_{21}\}.$$

Оптимальную стратегию игрока B можно определить по диаграмме, приведенной на рис. 4.15:

$$q_1 = \frac{KB_1}{KB_1 + KB_2}, \quad q_2 = \frac{LB_2}{LB_1 + LB_2}.$$

Однако проще это сделать непосредственно, построив для игрока B аналогичную диаграмму и отметив на ней минимум *верхней* границы выигрыша (рис. 4.16).

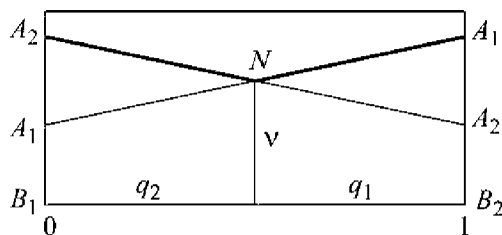


Рис. 4.16. Диаграмма матричной игры размерностью 2×2 с позиции игрока B

Поскольку любая конечная игра размерностью $m \times n$ имеет решение, в котором число активных стратегий каждой стороны не превосходит наименьшего из чисел m и n , то игра размерностью $2 \times n$ всегда имеет решение, при котором с каждой стороны участвует не более двух активных стратегий. Поэтому для решения игры размерностью $2 \times n$ строится геометрическая интерпретация игры и ищется пара стратегий, пересекающихся в точке N (рис. 4.17). Если в точке N пересекаются больше двух стратегий, то берется из них любая пара и ищется максимум на *нижней* границе выигрыша.

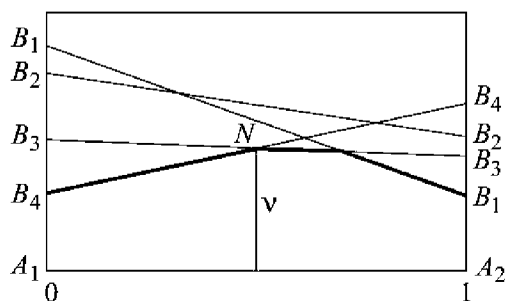


Рис. 4.17. Диаграмма матричной игры размерностью $2 \times n$

Для игры размерностью $m \times 2$ строится такая же диаграмма, в которой ищется минимум на *верхней* границе выигрыша.

Пример. Найдем оптимальную смешанную стратегию для конкретной игры. Предположим, что сторона A нападает на сторону B . Сторона A имеет два самолета, несущие поражающие средства. Сторона B имеет четыре зенитки, защищающие объект. Чтобы объект оказался разрушенным, достаточно, чтобы к нему прорвался хотя бы один самолет. Для подхода к объекту самолеты могут выбрать любой из четырех воздушных коридоров. Сторона A может послать оба самолета в одном и том же коридоре или направить их по разным коридорам. Сторона B может разместить свои четыре зенитки в пределах рассматриваемых коридоров разными способами. Каждая зенитка может произвести только один выстрел. Этот выстрел с вероятностью, равной 1, поражает самолет, если тот оказался в данном коридоре.

Сторона A имеет две чистые стратегии: стратегия A_1 — самолеты посылаются в разные воздушные коридоры (безразлично, какие именно), стратегия A_2 — оба самолета посылаются в какой-то один из коридоров. Возможные стратегии стороны B следующие:

B_1 — поставить по зенитке в каждом коридоре, B_2 — поставить по две зенитки в каких-то двух коридорах (остальные два коридора остаются неохранными), B_3 — поставить две зенитки в одном из коридоров и по одной зенитке еще в двух коридорах, B_4 — поставить три зенитки в одном из коридоров и одну зенитку еще в одном коридоре, B_5 — поставить все четыре зенитки в одном из коридоров. Стратегии B_4 и B_5 заведомо невыгодны хотя бы потому, что три, а тем более четыре зенитки в пределах одного коридора не нужны, поскольку сторона A имеет всего два самолета. Поэтому ограничимся рассмотрением стратегий B_1, B_2, B_3 .

Предположим, что сторона A выбрала стратегию A_1 , а сторона B — стратегию B_1 . Ясно, что тогда ни один самолет не прорвется к объекту — выигрыш стороны A равен нулю ($a_{11} = 0$). Пусть выбраны стратегии A_1 и B_2 . Допустим при этом, что зенитки находятся в коридорах I и II. Самолеты летят в разных коридорах, причем равновероятны шесть вариантов: они летят в коридорах I и II, летят в коридорах I и III, в коридорах I и IV, в II и III, в II и IV, в III и VI. Только в одном из указанных шести случаев ни один из самолетов не прорвется к объекту (когда они летят в коридорах I и II). Какие бы два коридора ни выбирала сторона B для размещения пары зениток, всегда для самолетов существуют шесть равновероятных вариантов, и только один из них проигрышный. Таким образом, при выборе стратегий A_1 и B_2 вероятный выигрыш стороны A составляет $5/6$ ($a_{12} = 5/6$). Рассуждая подобным образом, нетрудно найти остальные элементы платежной матрицы данной игры, которая приведена в табл. 4.10, это есть матрица размерностью 2×3 . Заметим, что элементы матрицы — *вероятностные* выигрыши; здесь уже чистые стратегии включают в себя случайность. Нижняя цена игры равна $1/2$, верхняя равна $3/4$. Максиминной стратегией является A_2 , минимаксной — B_3 . Седловой точки нет, оптимальное решение игры лежит в области смешанных стратегий.

Таблица 4.10

Платежная матрица

Стратегии	B_1	B_2	B_3	α_i
A_1	0	$5/6$	$1/2$	0
A_2	1	$1/2$	$3/4$	$1/2$
β_j	1	$5/6$	$3/4$	

Чтобы найти оптимальную смешанную стратегию, воспользуемся видом платежной матрицы и соотношениями (4.7) и (4.8). В данном случае эти соотношения принимают вид

$$x_2 \geq 1, \quad \frac{5}{6}x_1 + \frac{1}{2}x_2 \geq 1, \quad \frac{1}{2}x_1 + \frac{3}{4}x_2 \geq 1, \quad (4.9)$$

$$x_1 + x_2 = \frac{1}{v}. \quad (4.10)$$

Решением, соответствующим *оптимальной смешанной стратегии*, является $x_1 = 3/5$, $x_2 = 1$. Отсюда находим $v = 5/8$, $p_1 = 3/8$, $p_2 = 5/8$. Итак, оптимальная смешанная стратегия стороны A предполагает использование стратегии A_1 с вероятностью $3/8$ и стратегии A_2 — с вероятностью $5/8$.

Как воспользоваться этой рекомендацией на практике? Если игра происходит один раз, то стороне A следует, по-видимому, избрать стратегию A_2 , так как $p_2 > p_1$. Если данная игра совершается *множественно* (например, по отношению к многим объектам, подлежащим бомбардировке) — N раз ($N \gg 1$), то в $3N/8$ случаях сторона A должна избрать стратегию A_1 , а в $5N/8$ случаях — стратегию A_2 .

При выборе стороной A оптимальной смешанной стратегии ее средний выигрыш оказывается в пределах между верхней ценой игры, равной $3/4$, и ценой игры $v = 5/8$. При неразумном поведении стороны B выигрыш стороны A может оказаться равным верхней цене игры (и даже может стать больше). Если же сторона B , в свою очередь, будет придерживаться оптимальной смешанной стратегии, то выигрыш стороны A окажется равным цене игры v . Оптимальная смешанная стратегия стороны B сводится к тому, что эта сторона вообще не применяет стратегию B_3 , стратегию B_1 использует с вероятностью $1/4$, а стратегию B_2 с вероятностью $3/4$. Нецелесообразность применения стратегии B_3 видна из того, что прямая

$$\frac{1}{2}x_1 + \frac{3}{4}x_2 = 1$$

не принадлежит области допустимых значений D системы (4.9). Для определения вероятностей, с которыми должны применяться

стратегии B_1 и B_2 , воспользуемся уже найденным значением цены игры ($v = 5/8$):

$$q_1 \cdot 0 + (1 - q_1) \frac{5}{6} = \frac{5}{8}.$$

Отсюда видно, что $q_1 = 1/4$, $q_2 = 1 - q_1 = 3/4$.

На практике часто не возникает необходимость находить точное решение игры и решать задачи ЛП большой размерности, достаточно бывает найти приближенное решение, обеспечивающее средний выигрыш, близкий к цене игры. Например, если нижняя и верхняя цены игры (α и β) близки, то достаточно взять чистые минимаксные стратегии; если α и β не близки, то можно воспользоваться *методом итераций*. Здесь один из игроков, например игрок A , выбирает стратегию A_i , игрок B отвечает стратегией B_j , которая наименее выгодна для игрока A , т. е. обращает выигрыш стратегии A_i в минимум. Игрок A отвечает стратегией A_k , которая дает ему максимальный выигрыш при стратегии B_j , и т. д. Этот процесс сходится, но сходимость медленная. Однако в то же время сложность метода практически не возрастает с увеличением размерности платежной матрицы.

Мы рассмотрели антагонистические игры двух лиц, т. е. игры, в которых интересы сторон прямо противоположны. Однако реальные задачи принятия решения в условиях конфликта характеризуются большим числом участников и, как следствие этого, неантагонистичностью конфликтной ситуации. Даже для двух игроков их интересы могут пересекаться, что может приводить к ситуациям, взаимовыгодным обоим игрокам. Это приводит к выбору согласованного решения, приводящего к увеличению выигрыша обоих игроков. Поэтому в неантагонистических играх различают *бескоалиционное поведение*, когда соглашения между игроками запрещены правилами, и *кооперативное поведение* игроков, когда разрешается кооперация: выбор совместных стратегий, совершение побочных платежей.

Обобщением случаев, когда число ходов в игре становится бесконечным и игроки имеют возможность принимать решения непрерывно, занимаются в разделе теории игр, называемом *дифференциальные игры*. Здесь траектории движения игроков представляют собой решения систем обыкновенных дифференциаль-

ных уравнений, правые части которых зависят от параметров, находящихся под контролем игроков. Рассматриваются две системы обыкновенных дифференциальных уравнений

$$\dot{x} = f(x, u), \quad (4.11)$$

$$\dot{y} = g(y, v) \quad (4.12)$$

с начальными условиями (x_0, y_0) .

Игрок A (или B) начинает движение из фазового состояния x_0 (или y_0) и перемещается в фазовом пространстве \mathbb{R}^n согласно (4.11) (или (4.12)), выбирая в каждый момент времени значение параметра $u \in U$ (или $v \in V$) в соответствии со своими целями и информацией, доступной в каждом текущем состоянии.

О РАЗВИТИИ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

§ 5.1. Основные направления развития методов решения задач математического программирования

Можно построить математические модели, описывающие достаточно широкий круг явлений. Однако мы выяснили, что не всякую математическую задачу возможно решить, во-первых, по причине большой ее размерности, во-вторых, из-за наличия нелинейных целевых функций и условий-ограничений. Кроме того, нам хотелось бы, не решая заново задачу, иметь метод, который позволял бы изменить решения при небольших вариациях исходных данных (параметров целевой функции и условий-ограничений). Поскольку такие изменения исходных данных происходят случайно, то следовало бы определить и доверительные границы для решений реальной задачи. Однако тогда мы выйдем за рамки темы настоящей книги, поскольку решение таких задач относится к области *стохастического программирования*.

Тем не менее, удастся, пусть и достаточно громоздкими методами, установить в задачах линейного программирования, как будет изменяться решение задачи, если известны функциональные зависимости, описывающие изменение исходных данных. Подобные задачи решают методами *параметрического программирования*.

Несмотря на то, что разработано много методов решения сетевых задач, трудности в их решении остаются. В гл. 3 мы упоминали, что из-за большой размерности некоторые сетевые задачи строгими методами решить не удастся; в подобных случаях применяют *эвристические методы*, которые позволяют получить пусть и не строго оптимальное решение, но приемлемое в практических приложениях.

В гл. 3 мы рассмотрели сетевые задачи для однородного потока. В реальных условиях по сети требуется транспортировать (передавать) различные «продукты»: например, телефонные разгово-

ры, телеграммы, телевизионные передачи и другое, т. е. необходимо иметь алгоритмы, которые позволяли бы решать задачи о *многопродуктовых потоках* в сетях. Методы решения *многопродуктовых сетевых задач* сложны, и поэтому применяемые алгоритмы позволяют решить задачу только приближенно.

Естественно стремление исследователей создавать универсальные алгоритмы, которые можно было бы применять к достаточно широкому кругу задач математического программирования. К ним относят алгоритмы, реализующие *метод проекции градиента* и *методы штрафных (барьерных) функций: метод внутренней точки, метод внешней точки и комбинированный метод внутренней и внешней точки*. Методами штрафных функций можно решать задачи как линейного, так и нелинейного программирования, хотя вряд ли экономно использовать их для решения линейных задач.

Особое место в практике занимают многокритериальные задачи и задачи целевого программирования, которые более адекватно описывают реальные явления.

§ 5.2. Понятие о параметрическом программировании

Исходные данные, необходимые для численной постановки реальных задач математического программирования, определяют довольно часто неточно, приближенно. Это связано не только с наличием погрешностей измерений, но и с желанием описать в математической постановке задачи возможное изменение исходных данных, чтобы в дальнейшем для оптимизации решения использовать наилучшие их значения.

Исследование таких математических проблем, как анализ изменения решения при вариации исходных данных, оценка устойчивости решения, требует разработки методов, учитывающих неопределенность задания исходных данных.

Как было показано ранее, к настоящему времени наиболее доступны для решения задачи линейного программирования. Алгоритм, учитывающий неопределенности исходных данных, мы рассмотрим на примере решения задач линейного программирования, так как в других случаях это может оказаться крайне сложной проблемой. Раздел математического программирования, в котором может быть решена данная проблема, называют *параметрическим программированием*. В этом разделе изучают в основном задачи,

которые являются естественным обобщением задач линейного программирования, когда исходные данные (коэффициенты) в целевой функции и условиях-ограничениях предполагают не постоянными величинами, а функциями, зависящими определенным образом (чаще всего линейно) от некоторых параметров. Всякой задаче параметрического программирования можно поставить в соответствие некоторую задачу линейного программирования, называемую *исходной*. От того, как именно трактуется исходная задача, зависит трактовка параметрической задачи. Введение параметра обычно отражает некоторую реальную ситуацию.

Приведем несколько задач параметрического программирования, которые наиболее естественным образом могут быть сопоставлены принятой исходной задаче. Рассмотрим, помимо этого, одну из интерпретаций задачи параметрического программирования.

Пусть коэффициенты целевой функции исходной задачи линейного программирования (2.2) зависят от одного параметра. В задаче (2.2) коэффициенты целевой функции представляют собой цены разных продуктов, а координаты векторов-ограничений могут быть истолкованы как запасы различных ресурсов.

Постановка рассматриваемой задачи параметрического программирования может иметь (в простейшем случае) следующий вид:

$$f(x) = (a_{01} + b_1 t)x_1 + \dots + (a_{0n} + b_n t)x_n \rightarrow \max,$$

где a_{01}, \dots, a_{0n} — исходные (старые) коэффициенты задачи линейного программирования, b_1, \dots, b_n — новые коэффициенты, t — параметр, $t \in \mathbb{R}^1$.

Зависимость коэффициентов этой функции от параметра t можно понимать как зависимость цены единицы продукта от времени. Различные новые коэффициенты b_1, \dots, b_n отражают индивидуальный характер зависимости цен разных продуктов от параметра t . Значение целевой функции исходной задачи равно стоимости выпущенной продукции, а значение целевой функции соответствующей параметрической задачи показывает, чему равна стоимость выпускаемой продукции при условии изменения цен разных продуктов, когда закон изменения этих цен (от времени, от качества продукции и т. п.) задан.

Рассмотрим случай, когда от параметра зависят координаты системы ограничений. Условия-ограничения принимают вид

Здесь a_{ij} , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$, — исходные коэффициенты из условий-ограничений задачи, а c_{ij} и d_i — новые коэффициенты, определяющие зависимость исходных коэффициентов от параметра t .

$$f(x) = (a_{01} + b_{11}t_1 + \dots + b_{1s}t_s)x_1 + \dots$$

$$\dots + (a_{0n} + b_{n1}t_1 + \dots + b_{ns}t_s)x_n + e_1t_1 + \dots + e_st_s$$
$$(a_{11} + c_{111}t_1 + \dots + c_{11s}t_s)x_1 + \dots + (a_{1n} + c_{1n1}t_1 + \dots + c_{1ns}t_s)x_n \leq \\ \leq a_{10} + d_{11}t_1 + \dots + d_{1s}t_s,$$

Рассмотрим метод решения конкретной задачи линейного параметрического программирования, в которой все коэффициенты целевой функции линейно зависят от некоторого действительного параметра t :

$$\begin{aligned} f(x) &= (2+t)x_1 + (3-t)x_2 + t \rightarrow \max; \\ -x_1 + 2x_2 &\leq 4, \\ x_1 + x_2 &\leq 5, \\ 2x_1 - x_2 &\leq 8, \\ x_1 \geq 0, \quad x_2 &\geq 0, \quad t \in \mathbb{R}^1. \end{aligned} \tag{5.1}$$

В матрично-векторной форме эта задача записывается следующим образом:

$$f(x) = (a_0 + bt)^T x \rightarrow \max, \quad Ax \leq a_0, \quad x \geq 0, \quad t \in \mathbb{R}^1,$$

где

$$a_0 = (2, 3, 0)^T, \quad b = (1, -1, 1)^T, \quad x = (x_1, x_2, 1)^T, \quad a_0 = (4, 5, 8)^T,$$

$$A = \begin{pmatrix} -1 & 2 \\ 1 & 1 \\ 2 & -1 \end{pmatrix}.$$

Для каждого фиксированного значения t задача (5.1) становится задачей линейного программирования, которую называют *принадлежащей значению t* .

Решением параметрической задачи (5.1) называют явным образом заданную функцию

$$\hat{f}(t) = \max \{f(x) = (a_0 + bt)^T x \mid Ax \leq a_0; \quad x \geq 0\},$$

являющуюся *решающей функцией задачи (5.1)*, и набор *решающих отношений* $\hat{x}_1(t), \dots, \hat{x}_n(t)$, каждое значение которых при данном t равно значениям переменных x_1, x_2, \dots, x_n , образующих оптимальное решение задачи, принадлежащей данному значению t (если это решение существует). Доказано, что r -я *критическая область* K^r задачи (5.1), $r = 1, 2, \dots$, определяемая условием

$$K^r = \{t \mid a_{0j}^r(t) = a_{0j}^r + b_j^r t \geq 0, \quad j = 1, 2, \dots, n + m\},$$

является отрезком (замкнутым интервалом). Область определения Q решающей функции $\hat{f}(t)$ выпукла; решающая функция $\hat{f}(t)$ выпукла в своей области определения.

При любом $r = 1, 2, \dots$ для всех значений t из критической области K^r решающая функция является линейной и в силу этого непрерывной в области Q .

В критической области K^r при любом $r = 1, 2, \dots$ множества значений решающих отношений $\hat{x}_j(t)$, $j = 1, 2, \dots, n$, ограничены постоянными величинами; сами эти отношения полунепрерывны сверху.

Для построения решающей функции будем пользоваться симплекс-методом в следующей его модификации. Под последней строкой первой симплекс-таблицы, полученной при $t = t_0 = 0$ и

содержащей некоторое допустимое базисное решение, приписывают еще одну строку коэффициентов $b_j^{(1)}t = -b_jt$, $j = 1, 2, \dots, n + m$, где $b_j = 0$ для $j = n + 1, \dots, n + m$. Полученная строка подвергается тем же преобразованиям, что и остальные (табл. 5.1; $r = 1$). Естественно, что в задаче (5.1) мы предварительно перешли к ограничениям-равенствам и ввели новые переменные x_3, x_4, x_5 . За три итерации ($r = 1, 2, 3$) мы получили оптимальное решение при $t = 0$ (в табл. 5.1 для каждой итерации справа выделены разрешающие элементы).

Коэффициенты в целевой функции для выбора разрешающего элемента задаются двумя последними строками при фиксированном (заданном) значении $t = t_0$. Для данного допустимого базисного решения необходимо построить соответствующую критическую область — множество всех значений t , при которых это решение будет оптимальным. Это множество будет замкнутым интервалом, границы которого находят по формулам

$$m_{-1}^r = \max \left\{ -\frac{a_{0j}^r}{b_j^r}, -\infty \mid b_j^r > 0, j = 1, 2, \dots, n + m \right\},$$

$$m_{+1}^r = \min \left\{ -\frac{a_{0j}^r}{b_j^r}, +\infty \mid b_j^r < 0, j = 1, 2, \dots, n + m \right\}.$$

Если нижней (верхней) границей этого интервала является $-\infty$ ($+\infty$), то найдена нижняя (верхняя) граница области Q . В противном случае по крайней мере один из коэффициентов целевой функции $a_{0j}^r(t) = a_{0j}^r + b_j^r t$ равен нулю. При значениях $t = t_p$, обращающих выражение $a_{0j}^r + b_j^r t$ в нуль и называемых *критическими*, решения задачи определены неоднозначно. Если существует второе базисное решение, то, изменяя базис, можно задавать соседнюю критическую область значений t . Если при попытке заменить базис обнаружена неограниченность решения, то гиперплоскость, соответствующая целевой функции при рассматриваемом ее значении, оказывается параллельной одной из граней выпуклого многогранника, определяемого системой ограничений задачи. Такая грань содержит точки, хотя бы одна из координат которых сколь угодно велика по абсолютному значению. В этом случае до-

Таблица 5.1

**Симплекс-таблицы для линейной задачи
параметрического программирования**

Номер итерации	x_1	x_2	x_2	x_2	x_2	$a_{\cdot 0}$	Базис
$r = 1$	-1	<u>2</u>	1	0	0	4	x_3
	1	1	0	1	0	5	x_4
	2	-1	0	0	1	8	x_5
	-2	-3	0	0	0	0	$-a_0^1$
	$-t$	t	0	0	0	t	$-b_j^1 t$
$r = 2$	1/2	1	1/2	0	0	2	x_2
	<u>3/2</u>	0	-1/2	1	0	3	x_4
	3/2	0	1/2	0	1	10	x_5
	-7/2	0	3/2	0	0	6	$-a_0^2$
	$-t/2$	0	$-t/2$	0	0	$-t$	$-b_j^2 t$
$r = 3$	0	1	1/3	1/3	0	3	x_2
	1	0	-1/3	<u>2/3</u>	0	2	x_1
	0	0	<u>1</u>	-1	1	7	x_5
	0	0	1/3	7/3	0	13	$-a_0^3$
	0	0	$-2t/3$	$t/3$	0	0	$-b_j^3 t$
$\pi = -1$							
$r = 4$	1/2	1	1/2	0	0	2	x_2
	3/2	0	-1/2	1	0	3	x_4
	3/2	0	1/2	0	1	10	x_5
	-7/2	0	3/2	0	0	6	$-a_0^4$
	$-t/2$	0	$-t/2$	0	0	$-t$	$-b_j^4 t$
$\pi = +1$							
$r = 4$	0	1	0	2/3	-1/3	2/3	x_2
	1	0	0	1/3	1/3	13/3	x_1
	0	0	1	-1	1	7	x_3
	0	0	0	8/3	-1/3	32/3	$-a_0^5$
	0	0	0	$-t/3$	$2t/3$	$14t/3$	$-b_j^4 t$
$r = 5$	0	3/2	0	1	-1/2	1	x_4
	1	-1/2	0	0	1/2	4	x_2
	0	1	1	0	1/2	8	x_3
	0	-8/2	0	0	1	8	$-a_0^6$
	0	$t/2$	0	0	$t/2$	$5t$	$-b_j^5 t$

стигнута конечная нижняя (верхняя) грань области Q . Пусть при построении критической области стало известно, что некоторая точка найденного интервала имеет координату, меньшую (большую), чем координата верхней (нижней) границы интервала, полученного на предыдущем шаге построения. В таком случае говорят, что эта верхняя (нижняя) граница *превзойдена*. Тогда сама эта верхняя (нижняя) граница является и нижней (верхней) границей строящегося интервала.

Определим критическое значение $t_p = m_{\pi}^r$. Индекс $\pi = -1$ применяют при нахождении нижней границы области Q , индекс $\pi = +1$ — при нахождении верхней границы. В нашем случае $r = 3$; ищем нижнюю границу области Q , т. е. $\pi = -1$. Положив $p = \pi = -1$ (для $b_j > 0$, $j = 1, 2, \dots, n + m$), получим

$$t_{-1} = m_{-1}^3 = \max \left\{ -\frac{7/3}{1/3}; -\infty \right\} = -7.$$

Из табл. 5.1 для $r = 3$ следует, что при $t = 7$ получим $\hat{f}(t) = 13$, $\hat{x}_1 = 2(1 - \lambda)$, $\hat{x}_2 = 2\lambda + 3(1 - \lambda)$, $0 \leq \lambda \leq 1$. Это критическое значение может быть превзойдено. В табл. 5.1 при $r = 3$ выбран разрешающий элемент, равный $2/3$ при $j = 4$. Общее условие для выбора этой небазисной переменной имеет вид

$$\tau_p^r \in \{\tau \mid a_{0\tau}^r + b_{0\tau}^r t_p = 0, \quad x_{\tau} \text{ — небазисная переменная.}\}$$

Здесь $\tau = 4$.

Получим новые значения элементов табл. 5.1 при $\pi = -1$ и $r = 4$, которые совпали с элементами этой таблицы при $r = 2$. Поскольку все b_j^4 отрицательны или равны нулю, то $t_{-2} = -\infty$ и является нижней границей области Q . Из табл. 5.1 при $r = 4$ и $t < -7$ получаем, что $\hat{f}(t) = 6 - t$, $\hat{x}_1 = 0$, $\hat{x}_2 = 2$.

Найдем верхнюю границу области Q , полагая $\pi = p = +1$, $r = 3$. Тогда для $b_j < 0$, $j = 1, 2, \dots, n + m$, имеем

$$t_{+1} = m_{+1}^3 = \min \left\{ -\frac{1/3}{-2/3}; +\infty \right\} = \frac{1}{2}.$$

При $t = 1/2$ получим $\hat{f}(t) = 13$, $\hat{x}_1 = 2\lambda + 13(1 - \lambda)/3$, $\hat{x}_2 = 3\lambda + 2(1 - \lambda)/3$, $0 \leq \lambda \leq 1$.

Выбираем разрешающий элемент из столбца для x_3 и вычисляем новые элементы табл. 5.1 при $\pi = +1$ и $r = 4$, откуда найдем новое t_{+2} , превышающее критическое значение $t_{+1} = 1/2$. Получим новое значение t_{+1} при $-7 < t < 1/2$, $\hat{f}(t) = 13$, $\hat{x}_1 = 2$, $\hat{x}_2 = 3$:

$$t_{+1} = m_{+1}^4 = \min \left\{ -\frac{8/3}{-1/3}; +\infty \right\} = 8.$$

При $t = 8$ имеем

$$\hat{f}(t) = \frac{1}{3}(32 + 14 \cdot 8) = 48, \quad \hat{x}_1 = \frac{13}{3}\lambda + 4(1 - \lambda), \quad \hat{x}_2 = \frac{2\lambda}{3},$$

при $\frac{1}{2} < t < 8$ имеем

$$\hat{f}(t) = \frac{32 + 14t}{3}, \quad \hat{x}_1 = \frac{13}{3}, \quad \hat{x}_2 = \frac{2}{3}.$$

Строим табл. 5.1 при $r = 5$ и $\pi = +1$. Получаем $t > 8$.

Окончательные результаты приведены в табл. 5.2. На рис. 5.1 показаны «вращение» целевой функции, поведение решающей функции $\hat{f}(t)$ и геометрические образы отношений $\hat{x}_1(t)$ и $\hat{x}_2(t)$.

Таблица 5.2

Решение параметрической задачи

t	$\hat{f}(t)$	\hat{x}_1	\hat{x}_2
$t < -7$	$6 - t$	0	2
$t = -7$	13	$2(1 - \lambda)$	$2\lambda + 3(1 - \lambda)$
$-7 < t < \frac{1}{2}$	13	2	3
$t = \frac{1}{2}$	13	$2\lambda + \frac{13(1 - \lambda)}{3}$	$3\lambda + \frac{2(1 - \lambda)}{3}$
$\frac{1}{2} < t < 8$	$\frac{32 + 14t}{3}$	$\frac{13}{3}$	$\frac{2}{3}$
$t = 8$	48	$\frac{13\lambda}{3} + 4(1 - \lambda)$	$\frac{2\lambda}{3}$
$t > 8$	$8 + 5t$	4	0

Алгоритм решения линейной параметрической задачи в случае зависимости от параметра коэффициентов целевой функции в

предположении, что допустимая область, заданная ограничениями задачи, не пуста и что для некоторого значения параметра $t = t_0$ существует оптимальное решение, имеет следующий вид.

1. При $t = t_0$ r -я симплекс-таблица является оптимальной. Полагаем $\pi = -1$, $p = \pi$ и переходим к п. 2.

2. Определяем критическое значение $t_p = m_\pi^r$:

а) если $|t_p| = \infty$, то достигнута граница области Q ;

а') при $\pi = -1$ находим нижнюю границу области Q , затем полагаем $\pi = +1$, $p = \pi$, заменяем r на $r + p + 1$ и определяем новое критическое значение t_p ;

а'') при $\pi = +1$ находим верхнюю границу области Q и переходим к п. 3;

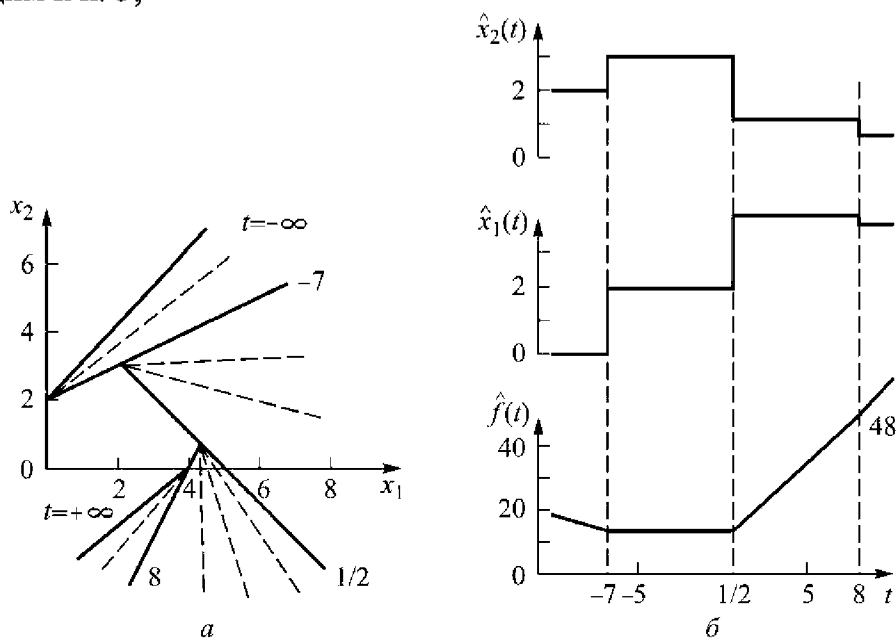


Рис. 5.1. Решение задачи параметрического линейного программирования:

a — «вращение» целевой функции; b — поведение $\hat{x}_1(t)$, $\hat{x}_2(t)$ и решающей функции $\hat{f}(t)$ в зависимости от параметра t

б) если $|t_p| < \infty$, то определяем, можно ли превзойти критическое значение t_p : с помощью выражения для τ_p^r , иногда выбором из нескольких возможных значений, определяем новую базисную переменную; в этом случае следует учитывать существование множества различных решений;

б') если $a_{ij}^r \leq 0$, $i = 1, 2, \dots, m$, то достигнута граница области Q ; переходим к п. а') или п. а'');

б'') в противном случае строим новую симплекс-таблицу; заменяя r на $r+1$, p на $p+\pi$, определяем новое критическое значение t_p .

3. Вычисления прекращаем. Для каждой критической области значение решающей функции $\hat{f}(t)$ берем из табл. 5.1. Там же определены оптимальные значения решающих отношений $\hat{x}_1(t), \dots, \hat{x}_n(t)$.

На практике часто возникают следующие частные случаи параметрической задачи:

1) значение параметра t принадлежит некоторому заранее выбранному интервалу $L \subset \mathbb{R}^1$;

2) требуется найти только минимум (максимум) решающей функции $\hat{f}(t)$ для всех $t \in \mathbb{R}^1$, т. е. требуется определить те значения t , при которых возможно достижение этого минимума (максимума);

3) требуется найти интервал возможных изменений какого-то одного из коэффициентов целевой функции.

Особый интерес представляет задача нахождения наибольшего отрезка $t_1 \leq t \leq t_2$ такого, что оптимальное базисное решение любой из задач, принадлежащих значениям t из этого отрезка, совпадает с оптимальным базисным решением задачи, принадлежащей значению $t = 0$.

Если область определения Q решающей функции $\hat{f}(t)$ ограничена заданным заранее интервалом L , то вычисления, проводимые в п. 2 алгоритма, заканчивают, когда будут достигнуты границы интервала L . Если отрезок $t_{-1} \leq t \leq t_{+1}$ не имеет общих точек с интервалом L , то полагают $\pi = -1$ или $\pi = +1$ в зависимости от того, будут ли значения $t \in L$ меньше t_{-1} или больше t_{+1} .

Если в задаче необходимо установить только минимум (максимум) значения решающей функции $\hat{f}(t)$, то вычисления, предписываемые п. 2, проводят при $\pi = -1$ или $\pi = +1$ в зависимости от того, в каком направлении уменьшается значение функции $\hat{f}(t)$.

Пусть необходимо определить только возможность вариации одного из коэффициентов. Сначала устанавливают, существует ли решение при $t = 0$, и находят это решение. Вводя затем параметр t , определяют значения t_{-1} (равные t_1) и t_{+1} (равные t_2).

§ 5.3. Многопродуктовые потоки в сетях

Существует немало практических задач о перевозке нескольких различных продуктов.

Рассмотрим упрощенную задачу [78] транспортировки трех видов фруктов из садов, расположенных в Калифорнии, в магазины, ведущие оптовую торговлю. Предположим, что сады расположены в городах Санта-Барбара, Бейкерсфилд и Сакраменто. В период уборки урожая из этих садов поставляют в различных количествах апельсины, лимоны и лаймы (табл. 5.3).

Таблица 5.3

Производительность (ящики в неделю)

Город	Перевозимый продукт		
	Апельсины	Лимоны	Лаймы
Санта-Барбара	800	700	700
Бейкерсфилд	1000	800	500
Сакраменто	500	1000	500

Таблица 5.4

Величина спроса (ящики в неделю)

Город	Перевозимый продукт		
	Апельсины	Лимоны	Лаймы
Денвер	900	900	700
Сиэтл	700	1000	500
Канзас-Сити	700	600	500

По контракту необходимо доставлять в оптовые магазины, расположенные в Денвере, Сиэтле и Канзас-Сити, такое количество фруктов, которое указано в табл. 5.4. Перевозка осуществляется по железной дороге, однако в каждом поезде для фруктов отведено ограниченное место. Пропускные способности путей между различными пунктами отправления и пунктами назначения приведены в табл. 5.5. Независимо от вида фруктов данные величины задаются числом ящиков, перевозимых в неделю (предполагается, что все фрукты упакованы в ящики одинаковых размеров). Вес каждого ящика, а следовательно, и соответствующие транспортные затраты зависят от вида упакованных в него фруктов. Затраты на транспортировку одного ящика приведены в табл. 5.6.

Таблица 5.5

**Пропускная способность железных дорог
(ящики в неделю)**

Из города	В город		
	Денвер	Сиэтл	Канзас-Сити
Санта-Барбара	1200	900	1000
Бейкерсфилд	1200	1000	1100
Сакраменто	950	1300	1000

Таблица 5.6

Транспортные затраты (центы за ящик)

Из города	В город		
	Денвер	Сиэтл	Канзас-Сити
Санта-Барбара			
Апельсины	3,2	2,5	5,6
Лимоны	2,4	1,9	4,3
Лаймы	2,3	1,7	4,0
Бейкерсфилд			
Апельсины	3,0	2,1	5,5
Лимоны	2,3	1,7	4,4
Лаймы	2,0	1,4	4,3
Сакраменто			
Апельсины	3,1	2,0	5,7
Лимоны	2,2	1,6	4,8
Лаймы	2,0	1,5	4,3

Задача определения схемы перевозки фруктов, минимизирующей транспортные затраты, является *задачей о многопродуктовом потоке*. Характерная особенность задач о многопродуктовом потоке состоит в том, что по дугам сети протекает несколько неоднородных потоков. При этом суммарный поток всех продуктов по дуге ограничен ее пропускной способностью. Если бы это было не так, то можно было бы для каждого продукта независимо решить минимизационную транспортную задачу. Однако указанные выше ограничения на пропускные способности дуг существенно усложняют решение задачи.

Задачи о многопродуктовом потоке, как и об однопродуктовом потоке, могут быть сформулированы в виде задач линейного программирования. Задача о транспортировке фруктов является примером многопродуктовой транспортной задачи.

Обозначим через x_{ij}^k поток k -го продукта из i -го источника в j -й сток, а через c_{ij}^k — стоимость транспортировки единицы этого продукта. Пусть далее a_i^k и b_j^k — это соответственно предложение узла i и спрос узла j для k -го продукта, а u_{ij} — пропускная способность дуги (i, j) . Математическая постановка многопродуктовой транспортной задачи выглядит следующим образом:

$$\sum_{k=1}^r \sum_{i=1}^m \sum_{j=1}^n c_{ij}^k x_{ij}^k \rightarrow \min \quad (5.2)$$

при условии, что

$$\sum_i x_{ij}^k = b_j^k, \quad j, k \in \mathbb{N},$$

$$\sum_j x_{ij}^k = a_i^k, \quad i, k \in \mathbb{N},$$

$$\sum_k x_{ij}^k \leq u_{ij}, \quad i, j \in \mathbb{N},$$

$$x_{ij}^k \geq 0, \quad i, j, k \in \mathbb{N}.$$

Как и в однопродуктовой транспортной задаче, предполагается, что для каждого продукта суммарное предложение равно суммарному спросу, т. е. для всех k

$$\sum_i a_i^k = \sum_j b_j^k.$$

Во многих задачах, называемых *многопродуктовыми задачами о перевозках*, вводят промежуточные узлы, т. е. такие узлы, которым не приписывают ни спрос, ни предложение, а только требуют, чтобы для них выполнялось условие сохранения потока.

В качестве одного из таких примеров рассмотрим сеть, изображенную на рис. 5.2. Узлы 1 и 2 являются источниками первого продукта, при этом узел 2 — источник второго продукта. Узел 5 является пунктом назначения для обоих продуктов, а узлы 3 и 4 являются промежуточными.

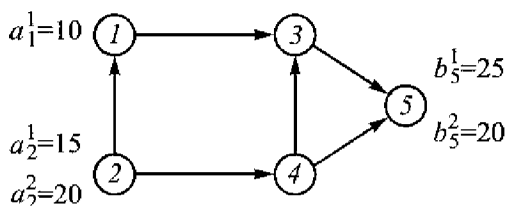


Рис. 5.2. Сеть в многопродуктовой задаче о перевозках

Многопродуктовая задача о перевозках может быть сфор-

мулирована в виде следующей задачи линейного программирования:

$$\sum_{k=1}^r \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k \rightarrow \min$$

при условии, что:

$$1) \sum_j x_{ij}^k - \sum_j x_{ji}^k = a_i^k, \text{ если узел } i \text{ является источником } k\text{-го}$$

продукта;

$$2) \sum_j x_{ij}^k - \sum_j x_{ji}^k = 0, \text{ если узел } i \text{ является промежуточным уз-}$$

лом;

$$3) \sum_i x_{ij}^k - \sum_i x_{ji}^k = -b_j^k, \text{ если узел } j \text{ является стоком } k\text{-го про-}$$

дукта;

$$4) \sum_k x_{ij}^k \leq u_{ij}, \quad x_{ij}^k \geq 0 \text{ для всех } k \text{ и } (i,j) \in A, \text{ где } A \text{ — множе-}$$

ство дуг сети.

Одна из трудностей решения задачи о многопродуктовом потоке вызвана тем, что оптимальные решения могут быть нецелочисленными. В качестве примера рассмотрим двухпродуктовую транспортную задачу, сетевая формулировка которой представлена на рис. 5.3. Дугам сети приписаны числа $(c_{ij}^1, c_{ij}^2, u_{ij})$. Оптимальное решение соответствующей задачи линейного программирования представлено в табл. 5.7.

Нецелочисленные решения возникают по той причине, что матрицы ограничений в задачах линейного программирования, вообще говоря, не унимодулярны. Напомним, что условие унимо-

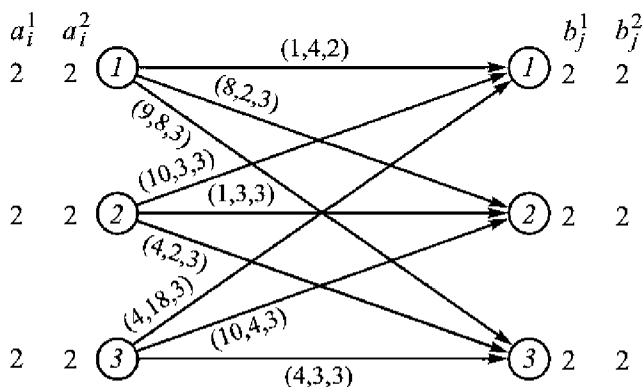


Рис. 5.3. Двухпродуктовая транспортная задача

дулярности матрицы ограничений является достаточным для существования целочисленных оптимальных решений. Поскольку в задачах об однопродуктовом потоке данное условие выполнено, то для них можно разработать высокоэффективные алгоритмы, в которых по существу выполняются только две операции: сложение и вычитание, а такие операции, как, например, деление и обращение матрицы, не требуются. С вычислительной точки зрения арифметические операции, выполняемые с целыми числами, проводятся намного быстрее, чем операции, использующие числа с плавающей точкой (десятичные). Благодаря этому создают программы, позволяющие очень быстро решать задачи большой размерности.

Таблица 5.7

**Оптимальное решение задачи
линейного программирования**

Дуга	Продукт 1	Продукт 2
(1, 1)	3/2	1/2
(1, 2)	0	3/2
(1, 3)	1/2	0
(2, 1)	0	3/2
(2, 2)	2	0
(2, 3)	0	1/2
(3, 1)	1/2	0
(3, 2)	0	1/2
(3, 3)	3/2	3/2

Для задач о многопродуктовом потоке были разработаны специальные алгоритмы, в которых учтены специфика структуры и свойства данного класса задач. Было показано, что эти алгоритмы являются более быстрыми, чем процедуры решения общей задачи ЛП, но значительно более медленными, чем соответствующие алгоритмы задач об однопродуктовом потоке. Детальное изучение этих методов выходит за рамки настоящей книги. Однако в данной главе мы рассмотрим ряд специальных вопросов, связанных с задачами о многопродуктовом потоке и методами их решения.

§ 5.4. Специальный класс целочисленных задач о многопродуктовом потоке

Как отмечалось выше, общая задача о многопродуктовом потоке может не иметь целочисленного оптимального решения. Однако существует несколько классов этих задач, в которых матрицы ограничений являются унимодулярными, и, следовательно, целочисленные оптимальные решения существуют. Отметим, что многие из этих задач могут быть сведены к эквивалентным задачам об однопродуктовом потоке, которые несложно решаются с помощью любого алгоритма решения задачи о потоке минимальной стоимости.

Для многопродуктовой транспортной задачи известен следующий результат. Матрица ограничений в многопродуктовой транспортной задаче является унимодулярной в том и только в том случае, когда число источников m или число стоков n не превосходит двух.

Этот результат гарантирует существование целочисленного оптимального решения в случае, если $m \leq 2$ или $n \leq 2$, независимо от числа продуктов. Очевидно, что если $m = 1$ или $n = 1$, то решение тривиально. Однако, когда и m , и n превосходят 2, возникает общая задача линейного программирования. Покажем, что при этом существует более простой метод решения.

Введя слабые переменные s_{ij} в ограничения на пропускные способности дуг, переформулируем многопродуктовую транспортную задачу (5.2) следующим образом:

$$\sum_{k=1}^r \sum_{i=1}^m \sum_{j=1}^n c_{ij}^k x_{ij}^k \rightarrow \min$$

при условии, что для всех i, j, k выполнено

$$\sum_i x_{ij}^k = b_j^k, \quad \sum_k x_{ij}^k + s_{ij} = u_{ij},$$

$$\sum_j x_{ij}^k = a_i^k, \quad x_{ij}^k, s_{ij} \geq 0.$$

При $m = 2$ данная задача сводится к следующей задаче линейного программирования:

$$\sum_{k=1}^r \sum_{j=1}^n [(c_{2j}^k - c_{1j}^k)x_{2j}^k + c_{1j}^k b_j^k] \rightarrow \min \quad (5.3)$$

при условии, что для всех i, j, k выполнено

$$\sum_k x_{2j}^k + s_{2j} = u_{2j}, \quad (5.4)$$

$$-\sum_j x_{2j}^k = -a_2^k, \quad (5.5)$$

$$-\sum_j s_{2j} = -\sum_j u_{2j} + \sum_k a_2^k, \quad (5.6)$$

$$x_{2j}^k + x_{1j}^k = b_j^k, \quad (5.7)$$

$$s_{2j} + s_{1j} = u_{2j} + u_{1j} - \sum_k b_j^k, \quad (5.8)$$

$$x_{ij}^k, s_{ij} \geq 0, \quad i, j, k \in \mathbb{N}. \quad (5.9)$$

Нетрудно заметить, что в равенствах (5.4)–(5.6) каждая из переменных x_{2j}^k и s_{2j} появляется дважды: один раз со знаком плюс и один раз со знаком минус. Отметим также, что переменные x_{1j}^k и s_{1j} входят только в равенства (5.7) и (5.8) (со знаком плюс). Поэтому можно рассматривать эти переменные как слабые и исключить их, записав (5.7) и (5.8) в виде

$$x_{2j}^k \leq b_j^k, \quad s_{2j} \leq u_{2j} + u_{1j} - \sum_k b_j^k.$$

Величины, стоящие в правых частях равенств (5.4)–(5.6), представляют собой предложение (если они положительные) и спрос (если отрицательные). Равенства (5.4)–(5.6) описывают транспортную модель, сетевая структура которой показана на рис. 5.4, где

$$\Pi = \sum_{j=1}^n u_{2j} - \sum_{k=1}^r a_2^k.$$

Правые части равенств (5.7) и (5.8) представляют собой пропускные способности дуг, соответствующих переменным x_{2j}^k и s_{2j} .

Для решения данной задачи определяют оптимальные потоки x_{2j}^k и s_{2j} . Затем из равенств (5.7) и (5.8) могут быть найдены величины x_{1j}^k и s_{1j} .

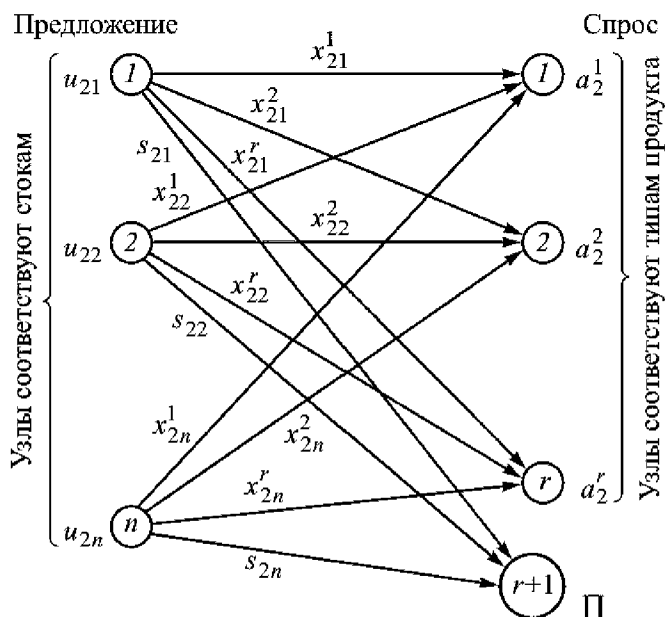


Рис. 5.4. Сеть в многопродуктовой транспортной задаче, сведенной к задаче об однопродуктовом потоке

Интересно то, что исходная задача линейного программирования не имела форму сетевой задачи. Однако с помощью преобразования ограничений удалось свести ее к сетевой и тем самым существенно упростить решение.

§ 5.5. Приближенное решение многопродуктовой транспортной задачи методом агрегирования

Нередко для лица, принимающего решение, достаточно найти субоптимальные решения сложных задач. Как правило, это происходит в тех случаях, когда отсутствуют программы, позволяющие проводить точную оптимизацию, или когда использование таких программ требует слишком больших затрат вследствие циклического обращения к ним, или когда нет возможности усовершенствовать основное программное обеспечение из-за жестких ограничений на время ответа.

Одним из методов, позволяющих упростить решение задачи, является *метод агрегирования*, заключающийся в замене множества объектов (таких, как переменные, узлы сети и т. п.) одним объектом. В этом случае сокращаются как время решения задачи, так и требуемая машинная память. Однако полученное решение, как правило, не является оптимальным.

Будем рассматривать многопродуктовую транспортную задачу с m источниками, n стоками и r продуктами. Сведем ее к сетевой задаче с двумя источниками, разбив множество источников на два подмножества: S_1 и S_2 . Теперь необходимо определить новые предложения агрегированных узлов, решить вопрос о стоимостях и пропускных способностях агрегированных дуг и найти способ построения допустимого решения исходной задачи с помощью решения агрегированной задачи.

Поскольку каждый из двух агрегированных узлов представляет собой совокупность источников в исходной задаче, то величины предложения этих двух узлов \bar{a}_l^k определяют следующим образом:

$$\bar{a}_l^k = \sum_{i \in S_l} a_i^k, \quad l=1,2,$$

где a_i^k — предложение k -го продукта из i -го узла исходной задачи.

Пусть y_{lj}^k — поток k -го продукта из узла l в узел j агрегированной сети. Поскольку дуга (l, j) получена в результате агрегирования дуг, ведущих из узлов $i \in S_l$ в сток j , то

$$y_{lj}^k = \sum_{i \in S_l} x_{ij}^k,$$

где x_{ij}^k — поток k -го продукта из узла i в узел j исходной задачи.

Для определения стоимостей воспользуемся понятием *взвешенного агрегирования*. Стоимости взвешивают пропорционально величинам предложения источников, представленных агрегированными узлами:

$$\bar{c}_{lj}^k = \sum_{i \in S_l} c_{ij}^k \left(\frac{a_i^k}{\bar{a}_l^k} \right).$$

Чтобы определить пропускные способности \bar{u}_{lj} агрегированных дуг, найдем величины

$$\delta_i = \min_k \left\{ \frac{\bar{a}_l^k}{a_i^k} \right\}, \quad i \in S_l.$$

Тогда $\bar{u}_{lj} = \min_{i \in S_l} \{\delta_i u_{ij}^k\}$.

Если принять \bar{u}_{ij} как сумму пропускных способностей исходных дуг u_{ij} из источников $i \in S_l$ в сток j , то могут возникнуть трудности при построении допустимого решения исходной задачи.

Сформулируем агрегированную задачу в виде следующей задачи линейного программирования:

$$\sum_{k=1}^r \sum_{l=1}^2 \sum_{j=1}^n \bar{c}_{lj}^k x_{lj}^k \rightarrow \min$$

при условии, что для всех l, i, k выполнено

$$\sum_l y_{lj}^k = b_j^k, \quad \sum_j y_{lj}^k = \bar{a}_l^k, \quad \sum_k y_{lj}^k \leq \bar{u}_{lj}, \quad y_{lj}^k \geq 0.$$

Теперь построим решение исходной задачи, используя оптимальное решение агрегированной задачи. Определим величины

$$x_{ij}^k = y_{lj}^k \left(\frac{a_i^k}{\bar{a}_l^k} \right), \quad i \in S_l,$$

откуда следует, что

$$\sum_{i \in S_l} x_{ij}^k = y_{lj}^k \left(\sum_{i \in S_l} \frac{a_i^k}{\bar{a}_l^k} \right) = y_{lj}^k \left(\frac{\bar{a}_l^k}{\bar{a}_l^k} \right) = y_{lj}^k.$$

Эту процедуру называют *деагрегированием с фиксированными весами*. Кроме того,

$$\sum_k \sum_i \sum_j c_{ij}^k x_{ij}^k = \sum_k \sum_l \sum_j \bar{c}_{lj}^k y_{lj}^k.$$

Иными словами, значение целевой функции в результате деагрегирования с фиксированными весами не изменяется. Однако следует помнить, что в силу определения величин \bar{u}_{ij} в агрегированной задаче может не существовать допустимое решение, даже если в исходной задаче оно существует. В этом случае можно попытаться воспользоваться другими методами агрегирования.

Продemonстрируем процедуру агрегирования на конкретном примере. Перейдем к решению задачи о транспортировке фруктов, сформулированной в § 5.3. Агрегируем источники 2 и 3. Величины предложения, стоимости и пропускные способности для агрегированной задачи содержатся в табл. 5.8.

Таблица 5.8

Параметры агрегированной задачи

Предложение			
Город	Апельсины	Лимоны	Лаймы
Санта-Барбара	800	700	700
{Бейкерсфилд Сакраменто	1500	1800	1000
Стоимость			
Из города	В город		
	Денвер	Сиэтл	Канзас-Сити
Санта-Барбара			
Апельсины	3200	2500	5600
Лимоны	2400	1900	4300
Лаймы	2300	1700	4000
{Бейкерсфилд Сакраменто			
Апельсины	3033	2067	5567
Лимоны	2244	1644	4622
Лаймы	2000	1450	4300
Пропускная способность			
Из города	В город		
	Денвер	Сиэтл	Канзас-Сити
Санта-Барбара	1200	900	1000
{Бейкерсфилд Сакраменто	1500	1000	1000

Стоимость в оптимальном решении исходной задачи равна 18 570 долл. Стоимость в оптимальном решении агрегированной задачи равна 18 799,10 долл. и на 0,26 % превосходит стоимость действительного оптимального решения.

§ 5.6. Приложения задач о многопродуктовом потоке

Задачи о многопродуктовом потоке находят широкое применение при проектировании коммуникационных систем, осуществлении железнодорожных перевозок, планировании производства и распределения, в военном деле, а также в других областях. В настоящем параграфе мы рассмотрим несколько приложений много-

продуктовых сетевых моделей и приведем новые формулировки некоторых задач о многопродуктовом потоке.

Составление расписания движения судов

Одной из наиболее часто возникающих задач в области планирования перевозок является задача составления оптимального расписания движения транспортных средств. Предположим, что известно число судов различных типов, которые отличаются друг от друга скоростью передвижения, грузоподъемностью и эксплуатационными расходами. Функция полезности определяется размерами поставки груза отдельным судном к заданному сроку и затратами на осуществление соответствующей перевозки. Кроме того, существуют затраты (отрицательная полезность), связанные с перегоном судна из порта, в котором оно было разгружено, в порт погрузки. Задача заключается в максимизации функции полезности путем составления оптимальных маршрутов и расписания движения судов.

Данная задача может быть сформулирована в виде следующей сетевой задачи. Каждый узел j сети соответствует прибытию судна в порт назначения в один из допустимых сроков доставки. Каждый узел i соответствует исходному порту и моменту, равному разности срока доставки и времени транспортировки, которое по предположению является детерминированным. Дуга (i, j) соответствует перевозке груза, а дуга (j, i) — перегону судна из порта доставки в исходный порт. Суда k -го типа первоначально располагают в источнике s^k , и дуги (s^k, i) , таким образом, соответствуют началу их эксплуатации. Далее вводят фиктивный сток t и дуги (j, t) , соответствующие завершению эксплуатации судов. И наконец, суммарный поток перевозимого груза по всем дугам, соответствующим различным датам перевозки и типам судов, не должен превосходить некоторой заданной величины. Математически эта задача формулируется следующим образом:

$$f(x) = \sum_k \sum_i \sum_j c_{ij}^k x_{ij}^k \rightarrow \max \quad (5.10)$$

при условии, что

$$\sum_k \sum_{A_v} r_{ij}^k x_{ij}^k \leq b_v, \quad (5.11)$$

$$\sum_i x_{ij}^k - \sum_j x_{ji}^k = \begin{cases} d^k, & i = s^k, \\ 0, & i \neq s^k, t, \\ -d^k, & i = t, \end{cases} \quad (5.12)$$

$$0 \leq x_{ij}^k \leq u_{ij}^k. \quad (5.13)$$

Здесь через c_{ij}^k обозначен коэффициент полезности, соответствующий отдельному судну, перевозимому грузу и маршруту; A_v — множество допустимых маршрутов для данного груза; r_{ij}^k — грузоподъемность судна k -го типа на маршруте (i, j) . Суммарный перевозимый груз не может превышать спроса на него b_v . Ограничения (5.12) описывают условие сохранения потока по числу судов d^k , а величины x_{ij}^k равны 0 или ∞ в зависимости от того, может быть использовано на дуге (i, j) судно k -го типа или не может.

В качестве примера рассмотрим задачу, представленную в табл. 5.9. Число судов каждого типа равно двум.

Таблица 5.9

**Исходные данные к задаче составления
расписания движения судов**

Груз	Максимально допустимый момент начала перевозки (через n дней)	Допустимые сроки доставки
1	0	4, 5
2	2	6, 7, 8
3	3	7, 8

Соответствующая сеть изображена на рис. 5.5. Отметим, что в рассмотренном классе задач узлы служат для обозначения места и времени.

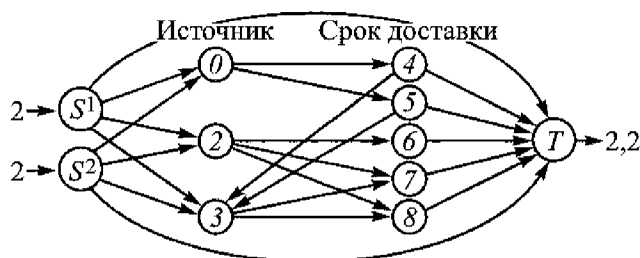


Рис. 5.5. Сеть в задаче составления расписания движения судов

Проектирование городской транспортной сети

Многопродуктовые сетевые модели используют и при проектировании городских транспортных сетей. В этих моделях узлы соответствуют участкам или районам города, а дуги — улицам или дорогам других видов. Требования к транспортной сети задаются матрицей поездок D , элементы d_{ij} которой равны числу транспортных средств, движущихся из участка i к участку j в течение фиксированного интервала времени. Каждая дуга имеет заданную пропускную способность u_{ij} , которая может быть увеличена (например, в результате улучшения дорог). Плата за увеличение пропускной способности дуги (i, j) на условную единицу равна c_{ij} . Общая сумма денежных средств, предназначенных для улучшения дорог, равна B . «Продуктами» в данной модели являются потоки из каждого источника во все пункты назначения. Пусть x_{ij}^k — число транспортных средств, движущихся по дуге (i, j) и начавших свой путь в узле k , и y_{ij} — увеличение пропускной способности дуги (i, j) . Предположим, что время проезда по дуге (i, j) является некоторой функцией потока

$$F_{ij} = f_{ij} \left(\sum_k x_{ij}^k \right).$$

Задача проектирования сети заключается в определении дуг, пропускную способность которых следует увеличить, и вычислении потоков по каждой дуге, минимизирующих общее время поездки. Математически данная задача формулируется следующим образом:

$$\sum_i \sum_j f_{ij} \left(\sum_k x_{ij}^k \right) \rightarrow \min \quad (5.14)$$

при условии, что

$$\sum_j x_{ji}^k - \sum_j x_{ij}^k = d_{ki},$$

$$\sum_k x_{ij}^k \leq u_{ij} + y_{ij}, \quad \sum_i \sum_j c_{ij} y_{ij} \leq B,$$

$$x_{ij}^k \geq 0, \quad y_{ij} \geq 0.$$

В задаче (5.14) сделано одно важное допущение, касающееся поведения водителя. Оно основано на классическом принципе распределения транспортных средств: водители выбирают маршруты таким образом, чтобы суммарное время поездок для всей системы было минимальным. Как правило, целевая функция задачи (5.14) является нелинейной и выпуклой.

Модели вычислительных систем

Многие модели вычислительных систем очень похожи на только что рассмотренную нами модель городской транспортной сети. В этих моделях дуги соответствуют каналам, а узлы — терминалам, системным и запоминающим устройствам и т. п. Каждая дуга имеет фиксированную пропускную способность u_{ij} , которая может быть увеличена на y_{ij} единиц вплоть до максимального значения b_{ij} . Стоимость увеличения пропускной способности на единицу измерения равна c_{ij} . Роль продуктов вновь играют потоки (информация) из источника во все пункты назначения, а вместо матрицы поездок D вводят матрицу, элементы d_{ij} которой соответствуют объему информации, передаваемой из узла i в узел j . Стоимость передачи единицы информации по дуге (i, j) равна e_{ij} . Математически эта задача может быть сформулирована следующим образом:

$$\sum_k \sum_i \sum_j e_{ij} x_{ij}^k + \sum_i \sum_j c_{ij} y_{ij} \rightarrow \min$$

при условии, что

$$\sum_j x_{ji}^k - \sum_j x_{ij}^k = d_{ij},$$

$$\sum_k x_{ij}^k \leq u_{ij} + y_{ij}, \quad y_{ij} \leq b_{ij},$$

$$x_{ij}^k \geq 0, \quad y_{ij} \geq 0.$$

Описанная модель может быть использована для определения необходимой пропускной способности каналов, при которой заданные требования выполняются с минимальными затратами. В этом случае значения e_{ij} обычно полагают равными нулю, если

$u_{ij} = 0$, а величины b_{ij} выбирают достаточно большими. Другим примером использования этой модели является задача минимизации стоимости передачи информации при фиксированных пропускных способностях дуг. В этом случае $b_{ij} = 0$.

§ 5.7. Эвристический алгоритм решения задачи синтеза сети связи

Как уже отмечалось, сетевые задачи большой размерности решают, в основном, приближенными эвристическими методами. Рассмотрим один из эвристических методов на примере задачи синтеза некоторой сети связи.

Одна из основных задач при разработке схемы сети связи — это определение наивыгоднейшей конфигурации и распределения каналов на сети. При этом могут решаться и другие задачи:

- 1) выбор типа линий связи (кабельные, радиорелейные, воздушные и т. п.) с соответствующими системами передачи;
- 2) последующее распределение стандартных каналов на сети.

Структура сети — ее *топология* — это совокупность пунктов (узлов, станций и т. п.) и соединяющих их линий связи или каналов в их взаимном расположении. По структуре можно понять потенциальные возможности обеспечения связью отдельных пунктов этой сети.

Экономически оптимальный выбор сетки линий, их емкостей и плана распределения каналов при заданном наборе пунктов сети называют *задачей оптимального синтеза структуры сети*. Ее решение существенно зависит от того, насколько тщательно отобраны допустимые направления.

В качестве основных критериев оптимальности построения сети обычно рассматривают объем капитальных затрат, расход кабельной продукции и т. д. Минимум расхода кабельной продукции обеспечивается при минимуме общей протяженности трасс кабельных и радиорелейных линий связи. Иногда в качестве критерия оптимальности может рассматриваться и объем строительно-монтажных работ.

Реальная сеть должна отвечать одновременно в той или иной мере всем критериям оптимальности, т. е. задача определения наивыгоднейшей конфигурации и распределения каналов сети является *многокритериальной*. Однако в связи со сложностью решения многокритериальных задач обычно в качестве оптимальной

принимают сеть, отвечающую одному, какому-либо заранее заданному, критерию оптимальности. Очевидно, что сеть будет зависеть от выбранного критерия оптимальности и в общем случае будет различной в зависимости от используемого критерия оптимальности.

Рассмотрим задачу синтеза сети при следующих требованиях.

1. Каждый пункт сети должен иметь два независимых пути (прямой и обходной) к центральному узлу (ЦУ). Прямой путь должен обеспечивать передачу по 70–75 % каналов от их расчетного числа, а обходной путь — по 25–30 % каналов. Это соотношение может измениться, а поэтому должно задаваться на входе задачи.

2. Должен существовать путь, соединяющий каждый пункт сети с дополнительным узлом (ДУ) связи, не совпадающим с ЦУ. На некоторых участках этот путь может совпадать с прямым и обходным путями.

В сетях связи налагают следующие дополнительные требования.

3. На линиях связи «пункт — ЦУ» число транзитов по высокой частоте (ВЧ) должно быть не более двух в случае применения аналоговых систем передачи.

4. Суммарная емкость (число каналов) на любом участке зонной сети должна быть кратна 12 для аналоговых систем передачи и 30 для цифровых систем.

5. Иногда требуется предусмотреть возможность того, чтобы через некоторые пункты проходило минимальное число путей. Эти пункты отмечают в исходных данных.

Задача синтеза структуры сети может быть наиболее адекватно представлена (в силу нелинейного ступенчатого изменения целевых функций, в частности функции стоимости линии связи) как задача целочисленного нелинейного программирования в различных формах.

Рассмотрим задачу выбора оптимального варианта построения сети с двумя независимыми путями к единственному узлу (центральному узлу), к которому передается информация со всех других узлов сети. В качестве целевой функции рассматривается функция стоимости сети.

Задан неориентированный граф (сеть) G с $N+1$ вершинами. Все вершины, кроме $(N+1)$ -й, являются источниками информации. Вершина $N+1$ является стоком (центральным узлом). Путем

μ^k от вершины k к стоку $N+1$ будем называть последовательность вершин $(k, i, j, \dots, N+1)$, через которые проходит данный путь. Для каждой вершины k , $k=1, 2, \dots, N$, необходимо выбрать два пути к стоку, не имеющих общих вершин, кроме k и $N+1$. Заданы потоки информации P_1 и P_2 , которые требуется передать от вершины k к стоку соответственно по первому и второму путям. Дугам графа G приписаны веса, равные их длинам. Для общности рассмотрения заданный граф G дополним до полного дугами с достаточно большим весом.

Цель задачи — выбор топологии сети и маршрутов передачи информации на ней, минимизирующих общую стоимость сети. Общая стоимость сети определяется как сумма стоимостей входящих в нее дуг. Стоимость c_{ij} дуги является функцией от длины дуги (i, j) и суммарного потока информации, передаваемой по ней, — суммы потоков от вершины i к вершине j и от вершины j к вершине i .

Конкретный вид функции стоимости не имеет принципиального значения и может быть произвольным. В дальнейшем предполагается, что функция стоимости возрастает с ростом расстояния и не убывает с ростом суммарного потока. Известно, что оптимальная сеть при различных минимизируемых величинах может быть получена по одному и тому же алгоритму при соответствующем выборе величины c_{ij} .

Введем следующие обозначения: x_{ij}^{1k} — булева переменная, равная 1, если вершина j непосредственно следует за вершиной i на первом пути μ_1^k от вершины k к стоку, и равная 0 в противном случае, $i=1, 2, \dots, N$, $j=1, 2, \dots, N+1$, $i \neq j$, $k=1, 2, \dots, N$; x_{ij}^{2k} — булева переменная, равная 1, если вершина j непосредственно следует за вершиной i на втором пути μ_2^k от вершины k к стоку, и равная 0 в противном случае, $i=1, 2, \dots, N$, $j=1, 2, \dots, N+1$, $i \neq j$, $k=1, 2, \dots, N$; Q_{ij} — суммарный поток информации по дуге (i, j) , определяемый формулой

$$Q_{ij} = \sum_{k=1}^N \sum_{l=1}^2 P_l^k (x_{ij}^{lk} + x_{ji}^{lk}), \quad (5.15)$$

$$i=1, 2, \dots, N, \quad j=2, 3, \dots, N+1, \quad i < j.$$

Тогда рассматриваемую задачу можно сформулировать следующим образом. Необходимо найти значения булевых переменных x_{ij}^{lk} , минимизирующих общую стоимость сети:

$$S = \sum_{i=1}^N \sum_{j=i+1}^{N+1} c_{ij} Q_{ij} \rightarrow \min, \quad (5.16)$$

где Q_{ij} вычисляются по формуле (5.15), при следующих ограничениях:

$$\sum_{\substack{j=1 \\ j \neq k}}^{N+1} x_{kj}^{lk} = 1, \quad \sum_{\substack{i=1 \\ i \neq k}}^N x_{ki}^{lk} = 0, \quad (5.17)$$

$$\sum_{i=1}^N x_{i,N+1}^{lk} = 1, \quad k = 1, 2, \dots, N, \quad l = 1, 2, \quad (5.18)$$

$$\sum_{\substack{i=1 \\ i \neq r}}^N x_{ir}^{lk} = \sum_{\substack{j=1 \\ j \neq r}}^{N+1} x_{rj}^{lk} \leq 1, \quad r, k = 1, 2, \dots, N, \quad l = 1, 2, \quad (5.19)$$

$$\sum_{l=1}^2 \sum_{\substack{i=1 \\ i \neq r}}^N x_{ir}^{lk} \leq 1, \quad r, k = 1, 2, \dots, N. \quad (5.20)$$

Ограничение (5.17) означает, что k — начало как первого, так и второго путей к стоку. Ограничение (5.18) предопределяет, что все пути должны оканчиваться в вершине $N+1$. Ограничение (5.19) указывает на условия сохранения потока (равенство) и на запрет существования петель в путях от вершины к стоку (неравенство). Заметим, что если функция стоимости монотонно возрастает с ростом суммарного потока, то ограничение (5.19) в виде неравенства можно исключить. Ограничения (5.20) реализуют условие непересечения в вершинах первого и второго путей от каждой вершины к стоку.

Данная математическая постановка предполагает наличие $2N^3$ булевых переменных и $5N^2 + 6N$ ограничений, что в практически интересных случаях ($N > 20$) не позволяет рассчитывать на точное решение.

Исключение составляет случай, когда функция стоимости линейна относительно суммарного потока. Поскольку целевая функция (5.16) становится линейной по переменным x_{ij}^{lk} , это позволяет

осуществить декомпозицию исходной задачи на N независимых задач булевого программирования с целевой функцией

$$S_l^k = \sum_{i=1}^N \sum_{j=i+1}^{N+1} c'_{ij} P_l^k x_{ij}^{lk} \quad (5.21)$$

и ограничениями (5.17)–(5.20) при фиксированных значениях k , $k=1,2,\dots,N$, где c'_{ij} — стоимость единицы потока, проходящего по дуге (i, j) .

Последняя задача (при фиксированном k) может быть сформулирована достаточно просто: найти два пути от вершины k к вершине $N+1$, не имеющих общих вершин, кроме k и $N+1$, и движение по которым обеспечивает минимум суммарной стоимости путей. При этом стоимость пути равна стоимости входящих в него дуг, а стоимость дуги (i, j) равна $c'_{ij} P_l^k$ для l -го пути, $l=1,2$.

Математическую постановку задачи с двумя независимыми путями к центральному узлу и одним путем к дополнительному узлу можно записать в следующем виде.

Пусть вершина с номером N является дополнительным узлом. Через μ_3^k обозначим путь от вершины k к ДУ, $k=1,2,\dots,N-1$. При этом для любого $k=1,2,\dots,N-1$ путь μ_3^k не должен содержать вершину $N+1$. Пусть x_{ij}^{3k} — булева переменная, равная 1, если вершина j связана с вершиной i на пути μ_3^k от вершины k к ДУ, и равная 0 в противном случае. Через P_3^k обозначим требуемые величины потоков информации, передаваемой от вершины k к ДУ, $k=1,2,\dots,N-1$.

Тогда, придерживаясь прежних обозначений, суммарный поток на дуге (i, j) определяем по формуле

$$Q_{ij} = \sum_{k=1}^N \sum_{l=1}^2 P_l^k (x_{ij}^{lk} + x_{ji}^{lk}) + \sum_{k=1}^{N-1} P_3^k (x_{ij}^{3k} + x_{ji}^{3k}). \quad (5.22)$$

Для дуг $(i, N+1)$, $i=1,2,\dots,N$, суммарный поток, как и ранее, вычисляют по формуле (5.15).

Рассматриваемую задачу можно сформулировать следующим образом. Найти значения булевых переменных x_{ij}^{lk} , $l=1,2,3$, минимизирующих общую стоимость сети, вычисляемую по формуле (5.16) при ограничениях (5.17)–(5.20) и ограничениях

$$\sum_{\substack{j=1 \\ j \neq k}}^N x_{kj}^{3k} = 1, \quad \sum_{\substack{i=1 \\ i \neq k}}^{N-1} x_{ik}^{3k} = 0, \quad k = 1, 2, \dots, N-1, \quad (5.23)$$

$$\sum_{i=1}^{N-1} x_{iN}^{3k} = 1, \quad k = 1, 2, \dots, N-1, \quad (5.24)$$

$$\sum_{i=1}^{N-1} x_{ir}^{3k} = \sum_{j=2}^{N-1} x_{rj}^{3k} \leq 1, \quad r, k = 1, 2, \dots, N-1, \quad r \neq k. \quad (5.25)$$

Ограничения (5.23)–(5.25) аналогичны ограничениям (5.17)–(5.19). Аналог ограничения (5.20) отсутствует, поскольку к ДУ необходимо построить только один путь от каждой вершины.

Как легко видеть, требование существования путей к ДУ увеличивает число переменных на $(N-1)^2$ и число ограничений на $(N-1)(N+2)$. Однако, как и ранее, в случае, когда функция стоимости дуги от величины суммарного потока линейна, задача распадается на две независимые задачи:

- 1) определение двух непересекающихся путей, направленных к центральному узлу;
- 2) определение путей к вершине N , имеющих минимальную суммарную стоимость.

Математическая постановка первой задачи уже рассматривалась, а вторая, как легко видеть, приводит к задаче построения кратчайших путей от каждой k -й вершины к N -й. В этих двух постановках случайными величинами являются значения P_i^k , так как они получаются в результате прогнозов и, в крайнем случае, должны задаваться в виде интервалов с соответствующими значениями вероятностей. Другой случайной величиной может быть доля информации, передаваемой по прямому (или обходному) пути.

Полученная нелинейная целочисленная задача может быть решена на современных ЭВМ лишь для сети малых размеров: число пунктов равно 10 при 20–30 допустимых трассах. Размеры реальных сетей, как правило, больше. Поэтому иногда проводят линеаризацию данной задачи, приводя ее к целочисленной линейной или просто к задаче линейного программирования. Однако в этом случае для получаемого решения не гарантируется определенная степень приближения критерия затрат к минимуму.

Большая размерность описанных задач и стохастическая природа исходных данных приводят к необходимости разработки эв-

ристических алгоритмов решения. Кроме того, сама специфика структуры сетевых задач позволяет находить более эффективные алгоритмы поиска решения, чем алгоритмы в точных методах решения.

Известен ряд эвристических методов, в которых учитывается специфика поставленной задачи, но в этих алгоритмах для решения задачи используют дуги минимальной длины. Следовательно, алгоритмы ориентированы на расчет сети минимальной длины, что не всегда позволяет найти наилучший результат, тем более этот недостаток будет сказываться при построении коротких сетей, в которых расстояния между узлами малы, так как в этом случае стоимость сети определяется в основном стоимостью аппаратуры и мало зависит от длины трасс.

Алгоритм синтеза сети без ДУ

Алгоритм для решения общей задачи (5.11)–(5.16) состоит из трех этапов:

- I — построение первых путей от всех вершин к стоку;
 - II — построение вторых путей от всех вершин к стоку;
 - III — улучшение сети, полученной на первых двух этапах.
- I. *Построение первых путей от всех вершин к стоку.*

Шаг 1. Строят начальную звездообразную сеть, в которой каждая вершина непосредственно, без промежуточных узлов, соединена со стоком, и считают стоимость такой сети согласно заданной функции стоимости. Если для какой-то вершины связь со стоком отсутствует в исходном графе, то стоимость такой несуществующей дуги полагают равной достаточно большому числу.

Шаг 2. Для каждой вершины k , путь от которой является радиальным путем:

$$\mu_1^k = (k, N + 1),$$

вычисляют экономию стоимости T_k^r , которая будет получена при замене пути μ_1^k на путь

$$\mu_1^k(r) = (k, r, \mu_1^k),$$

и равна разности их стоимостей. При этом промежуточная вершина r принимает значения всех вершин, инцидентных вершине k (т. е. связанных с вершиной k разрешенными дугами),

и вершин пути μ_1^k . Пути от вершины до стока не содержат несуществующих дуг. Затем из всех экономий стоимости T_k^r выбирают максимальную.

На рис. 5.6 приведен пример такой замены. Экономия достигается за счет ликвидации дуги $(k, N+1)$, но при этом может увеличиться стоимость дуг пути μ_1^r , где $\mu_1^r = (r, i_1, i_2, N+1)$, $\mu_1^k = (k, N+1)$, $\mu_1^k(r) = (k, r, i_1, i_2, N+1)$.

Шаг 3. Выбирают максимальное значение T_k^* из всех экономий стоимости T_k , $k=1, 2, \dots, N$. Если $T_k^* \leq 0$, то этап I считают завершенным.

Шаг 4. Для выбранного T_k^* осу-

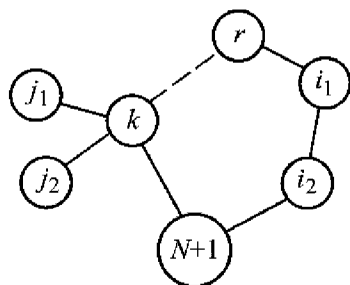


Рис. 5.6. Пример замены дуги

ществляют замену пути $\mu_1^{k^*} = (k^*, N+1)$ на путь $\mu_1^{k^*}(r) = (k^*, r, \mu_1^r)$, на котором достигалась максимальная экономия стоимости T_k^* . Осуществляют замену путей для всех вершин из множества $V(k^*)$, пути от которых проходили по дуге $(k^*, N+1)$, т. е. вместо пути $\mu_1^{V(k^*)} = (V(k^*), \dots, k^*, N+1)$ формируют новый путь $\mu_1^{V(k^*)}(r) = (V(k^*), \dots, k^*, r, \mu_1^r)$. На рис. 5.6 такими вершинами являются вершины j_1 и j_2 . Новый путь, например, для вершины j_1 будет $\mu_1^{j_1}(r) = (j_1, k, r, i_1, i_2, N+1)$ вместо $\mu_1^{j_1} = (j_1, k, N+1)$.

Шаг 5. Экономия стоимости T_k полагают равной нулю для всех $k=1, 2, \dots, N$. Переход к шагу 2.

Шаги 2–5 выполняют до тех пор, пока может быть достигнута экономия при замене какого-либо радиального пути на путь, проходящий через промежуточную вершину. Пересчет экономии стоимости T_k осуществляется после каждой замены радиального пути.

Поскольку каждый раз происходит уменьшение на единицу числа радиальных путей, то этап I конечен.

II. Построение вторых путей от всех вершин к стоку. Идея алгоритма на данном этапе та же — замена радиального второго пути на путь, проходящий через промежуточную вершину.

Шаг 1. Этот шаг аналогичен шагу 1 этапа I. Отличие состоит в следующем: если первый путь от некоторой вершины после завершения этапа I оставался радиальным, то радиальный второй путь прокладывают по несуществующей второй дуге к стоку и его стоимость полагают вдвое большей, чем стоимость просто несуществующей дуги. Смысл этого состоит в том, что именно такие пути выгоднее заменять, а значит, они будут заменены в первую очередь. Кроме того, при расчете стоимости сети учитывают первые пути от всех вершин к стоку.

Шаг 2. Этот шаг аналогичен шагу 2 этапа I. Только теперь существует две возможности замены радиального второго пути от вершины k на путь, проходящий через промежуточную вершину r , а именно:

$$\mu_2^{k,1}(r) = (k, r, \mu_1^r), \quad \mu_2^{k,2}(r) = (k, r, \mu_2^r),$$

т. е. новый путь идет от вершины k к вершине r , а затем либо по первому, либо по второму пути от вершины r к стоку (конечно, при этом второй путь μ_2^r должен быть «реальным»).

Обозначим через $V(k)$ множество вершин, вторые пути которых проходят по дуге $(k, N+1)$, при этом $k \in V(k)$. Тогда путь μ можно заменить на путь $\mu_2^{k,1}$ или $\mu_2^{k,2}$ только в том случае, если для любого $j \in V(k)$

$$\mu_1^j \cap \mu_2^{k,1} \in (k, N+1) \quad \text{или} \quad \mu_1^j \cap \mu_2^{k,2} \in (k, N+1)$$

соответственно. Сказанное выше схематично представлено на рис. 5.7.

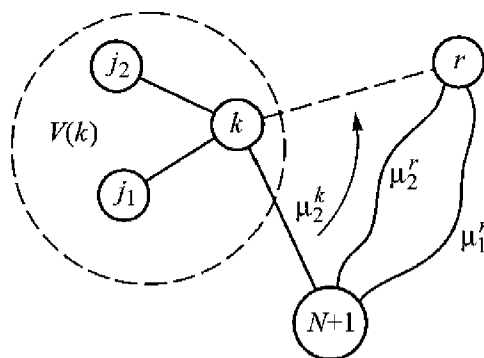


Рис. 5.7. Пример замены пути

Шаги 3–5 аналогичны тем же шагам I этапа. Они выполняются до тех пор, пока замена какого-либо радиального пути на путь,

проходящий через промежуточную вершину, может дать положительную экономию стоимости.

После выполнения этапов I и II мы получим сеть с размеченными на ней маршрутами первых и вторых путей от каждой вершины к стоку. Эта сеть с маршрутами удовлетворяет всем условиям задачи, т. е. является допустимой. Кроме того, она обладает следующими особенностями.

1. Каждая вершина $k=1,2,\dots,N$ имеет не менее двух инцидентных вершин сети, поскольку осуществляет два непересекающихся по вершинам, а следовательно, и по дугам пути. *Инцидентными вершинами* называют вершины, связанные разрешенными дугами.

2. По каждой дуге (i,j) сети согласно выбранным маршрутам обоих путей происходит передача информации либо только в одну сторону, например от вершины i к вершине j , либо в обе стороны. В первом случае дугу (i,j) ориентируем от вершины i к вершине j , а во втором случае дуга (i,j) будет ориентирована в обе стороны. После ориентации сети описанным образом получим, что каждая вершина сети имеет ровно две выходящих из нее дуги (кроме, конечно, стока). Входящих дуг может быть произвольное число. Это следует из алгоритма замены радиального пути на путь, проходящий через промежуточную вершину.

3. Если по дуге (i,j) от вершины i к вершине j идет поток информации P_{ij} , то в дальнейшем он полностью проходит по первому или второму пути от вершины j к стоку. Этот факт также следует из алгоритмов этапов I и II.

Особенности сети, отмеченные в пп. 2 и 3, облегчают реализацию следующего этапа III.

III. *Улучшение сети, полученной на первых двух этапах.* Если после выполнения первых двух этапов каждая вершина сети имеет ровно две инцидентные вершины (кроме, разумеется, стока), то полученная сеть не будет обладать избыточностью. В этом случае возможность улучшения сети заключается только в попытке заменить какой-либо радиальный первый путь. Если некоторые вершины имеют более двух инцидентных вершин, то полученная сеть обладает известной избыточностью. На данном этапе осуществляют попытку улучшить сеть (в смысле ее общей стоимости) в обоих случаях.

Шаг 1. Составляют множество дуг — претендентов на замену. В него включают все дуги (i, j) , ориентированные лишь в одну сторону, $i = 1, 2, \dots, N$, $j = 1, 2, \dots, N + 1$, $i \neq j$. Обозначим это множество через A . Для каждой дуги $(i, j) \in A$ составляют множество вершин $V(i, j)$, пути от которых проходят по дуге (i, j) , и множество путей $L(i, j)$ (первые или вторые пути от вершин множества $V(i, j)$, проходящие по дуге (i, j)).

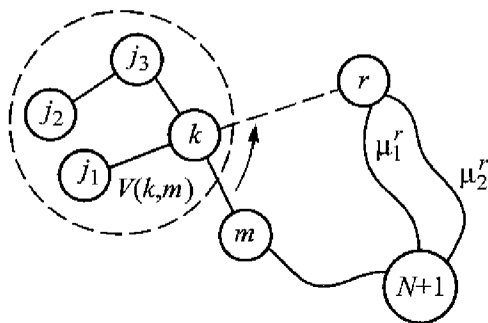


Рис. 5.8. Пример замены дуг с изменением маршрутов первых или вторых путей

Шаг 2. Для каждой дуги $(k, m) \in A$ (рис. 5.8) вычисляют экономию стоимости, которая может быть получена при замене дуги (k, m) на дугу (k, r) с соответствующим изменением маршрутов первых или вторых путей для всех вершин $j \in V(k, m)$ на путь $\mu_l^{j,1} = (j, \dots, k, r, \mu_1^r)$ или путь $\mu_l^{j,2} = (j, \dots, k, r, \mu_2^r)$, т. е. новый путь идет, как и раньше, до вершины k , а далее через вершину r и по первому или второму пути от вершины r к стоку. Такую замену можно осуществить лишь в том случае, когда:

- 1) для любой вершины $j \in V(k, m)$ другой путь не имеет общих вершин с μ_l^r , кроме $(N + 1)$ -й вершины;
- 2) путь μ_l^r не содержит вершин из $V(k, m)$, т. е.

$$\mu_l^r \cap V(k, m) = \emptyset.$$

Экономии стоимости пути при этом достигают за счет ликвидации дуги (k, m) , а также за счет уменьшения потока на величину $P_{k,m}$ для остальных дуг пути μ_l^r . Однако при этом может произойти увеличение стоимости сети от включения в нее дуги (k, r) или от возрастания стоимости на величину $P_{k,m}$ потока по дуге (k, r) и по дугам путей μ_1^r или μ_2^r .

Шаг 3. Выбирают дугу $(k, m) \in A$ и соответствующую вершину r , на которых достигается максимальная экономия стоимости пути. Если максимальная экономия положительна, то выполняют шаг 4, в противном случае происходит завершение этапа III.

Шаг 4. Осуществляют модификацию сети, маршрутов путей, множеств A , V и L в соответствии с выбранными на шаге 3 дугой (k, m) , вершиной r и путем от вершины r к стоку. После этого происходит переход к шагу 2.

После завершения этапа III работа эвристического алгоритма считается законченной.

Алгоритм синтеза сети с ДУ

Схема эвристического алгоритма для решения задачи (5.15)–(5.25) приведена на рис. 5.9.

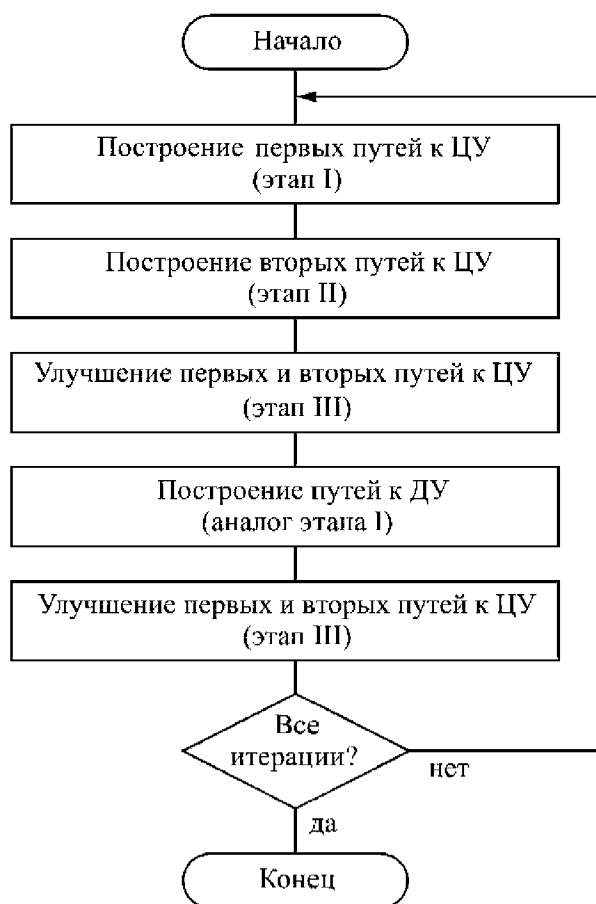


Рис. 5.9. Схема эвристического алгоритма

На каждой итерации осуществляют последовательное выполнение описанных выше трех этапов эвристического алгоритма построения двух непересекающихся путей к центральному узлу, после чего с помощью этапа III эвристического алгоритма осуществляют построение путей к ДУ и проводят попытку уменьшить стоимость сети с помощью замены некоторых маршрутов путей к центральному узлу с учетом новых путей к ДУ. На каждой итерации (кроме первой) при построении первых и вторых путей к центральному узлу учитывают потоки информации, протекающие по сети, полученные на путях к ДУ в предыдущей итерации. При построении путей к ДУ учитывают потоки информации, протекающие по сети согласно построенным ранее первым и вторым путям к центральному узлу. Для учета существующих на сети потоков информации в эвристическом алгоритме изменяют лишь вычисление экономии стоимости при замене одного пути на другой.

Итерационный процесс продолжают до тех пор, пока осуществление очередной итерации не приведет к изменению сети и потоков на ней или пока не будет исчерпан заданный лимит итераций. В качестве решения задачи принимают тот вариант сети, при котором достигается минимальное значение общей стоимости сети. Как показал опыт многих экспериментальных расчетов, требуемое для достижения минимума число итераций зависит от соотношения величин $\min\{P_k^1, P_k^2\}$ и P_k^3 . Чем ближе эти значения, тем больше требуется итераций (но их число, например, не превосходило 5 при $N = 25$). Если же необходимое число путей к запасному узлу существенно меньше необходимого числа путей к центральному узлу, то лучший результат, как правило, достигается на первой итерации.

Построение сети с учетом ограничений по транзитам

Доработку изложенного алгоритма для построения сети и распределения каналов с учетом ограничения по транзитам осуществляют следующим образом.

Под транзитностью пути понимают число дуг в этом пути. Ограничение по транзитам формулируют следующим образом: построить сеть, в которой число дуг в маршрутах прямых и обходных путей к ЦУ и путей к ДУ не будет превышать некоторого заданного числа (обозначим его τ). Для этого достаточно на шаге 2 этапов I–III описанного алгоритма рассматривать в качестве новых только пути, удовлетворяющие условию транзитности.

Для вершины i необходимо знать:

1) число дуг $A_1(i)$ на участке пути между вершиной i и вершиной, наиболее удаленной от i по количеству дуг, число дуг $A_2(i)$ от вершины i к ЦУ через прямой или обходной путь и число дуг $A_3(i)$ от вершины i к ДУ;

2) число дуг в прямом и обходном путях к ЦУ и пути к ДУ от i -й вершины; обозначим эти числа соответственно $B_1(i)$, $B_2(i)$, $B_3(i)$.

В процессе учета ограничения по транзитам при построении прямых путей к ЦУ при замене пути $\mu_1^k = (k, N+1)$ на путь $\mu_1^k(r) = (k, r, \mu_1^k)$ в качестве r берем лишь те вершины, для которых выполнено неравенство

$$A_1(k) + B_1(r) < \tau.$$

Если от вершины k , которая соединена с центром фиктивной дугой, нет маршрута, удовлетворяющего ограничению по транзитам, то путь от вершины k строят без ограничения по транзитам после того, как построены все маршруты прямых путей, удовлетворяющие ограничению по транзитам.

При построении нового маршрута обходного пути с учетом ограничения по транзитам для вершины k в качестве вариантов замены рассматривают лишь такие вершины r , для которых выполнены следующие требования:

1) если новый маршрут пройдет через прямой путь от r , то

$$A_1(k) + B_1(r) < \tau;$$

2) если новый маршрут пройдет через обходной путь от r , то

$$A_2(k) + B_2(r) < \tau.$$

Если вершина k соединена с центром фиктивной дугой и нет маршрута, удовлетворяющего требованию транзитности, маршрут от вершины k строят без ограничения по транзитам после построения всех маршрутов обходных путей, которые удовлетворяют ограничению по транзитам.

Учет ограничения по транзитам на этапе оптимизации построения прямых и обходных путей показан на рис. 5.10. Здесь W — множество вершин, пути от которых проходят по дуге (i, k) . Замена дуги (i, k) на дугу (i, j) на этом этапе происходит при следующих условиях:

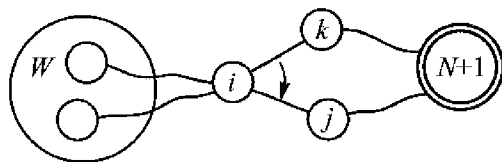


Рис. 5.10. Учет ограничения по транзитам

1) транзитность пути от вершины k (старый путь) не меньше транзитности пути от вершины j (новый путь); замену производят, если при этом есть выгода (если экономия стоимости пути положительна);

2) транзитность пути от вершины j превышает транзитность пути от вершины k ;

3) если ранее построенный путь от вершины i не удовлетворял ограничению по транзитам, то замена происходит в том случае, когда она выгодна;

4) если путь от вершины i удовлетворял ограничению по транзитам, но путь хотя бы от одной вершины из множества W (рис. 5.10) многотранзитный, то замену производят тогда и только тогда, когда, во-первых, она финансово выгодна и, во-вторых, не появляются новые недопустимые по транзитности маршруты (прежние могут оставаться многотранзитными);

5) если путь от вершины i и путь от любой другой вершины из множества W удовлетворяют ограничению по транзитам, то замену производят, когда она выгодна и не создает недопустимых по транзитности маршрутов.

При построении путей к ДУ учет ограничения по транзитам проводится следующим образом.

При замене пути $\mu_3^k = (k, N)$ на путь $\mu_3^k(r) = (k, r, \mu_3^k)$ проверяют выполнение неравенства

$$A_3(k) + B_3(r) < \tau.$$

После каждого изменения маршрутов производят пересчет чисел A_3 и B_3 тех вершин, в маршруты которых были внесены изменения.

Рассмотренный эвристический алгоритм не накладывает существенных ограничений на функцию стоимости дуг и может быть использован для решения широкого круга задач:

– построение сети кратчайшей длины или минимальной стоимости;

- построение сети, минимальной по величине задействованных канало-километров;
- распределение маршрутов на сети при наличии ограничений пропускных способностей дуг с целью достижения максимальной суммарной свободной емкости.

Кроме того, алгоритм позволяет легко учесть дополнительные требования по качеству передачи информации, например ограничение по транзитам.

Примеры синтеза сети с детерминированными исходными данными

Для иллюстрации работы эвристического алгоритма рассмотрим граф, который приведен на рис. 5.11, где изображены допустимые трассы прокладки кабеля с указанием протяженности каждой дуги графа в километрах. Центральным узлом считают узел 1, а дополнительным узлом — узел 2. Потребности в каналах первого и второго путей к ЦУ и пути к ДУ для каждого узла сети приведены в табл. 5.10. Стоимость 1 км дуги зависит от стоимости системы передачи, установленной на дуге, и составляет 2 тыс. усл. ед. при емкости 120 каналов, 4 тыс. усл. ед. при емкости 240 каналов и 5 тыс. усл. ед. при емкости 600 каналов. Такой

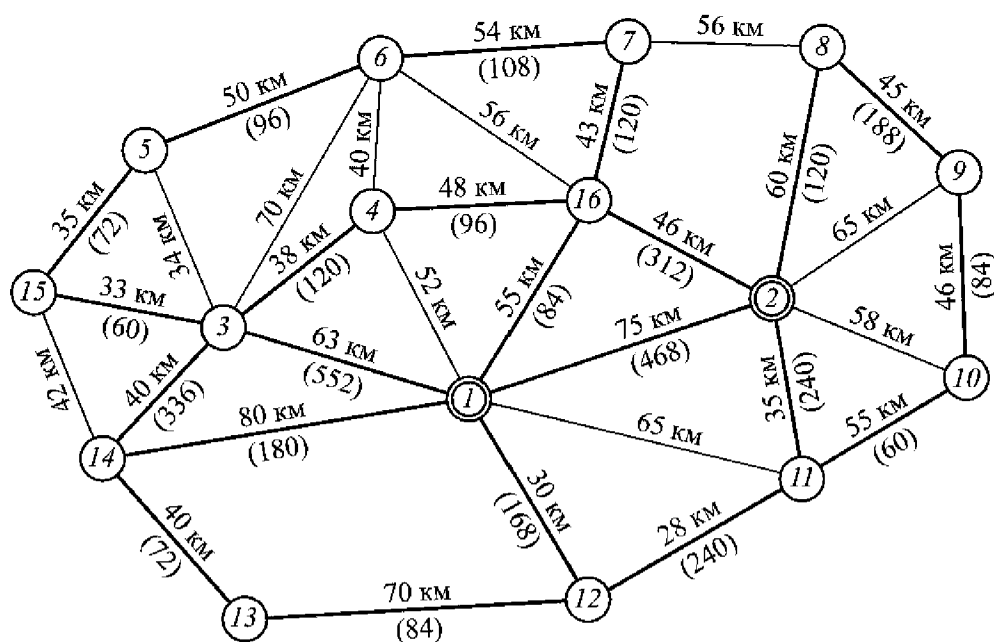


Рис. 5.11. Допустимые трассы прокладки кабеля

вид функции стоимости линий связи в общем виде соответствует реальной ситуации. Кроме того, в проектируемой сети между узлами 1 и 2 уже имеется система передачи емкостью 480 каналов, между узлами 1 и 3 — 600 каналов, между узлами 2 и 8, 3 и 15 — по 120 каналов.

Таблица 5.10

Исходные данные для задачи синтеза сети

Наименование пути	Потребности узлов (в каналах)															
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Первый путь к ЦУ	60	120	220	24	24	12	24	36	24	96	36	12	240	12	48	
Второй путь к ЦУ	36	60	72	12	24	12	24	24	12	60	36	12	120	12	48	
Путь к ДУ	0	24	24	12	12	12	12	12	12	24	12	12	36	12	48	

В табл. 5.11 приведены результаты расчетов по описанному алгоритму. Суммарное число свободных каналов на сети вычисляют как суммарную разность между емкостью установленной системы передач и потоком информации на каждой дуге. Дуги, включенные в оптимальную сеть, изображены на рис. 5.11 жирными линиями, в скобках указано число используемых каналов на дуге. Время счета примера составляет меньше 1 мин.

Таблица 5.11

Решение задачи синтеза сети

Параметр	Эвристический алгоритм
Стоимость сети, усл. ед.	2280
Число дуг, шт.	22
Протяженность сети, км	1069
Суммарная свободная емкость каналов	1260

На рис. 5.12 приведена сеть, полученная в результате работы эвристического алгоритма построения двух независимых путей к ЦУ без ДУ. Стоимость полученной сети составила 2 124 тыс. усл. ед. Маршруты первых и вторых путей сети существенно отличаются от маршрутов первых и вторых путей, приведенных на рис. 5.11, т. е. при выполнении итерационного процесса маршруты первых и вторых путей претерпевают значительные изменения. Следует отметить, что требование наличия путей к ДУ увеличивает на 15,7 % общую потребность в линиях связи по перво-

му и второму путям к ЦУ, а суммарную стоимость сети увеличивает лишь на 7,3 %.

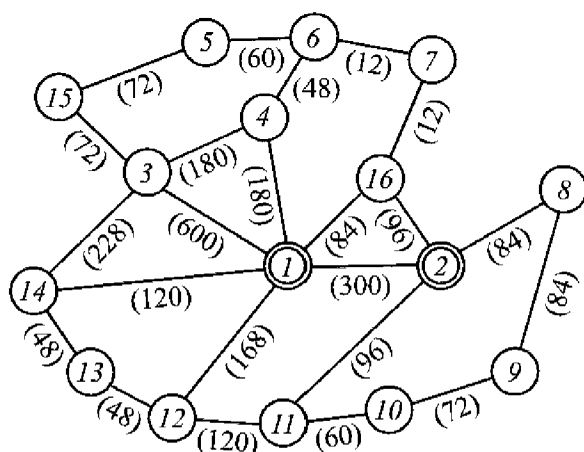


Рис. 5.12. Синтез сети согласно эвристическому алгоритму (два независимых пути к ЦУ без ДУ)

Для тех же исходных данных с ЦУ и ДУ был проведен расчет в случае, когда стоимость дуги не зависит от расстояния, а определяется только используемой системой передачи. Стоимости системы передач были взяты следующие: емкостью 120 каналов — 2 тыс. усл. ед.; 240 каналов — 4 тыс. усл. ед.; 600 каналов — 5 тыс. усл. ед. Сеть, полученная в результате работы предложенного алгоритма, приведена на рис. 5.13. Число дуг сети равно 23. Протяженность сети составляет 1 132 км, суммарная свободная емкость — 2 460 каналов.

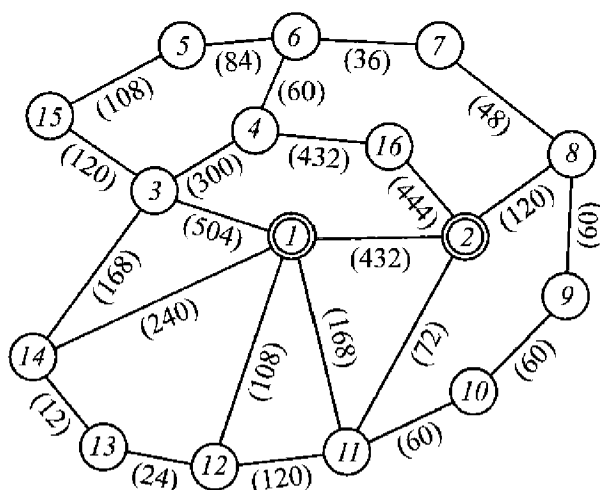


Рис. 5.13. Синтез сети согласно эвристическому алгоритму, когда стоимость дуги не зависит от расстояния

§ 5.8. Методы внутренней точки для решения задачи математического программирования

Рассмотрим общую задачу математического программирования, не содержащую ограничений в виде равенств: минимизировать функцию $f(x)$ при ограничениях $g_i(x) \geq 0$, $i = 1, 2, \dots, m$.

Пусть вблизи локального минимума x^* этой задачи существует окрестность, в которой найдется такая точка x^0 , что $g_i(x^0) \geq 0$, $i = 1, 2, \dots, m$, при этом выполнены условия строгой дополняющей нежесткости:

$$u_i(x^*) > 0, \text{ если } g_i(x^*) = 0, \quad i = 1, 2, \dots, m.$$

Видоизменим достаточные условия локального минимума в точке x^* , сформулированные в § 1.5. Предположим, что при малом параметре $r > 0$ в точке $(x(r), u(r))$ вблизи точки (x^*, u^*) выполнены условия

$$g_i(x) \geq 0, \quad u_i g_i(x) = r > 0, \quad u_i \geq 0, \quad i = 1, 2, \dots, m,$$

$$\nabla f(x) - \sum_{i=1}^m u_i \nabla g_i(x) = 0.$$

Из второго условия выразим u_i и подставим это выражение в последнее равенство:

$$\nabla f(x(r)) - \sum_{i=1}^m \frac{r}{g_i(x(r))} \nabla g_i(x(r)) = 0.$$

Непосредственной проверкой легко установить, что левая часть этого выражения — градиент функции

$$L(x, r) = f(x) - r \sum_{i=1}^m \ln g_i(x),$$

обращающийся в нуль в точке $x(r)$. Значения $x(r)$ стремятся к x^* при r , стремящемся к нулю. Функцию $L(x, r)$ в таком виде называют *логарифмической штрафной функцией*.

Аналогично получаем другой вид штрафной функции $L_1(x, r)$, полагая $u_i = \lambda_i^2$. При условии, что $\lambda_i^2 g_i(x) = r > 0$, $i = 1, 2, \dots, m$, имеем

$$\nabla f(x(r)) - \sum_{i=1}^m \frac{r^2}{g_i^2(x(r))} \nabla g_i(x(r)) = 0,$$

и функция $L_1(x, r)$ примет вид

$$L_1(x, r) = f(x) + r^2 \sum_{i=1}^m \frac{1}{g_i(x)}.$$

Задавая последовательность $\{r_k\}$, стремящуюся к нулю, получаем последовательность $\{x(r_k)\}$, сходящуюся к x^* . С помощью новых функций $L(x, r)$ мы свели задачу определения условного экстремума (задачу математического программирования) к задаче поиска безусловного экстремума функции $L(x, r)$. Точнее, решение задачи математического программирования свели к рассмотрению семейства функций, зависящих от параметра r и обладающих следующими свойствами:

- 1) в окрестности оптимальной точки они близки к заданной минимизируемой функции;
- 2) каждая функция из построенного семейства достаточно быстро возрастает при приближении к границе допустимой области из внутренней части допустимой области.

К минимизируемой функции исходной задачи мы добавили несколько слагаемых, называемых *штрафными (барьерными) функциями*. Каждое слагаемое зависит от параметра r и от функции одного из ограничений. При фиксированном значении параметра r второе слагаемое стремится к бесконечности при стремлении к нулю его аргумента. Каждую функцию семейства подвергают процедуре безусловной минимизации, и это не выведет оптимальную точку x^* за пределы допустимой области. Подобные методы названы *методами внутренней точки*.

В задаче математического программирования при наличии ограничений-равенств допустимая область (в смысле понятия области) не существует и метод внутренней точки не применим.

Рассмотрим некоторые примеры перехода от задачи математического программирования к задаче безусловной минимизации методом внутренней точки.

Пример. Минимизировать функцию

$$f(x) = x_1 + x_2$$

при ограничениях

$$g_1(x) = -x_1^2 + x_2 \geq 0,$$

$$g_2(x) = x_1 \geq 0.$$

Решение. Построим логарифмическую штрафную функцию

$$L(x, r) = x_1 + x_2 - r \ln(-x_1^2 + x_2) - r \ln x_1.$$

Определим точки минимума $L(x, r)$ аналитически, так как $L(x, r)$ дифференцируема в рассматриваемой области. Для нахождения $x_1(r)$ и $x_2(r)$ получим систему уравнений

$$\frac{\partial L}{\partial x_1} = 1 + \frac{r \cdot 2x_1}{-x_1^2 + x_2} - \frac{r}{x_1} = 0,$$

$$\frac{\partial L}{\partial x_2} = 1 - \frac{r}{-x_1^2 + x_2} = 0.$$

Отсюда найдем

$$x_1(r) = \frac{-1 \pm \sqrt{1 + 8r}}{4}$$

(здесь оставим только знак $+$, так как $x_1 \geq 0$),

$$x_2(r) = \frac{(-1 \pm \sqrt{1 + 8r})^2}{16} + r.$$

Заметим, что значения $x_1(r)$ и $x_2(r)$ удовлетворяют условию положительной определенности матрицы $\nabla^2 L$. Пусть r последовательно принимает значения 1,0; 0,5; 0,25; 0,1. Соответственно получим последовательности значений

$$x_1(r) = 0,50; 0,309; 0,183; 0,085;$$

$$x_2(r) = 1,25; 0,595; 0,283; 0,107,$$

сходящиеся при $r \rightarrow 0$ к точке $(0,0)$. Графическое решение данной задачи представлено на рис. 5.14.

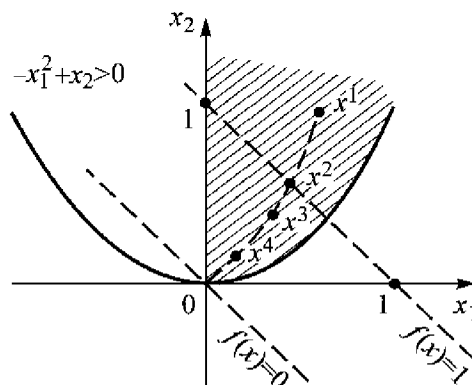


Рис. 5.14. Графическое решение задачи математического программирования методом внутренней точки

В общем случае в задачах (при слабых условиях регулярности) существует последовательность безусловных локальных минимумов, сходящаяся к каждому из условных локальных минимумов.

Рассмотрим еще пример решения задачи линейного программирования методом внутренней точки.

Пример. Минимизировать x_1 при условии $g_1(x) = x_1 > 0$.

Решение. Построим логарифмическую штрафную функцию

$$L(x, r) = x_1 - r \ln x_1.$$

Для существования минимума должно выполняться необходимое условие

$$\frac{dL}{dx_1} = \frac{\partial L}{\partial x_1} = 1 - \frac{r}{x_1} = 0.$$

Откуда получаем $x_1(r) = r$.

Из условия $\frac{d^2 L}{dx_1^2} = \frac{r}{x^2} > 0$ следует, что для $r < 1$ безусловный минимум будет при $x(r) = r$.

§ 5.9. Методы внешней точки для решения задачи математического программирования

Попытаемся приблизиться к оптимальной точке из *недопустимой* области. По аналогии с методами внутренней точки надо выбрать функцию $O(g(x))$ так, чтобы каждая из функций $T(x, t_k) = f(x) + t_k O(g(x))$ имела конечный минимум (при каждом $t_k > 0$). Это достигается, если для любой неограниченной последовательности значений t_k функция $t_k O(g(x))$ стремится к $+\infty$ быстрее, чем $f(x)$ стремится к $-\infty$. Поэтому $\{t_k\}$ — возрастающая неограниченная последовательность положительных чисел, или $t_k = 1/r_k$, если $0 < r_k < 1$.

Преобразуем достаточные условия локального минимума задачи математического программирования следующим образом:

1) рассмотрим ограничения в ослабленной форме:

$$g_i(x) \geq -r, \quad i = 1, 2, \dots, m, \quad r > 0; \quad (5.26)$$

2) преобразуем условие дополняющей нежесткости $u_i^* g_i(x^*) = 0$ так, чтобы оно имело смысл для отрицательных значений $g_i(x)$ и

сводилось к исходному условию при $r \rightarrow 0$ (это означает, что $u_i(x^*) = 0$ при $g_i(x^*) \neq 0$; если $g_i(x^*) = 0$, то $u_i(x^*)$ принимает любые значения), т. е.

$$u_i(r) = -\min \{0; g_i(x(r))\}. \quad (5.27)$$

Отсюда если $r > 0$ мало, $x(r)$ удовлетворяет условию (5.26) и $g_i(x(r)) \neq 0$ для некоторых i , то $u_i \equiv 0$ и $\lim_{r \rightarrow 0} u(r) = u_i(x^*) = 0$. Если $g_i(x(r)) < 0$, то $\lim_{r \rightarrow 0} (-\min \{0, g_i(x(r))\}) = 0$, так как из неравенства (5.26) следует, что $x(r)$ при $r \rightarrow 0$ принадлежит допустимой области. В силу (5.26) $g_i(x(r)) \geq -r$, а из равенства (5.27) вытекает, что $\lim_{r \rightarrow 0} u_i(r)r = 0$, поэтому $\lim_{r \rightarrow 0} u_i(r)g_i(x(r)) = 0$.

Условие дополняющей нежесткости в данном случае выполняется в пределе при $r \rightarrow 0$ (из соотношения (5.27) следует, что $u_i \geq 0$, $i = 1, 2, \dots, m$).

Аналогично преобразуют другие достаточные условия:

$$u_i(r) \geq 0, \quad i = 1, 2, \dots, m,$$

$$\nabla f(x(r)) - t \sum_{i=1}^m u_i(r) \nabla g_i(x(r)) = 0, \quad t = \frac{1}{r}, \quad (5.28)$$

и для каждого вектора y , для которого $y^T \nabla g_i(x(r)) = 0$ при всех $i \in D^* = \{i \mid u_i^* > 0\}$, справедливо неравенство

$$y^T \left\{ \nabla^2 f(x(r)) - \sum_{i \in D^*} u_i(r) \nabla^2 g_i(x(r)) \right\} y > 0.$$

Подставляя (5.27) в равенство (5.28), получим

$$\nabla f(x(r)) + \sum_{i=1}^m \frac{1}{r} \min \{0; g_i(x(r))\} \nabla g_i(x(r)) = 0.$$

Опять непосредственной проверкой убеждаемся в том, что функция, для которой выполняются приведенные достаточные условия существования точки минимума, имеет вид

$$T(x(r)) = f(x) + \sum_{i=1}^m \frac{1}{2r} (\min \{0; g_i(x(r))\})^2.$$

Это и есть функция, минимизируемая методом внешней точки. В нее могут входить ограничения и в виде неравенств, и в виде равенств. Можно доказать, что все необходимые условия

минимума этой функции выполняются, например $\nabla^2 T$ есть положительно определенная матрица. Причем для ограничений-неравенств имеют место соотношения

$$(\min \{0; g_i(x(r))\})^2 \equiv \left[\frac{g_i - |g_i|}{2} \right]^2.$$

Если среди ограничений в задаче математического программирования есть ограничения-равенства $h_j(x) = 0$, $j = 1, 2, \dots$, то можно записать

$$g_{j1}(x) = h_j(x) \geq 0, \quad g_{j2}(x) = -h_j(x) \geq 0,$$

откуда $g_{j1} = -g_{j2}$, или

$$\frac{1}{r} \left\{ \left[\frac{g_{j1} - |g_{j1}|}{2} \right]^2 + \left[\frac{-g_{j1} - |-g_{j1}|}{2} \right]^2 \right\} = \frac{g_{j1}^2}{r} = \frac{h_j^2(x(r))}{r}.$$

Рассмотрим пример применения метода внешней точки для решения задачи математического программирования.

Пример. Минимизировать функцию

$$f(x) = -x_1 x_2$$

при ограничениях

$$g_1(x) \equiv -x_1 - x_2^2 + 1 \geq 0, \quad g_2(x) \equiv x_1 + x_2 \geq 0.$$

Решение. Составим функцию штрафа для метода внешней точки:

$$T(x, r) = -x_1 x_2 + \frac{(-x_1 - x_2^2 + 1 - |-x_1 - x_2^2 + 1|)^2}{4r} + \frac{(x_1 + x_2 - |x_1 + x_2|)^2}{4r}.$$

Из необходимых условий экстремума $\frac{\partial T}{\partial x_1} = 0$, $\frac{\partial T}{\partial x_2} = 0$ определим последовательности значений x_1 и x_2 , сходящиеся к решению.

Зададим последовательность значений r : 1,0; 0,5; 1/3; 0,1; получим соответствующие последовательности значений $x_1(r)$: 0,89; 0,77; 0,73; 0,67; $x_2(r)$: 0,64; 0,62; 0,61; 0,58.

Последовательность значений $x_1(r)$ сходится к $2/3$, а последовательность значений $x_2(r)$ — к $\sqrt{3}/3$ (рис. 5.15).

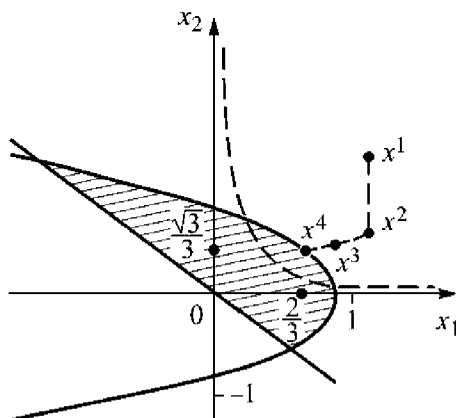


Рис. 5.15. Графическое решение задачи математического программирования методом внешней точки

Метод внутренней точки и метод внешней точки основаны на разных принципах: в первом случае штрафной член препятствует нарушению ограничений, во втором — он предотвращает блуждание точек слишком далеко от допустимой области.

§ 5.10. Комбинированный метод внутренней и внешней точек

Введем параметры r и $t = 1/r$ и рассмотрим комбинированную функцию

$$V(x, r, t) = f(x) + s(r)L(x) + p(t)T(x),$$

где $s(r)$ — функция от r , $r \rightarrow 0$, для метода внутренней точки; $L(x)$ — функция штрафа для метода внутренней точки; $p(t)$ — функция от t , $t \rightarrow \infty$, для метода внешней точки; $T(x)$ — функция штрафа для метода внешней точки; $f(x)$ — целевая функция задачи математического программирования.

Покажем на примере, как применяется комбинированная функция $V(x, r, t)$ для решения задачи математического программирования.

Пример. Минимизировать функцию

$$f(x) = \ln x_1 - x_2$$

при следующих ограничениях:

$$g(x) = x_1 - 1 \geq 0,$$

$$h(x) = x_1^2 + x_2^2 - 4 = 0, \text{ или } x_2^2 = 4 - x_1^2.$$

Решение. Построим комбинированную функцию

$$V(x, r, t) = \underbrace{\ln x_1 - x_2}_{f(x)} - \underbrace{r \ln(x_1 - 1)}_{s(r)L(x)} + \underbrace{r^{-1}(x_1^2 + x_2^2 - 4)^2}_{p(t)T(x)}.$$

С помощью этой комбинированной функции мы избежали использования модулей $|g(x)|$, которые появляются в методе внешней точки. Из необходимых условий существования минимума функции $V(x, r, t)$ имеем

$$\frac{\partial V}{\partial x_1} = \frac{1}{x_1} - \frac{r}{x_1 - 1} + 4x_1 r^{-1}(x_1^2 + x_2^2 - 4) = 0,$$

$$\frac{\partial V}{\partial x_2} = -1 + 4x_2 r^{-1}(x_1^2 + x_2^2 - 4) = 0.$$

Отсюда получаем

$$4r^{-1}(x_1^2 + x_2^2 - 4) = \frac{1}{x_2} = \frac{1}{\sqrt{4 - x_1^2}}.$$

Зададим последовательность значений для r : 1,0; 1/4; 1/16; 1/64; 1/256, и получим соответствующие ей последовательности значений:

$$\begin{array}{lllll} x_1(r): & 1,553; & 1,159; & 1,040; & 1,010; & 1,002, \\ x_2(r): & 1,334; & 1,641; & 1,711; & 1,727; & 1,731, \\ V: & -0,2648; & -1,0285; & -1,4693; & -1,6447; & -1,7048. \end{array}$$

Последовательности значений x_1 и x_2 сходятся к оптимальному решению $(1, \sqrt{3})$. Графическое решение задачи математического программирования комбинированным методом приведено на рис. 5.16.

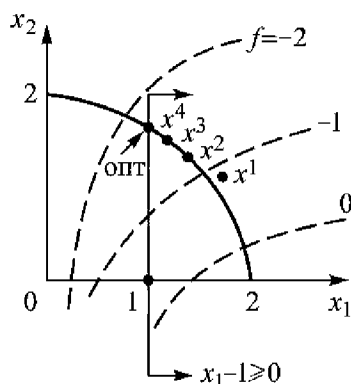


Рис. 5.16. Графическое решение задачи математического программирования с помощью комбинированного метода

§ 5.11. Метод проекции градиента

Известны и другие методы безусловной оптимизации для решения задач математического программирования. Одним из таких методов является *метод проекции градиента* (МПГ). Метод базируется на методе градиентного спуска безусловной оптимизации;

МПП — численный метод условной оптимизации для нелинейных задач. Рассмотрим задачу

$$\min \{f(x)\} \text{ при } x \in D, \quad (5.29)$$

где D — замкнутое выпуклое множество в \mathbb{R}^n , $f(x)$ — дифференцируемая функция на D .

Проекцией точки a на множество D называется точка $a_{\text{пр}}$, ближайшая к a среди всех точек D , т. е. $a_{\text{пр}}$ является решением задачи проектирования:

$$\varphi(x) = \|x - a\|^2 \rightarrow \min, \quad x \in D. \quad (5.30)$$

Свойства проекции точки на множество D определяются следующими леммами и теоремой.

Лемма 1. Пусть D — замкнутое выпуклое множество в \mathbb{R}^n . Тогда:

1) проекция $a_{\text{пр}}$ любой точки $a \in \mathbb{R}^n$ существует и единственна;

2) точка \bar{x} является проекцией точки a на множество D ($\bar{x} = a_{\text{пр}}$) в том и только в том случае, если

$$(\bar{x} - a, a - \bar{x}) \geq 0 \quad \text{при всех } x \in D;$$

3) для любых точек $a_1, a_2 \in \mathbb{R}^n$ справедливо неравенство

$$\|a_{1\text{пр}} - a_{2\text{пр}}\| \leq \|a_1 - a_2\|,$$

т. е. оператор проектирования обладает свойством нестяжения.

Доказательство. 1) Функция $\varphi(x)$ является строго выпуклой, поэтому решение задачи (5.30) существует и единственно.

2) На выпуклом множестве и при выпуклой функции $\varphi(x)$, если x^* — локальное решение задачи (5.29), должно выполняться условие

$$(\varphi'(x^*), x - x^*) \geq 0, \quad x \in D.$$

Производная $\varphi'(x)$ равна $2(x - a)$, т. е. $2(x^* - a, a - x^*) \geq 0$.

3) В силу утверждения 2) имеем

$$(a_{1\text{пр}} - a_1, a_{2\text{пр}} - a_{1\text{пр}}) \geq 0, \quad (a_{2\text{пр}} - a_2, a_{1\text{пр}} - a_{2\text{пр}}) \geq 0.$$

Суммируя эти неравенства, получим

$$(a_{1\text{пр}} - a_1 - a_{2\text{пр}} + a_2, a_{2\text{пр}} - a_{1\text{пр}}) \geq 0,$$

откуда в силу неравенства Коши — Буняковского имеем

$$\|a_{1\text{пр}} - a_{2\text{пр}}\|^2 \leq (a_{2\text{пр}} - a_{1\text{пр}}, a_2 - a_1) \leq \|a_{1\text{пр}} - a_{2\text{пр}}\| \|a_1 - a_2\|.$$

Необходимые и достаточные условия оптимальности в выпуклой задаче в терминах проекций формулируются следующим образом.

Лемма 2. Пусть множество D выпукло и замкнуто, функция $f(x)$ выпукла на D и дифференцируема в точке $x^* \in D$. Тогда x^* является решением задачи (5.29) в том и только в том случае, если

$$x^* = (x^* - \alpha f'(x^*))_{\text{пр}}$$

при произвольном $\alpha > 0$.

Доказательство. Согласно утверждению 2) леммы 1 приведенное равенство эквивалентно условию

$$(x^* - (x^* - \alpha f'(x^*)), x - x^*) \geq 0 \quad \forall x \in D.$$

Поэтому

$$(f'(x^*), x - x^*) \geq 0 \quad \forall x \in D.$$

Это неравенство является условием того, что x^* — решение задачи (5.29). Лемма доказана.

В методе градиентного спуска определяется последовательность точек

$$x^{k+1} = x^k - \alpha_k f'(x^k), \quad k = 0, 1, 2, \dots,$$

в методе проекции градиента на каждой k -й итерации требуется производить операцию проецирования точки на множество D , т. е. решать задачу (5.30) при

$$a = x^k - \alpha_k f'(x^k).$$

В качестве очередной точки приближения к решению задачи (5.29) выбирается точка

$$x^{k+1} = (x^k - \alpha_k f'(x^k))_{\text{пр}}, \quad k = 0, 1, 2, \dots \quad (5.31)$$

Сходимость последовательности (5.31) к решению задачи (5.29) вытекает из следующей теоремы.

Теорема. Пусть множество D выпукло и замкнуто, функция $f(x(\theta))$ строго выпукла на D при $\theta > 0$ и дифференцируема на D , причем ее градиент удовлетворяет условию Липшица:

$$\|f'(x) - f'(x^1)\| \leq M \|x - x^1\|, \quad x, x^1 \in D.$$

Тогда последовательность, генерируемая по правилу (5.31), где x^0 — произвольная точка из D , а $\alpha_k \equiv \alpha \in (0, 4\theta/M^2)$, сходится к решению x^* задачи (5.29) со скоростью геометрической прогрессии:

$$\|x^{k+1} - x^*\| \leq q \|x^k - x^*\|,$$

где $q = \sqrt{1 - 4\theta\alpha + \alpha^2 M^2} \in (0, 1)$.

Доказательство этой теоремы не приводится.

Конкретных рекомендаций для выбора константы θ не существует, поэтому значения α выбирают опытным путем. В общем случае на каждой итерации необходимо решать задачу (5.30). Однако для некоторых видов множеств D удается получить явную формулу для проекций.

1) Если D — шар: $D = \{x \in \mathbb{R}^n \mid \|x - x^0\|^2 \leq r\}$, то

$$a_{\text{пр}} = x^0 - \frac{a - x^0}{\|a - x^0\|} r.$$

2) Если D — координатный параллелепипед:

$$D = \{x \in \mathbb{R}^n \mid b_j \leq x_j \leq c_j, \quad j = 1, 2, \dots, n\},$$

то

$$a_{j\text{пр}} = \begin{cases} b_j, & \text{если } a_j < b_j, \\ a_j, & \text{если } b_j \leq a_j \leq c_j, \\ c_j, & \text{если } a_j > c_j. \end{cases}$$

3) Если D — неотрицательный ортант:

$$D = \{x \in \mathbb{R}^n \mid x_j \geq 0, \quad j = 1, 2, \dots, n\},$$

то

$$a_{\text{пр}} = (\max\{0, a_1\}; \dots; \max\{0, a_n\}).$$

4) Если D — гиперплоскость: $D = \{x \in \mathbb{R}^n \mid (p, x) = \beta\}$, $p \neq 0$, то

$$a_{\text{пр}} = a + (\beta - (p, a)) \frac{p}{\|p\|^2}.$$

5) Если D — полупространство: $D = \{x \in \mathbb{R}^n \mid (p, x) \geq \beta\}$, $p \neq 0$, то

$$a_{\text{пр}} = a + \max\{0; \beta - (p, a)\} \frac{p}{\|p\|^2}.$$

6) Если D — аффинное множество: $D = \{x \in \mathbb{R}^n \mid Ax = b\}$, строки матрицы A линейно независимы, то

$$a_{\text{пр}} = a - A^T(AA^T)^{-1}(Aa - b).$$

Если множество D задается сложной системой равенств и неравенств, то метод проекции градиента практически не применим, в этом случае задача (5.30) не проще исходной задачи.

Пример. Решить задачу математического программирования:

$$f(x) = x_1^2 - 2x_1 - x_2 \rightarrow \min$$

при $x_1^2 + x_2^2 \leq 4$.

Здесь множество D — круг радиуса $r = 2$ с центром в начале координат $x_0 = (0, 0)$. Последовательность точек в методе проекции градиента задается формулой

$$a_{\text{пр}} = x^0 - \frac{a - x^0}{\|a - x^0\|} r, \quad (5.32)$$

где $x^0 = (0, 0)^T$ — нулевое приближение. Пусть $\alpha_k \equiv \alpha = 0, 1$. Вычислим производные:

$$\frac{\partial f(x)}{\partial x_1} = 2x_1 - 2, \quad \frac{\partial f(x)}{\partial x_2} = -1, \quad \frac{\partial f(0, 0)}{\partial x_1} = -2, \quad \frac{\partial f(0, 0)}{\partial x_2} = -1;$$

тогда согласно соотношению (5.32) определим точку $a^1 = (0, 2; 0, 1)$, затем длину вектора $\|a^1 - x^0\| = 0, 22$, согласно (5.32) — проекцию $a_{\text{пр}}^1 = (2, 1)$ и вычислим значение целевой функции $f(2, 1) = -1$.

На следующем шаге находим: $x^1 = (2, 1)$, $a^2 = (1, 8; 1, 1)$, $a_{\text{пр}}^2 = (1, 8; 1, 1)$, $f(1, 8; 1, 1) = -1, 46$, т. е. итерационный процесс идет в нужном направлении: оптимальное решение имеет координаты $(0, 75; 1, 85)$ и значение целевой функции равно $-2, 8$.

§ 5.12. Многокритериальные задачи линейного программирования

Реальные проблемы, решаемые с помощью методов математического программирования, во многих случаях выдвигают несколько критериев, которым одновременно должно удовлетворять решение задачи. Причем эти критерии часто бывают антагонистическими.

За последние десятилетия разработано много методов анализа и решения подобных задач. Эти методы являются диалоговыми (или интерактивными), состоящими в последовательном анализе возможных решений и выборе наиболее предпочтительного решения. Методы могут основываться на предварительном выделении множества неулучшаемых (эффективных) решений и представлении ЛПР этого множества для принятия решений.

Многокритериальная задача линейного программирования (МКЛП) имеет вид

$$\begin{aligned} \max \{c_1^T x = z_1\}, \\ \max \{c_2^T x = z_2\}, \\ \dots\dots\dots \\ \max \{c_k^T x = z_k\} \end{aligned}$$

при $x \in D$, или в матричной форме:

$$\max \{Cx = z \mid x \in D\}.$$

Здесь k — число целевых функций (критериев); c_i — градиент (вектор коэффициентов) i -й целевой функции (критерия); z_i — значение i -го критерия (целевой функции); D — множество допустимых значений переменных; \max означает, что необходимо находить максимум всех целевых функций одновременно; C — матрица критериев размерностью $k \times n$ (матрица коэффициентов целевой функции, ее строки c_i являются градиентами критериев); z — вектор критериев.

Предположим, что ЛПР всегда имеет функцию *полезности* $u: \mathbb{R}^k \rightarrow \mathbb{R}^1$. Эта функция отображает векторы критериев на действительную прямую так, что большее значение на этой прямой соответствует более предпочтительному вектору критериев. Функции полезности бывают разных видов: неубывающие, возрастающие, вогнутые, строго вогнутые, убывающие, выпуклые, строго выпуклые, монотонные и т. д. Однако исходные функции полезности могут быть преобразованы таким образом, чтобы они приняли вид, необходимый ЛПР, например чтобы функция полезности стала покоординатно возрастающей.

Многокритериальные задачи графически исследуются и в пространстве решений (как в задачах линейного программирования), и в пространстве критериев.

Пусть D — допустимая область в пространстве решений, а Z — допустимое множество в пространстве критериев. Последнее представляет собой множество образов всех точек из D :

$$Z = \{z \in \mathbb{R}^k \mid z = Cx, x \in D\}.$$

Если известна функция полезности u , то многокритериальная задача преобразуется в однокритериальную задачу

$$\max \{u(z) \mid z = Cx, x \in D\}.$$

Этот подход применим только концептуально.

Пример. Рассмотрим задачу МКЛП

$$\max \{x_1 + x_2 = z_1\}, \quad \max \{x_1 = z_2\}$$

при условиях

$$4x_1 + 3x_2 \leq 12, \quad x_1, x_2 \geq 0,$$

где функция полезности имеет вид $u = 2z_1z_2$.

1. В пространстве решений имеем $u = 2x_1^2 + 2x_1x_2$. Из рис. 5.17 видно, что оптимальной точкой является точка $(3, 0)$, оптимальный вектор имеет вид $(3, 3)^T$, оптимальным значением является $u = 18$.

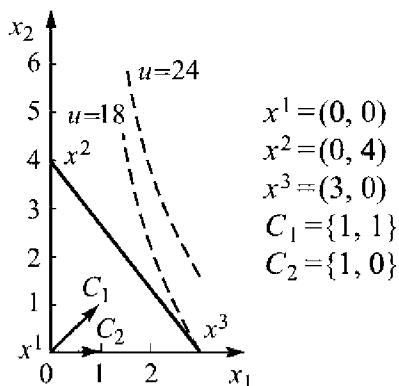


Рис. 5.17. Пространство решений

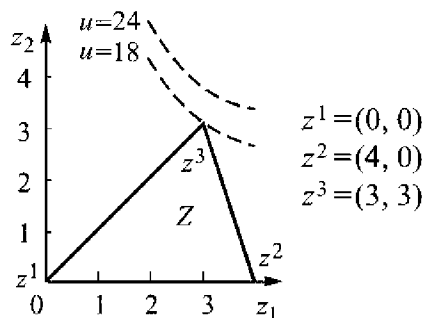


Рис. 5.18. Пространство критериев

2. В пространстве критериев имеем $u = 2z_1z_2$ (рис. 5.18).

Точка z^1 — образ x^1 , т. е. $z^1 = (0, 0)$, точка z^2 — образ x^2 , т. е. $z^2 = (4, 0)$, точка z^3 — оптимальный вектор критериев, x^3 — прообраз z^3 — оптимальная точка; оптимальным значением является $u = 18$.

Для критериальных векторов z вводится понятие *доминирования*. Пусть $z^1, z^2 \in \mathbb{R}^k$ — критериальные векторы.

Вектор z^1 *доминирует* вектор z^2 тогда и только тогда, когда $z^1 \geq z^2$ и $z^1 \neq z^2$ (т. е. $z_i^1 \geq z_i^2$ для всех i и $z_i^1 > z_i^2$ по крайней мере для одного i).

Таким образом, никакой компонент вектора z^1 не меньше соответствующего компонента вектора z^2 , и, одновременно, по крайней мере один компонент вектора z^1 больше соответствующего компонента вектора z^2 .

Вектор z^1 *сильно доминирует* вектор z^2 тогда и только тогда, когда $z^1 > z^2$ (т. е. $z_i^1 > z_i^2$ для всех i).

Критериальный вектор считается *недоминируемым*, если он не доминируется ни одним из допустимых критериальных векторов. Вектор $\bar{z} \in Z$ является недоминируемым тогда и только тогда, когда не существует другого вектора $z \in Z$ такого, что $z \geq \bar{z}$ и $z \neq \bar{z}$, иначе \bar{z} является доминируемым критерием.

Критериальный вектор не может быть оптимальным, если он не является недоминируемым. Для каждого недоминируемого критериального вектора существует *покоординатно возрастающая функция полезности*, для которой он является оптимальным.

Понятие доминированности определено только в пространстве критериев; в пространстве решений вводится понятие *эффективности*.

Точка $\bar{x} \in D$ *эффективна* тогда и только тогда, когда не существует другой точки $x \in D$ такой, что $Cx \geq C\bar{x}$ и $Cx \neq C\bar{x}$. В противном случае точка \bar{x} *неэффективна*. Точка $\bar{x} \in D$ эффективна, если ее критериальный вектор недоминируем критериальными векторами других точек из D . Из эффективной точки невозможно сдвинуться допустимым образом так, чтобы увеличить один из критериев, не уменьшив по крайней мере один из остальных.

Вместе с термином «эффективность» используют термины «неулучшаемость» или «оптимальность по Парето». На практике часто используют области Парето. В. Парето сформулировал проблему многокритериальной оптимизации в 1896 г., Л. Заде вернулся к ней в 1963 г. при проектировании систем управления. *Область Парето* задается в пространстве критериев, и любое принадлежащее этой области решение нельзя улучшить одновременно по всем скалярным критериям. Вне области Парето можно улучшать решение по всем критериям одновременно, но ни по одному

критерию оптимальное значение целевой функции здесь не достигается. Таким образом, необходимым условием решения задачи многокритериальной оптимизации является принадлежность решения области Парето, т. е. решение должно быть *оптимальным по Парето*, поскольку остальные решения заведомо не удовлетворяют сразу всем скалярным критериям. Однако, чтобы выбрать единственное решение на множестве оптимальных по Парето решений, необходима дополнительная информация.

Для примера рассмотрим двухкритериальную задачу. На плоскости критериев, где задано множество допустимых решений, первый критерий достигает своего оптимума в точке A , второй — в точке B . Кривая AB , принадлежащая области допустимых решений, определяет для данной задачи область Парето.

Множество всех эффективных точек называется *эффективным множеством*. Обычно легче показать, что точка неэффективна, чем доказать ее эффективность. Чтобы доказать неэффективность точки $x_1 \in D$ достаточно найти другую точку $x_2 \in D$, критериальный вектор которой доминирует критериальный вектор точки x_1 .

В задаче линейного программирования эффективная точка — это некоторая оптимальная точка, так как не существует точка $x \in D$ такая, что $cx > c\bar{x}$.

В задаче ЛП с одним критерием эффективное множество E и оптимальное множество Θ совпадают и оба выпуклы. В многокритериальной задаче множества E и Θ обладают разными свойствами:

1) оптимальное множество Θ может быть ограниченным и несвязным (наличие более чем одной оптимальной точки не означает, что число их обязательно бесконечно);

2) в задаче МКЛП может быть более одного оптимального критериального вектора (в задачах ЛП существует единственное оптимальное значение критерия независимо от числа оптимальных точек);

3) эффективное множество E может оказаться невыпуклым;

4) оптимальная точка не обязательно является граничной (многое зависит от вида функции полезности);

5) при сведении задачи МКЛП к однокритериальной задаче с помощью функции полезности могут существовать локальные оптимумы, не являющиеся глобальными;

6) оптимальное множество Θ может быть пусто в том случае, когда не пусто эффективное множество E .

Для решения задач МКЛП применяются различные методы сведения их к одному скалярному критерию: метод взвешенных сумм, методы сжатия допустимой области, метод ϵ -ограничений, алгоритмы векторной максимизации, различные интерактивные процедуры и др. Процедура сведения задачи к одному скалярному критерию с помощью функции полезности уже разобрана. Рассмотрим другой метод сведения к одному скалярному критерию — метод взвешенных сумм с точечным оцениванием весов.

§ 5.13. Метод взвешенных сумм с точечным оцениванием весов

Метод взвешенных сумм с точным оцениванием весов заключается в следующем. Каждый критерий $c_i^T x$ умножается на положительный весовой коэффициент λ_i , все k взвешенных критериев суммируются и образуют составную целевую функцию $\lambda^T Cx$ (целевую функцию, состоящую из взвешенной суммы). Предположим, что все *весовые векторы* $\lambda \in \mathbb{R}^k$ нормированы так, что сумма их координат λ_i , $i = 1, 2, \dots, k$, равна 1 (в норме пространства L_1). Множество таких весовых векторов имеет вид

$$\Lambda = \left\{ \lambda \in \mathbb{R}^k \mid \lambda_i > 0, \sum_{i=1}^k \lambda_i = 1 \right\}.$$

При известных весовых векторах $\lambda \in \Lambda$ получаем однокритериальную задачу ЛП

$$\max \{ \lambda^T Cx \mid x \in D \},$$

которая будет иметь оптимальное или достаточно близкое к нему решение не известной нам функции полезности. Основная трудность заключается в нахождении подходящего весового вектора. В ряде случаев весовые коэффициенты выбирают пропорционально важности критериев или применяют метод взвешенных сумм, если считать его способом ранжирования точек из допустимого множества в соответствии с их коэффициентом качества, под которым понимают значение составной критериальной функции.

Теорема 1. Точка $\bar{x} \in D$, в которой существует максимум взвешенных сумм задачи ЛП $\max \{ \lambda^T Cx \mid x \in D \}$, где $\lambda \in \Lambda$, является эффективной точкой.

Теорема 2. Если $\bar{x} \in D$ — эффективная точка, то существует вектор $\lambda \in \Lambda$, при котором \bar{x} является решением задачи $\max\{\lambda^T Cx \mid x \in D\}$.

В силу этих теорем все точки, максимизирующие при $\lambda > 0$ взвешенные суммы в задаче ЛП, являются эффективными.

Если взвешенная сумма $\lambda^T Cx$ в задаче ЛП оказалась неограниченной для некоторого весового вектора, то этому весовому вектору нельзя поставить в соответствие ни одной эффективной точки, но могут существовать другие положительные весовые векторы, для которых взвешенные суммы будут ограничены.

Если один или более весовых коэффициентов равны нулю, то нельзя гарантировать, что все точки, максимизирующие взвешенную сумму в задаче ЛП, являются эффективными. Следует помнить, что стандартные пакеты программ ЛП могут не выявлять все граничные точки, максимизирующие целевую функцию, а выводят лишь первую подходящую точку, даже если эта точка не является эффективной.

Один из способов задать весовые коэффициенты — назначить разным критериям их так, чтобы градиент взвешенной суммы $\lambda^T Cx$ совпадал по направлению с градиентом функции полезности $u = u(c_1^T x, c_2^T x, \dots, c_k^T x)$. Пусть функция полезности u дифференцируема. Градиент сложной функции u в точке \bar{x} имеет вид

$$\nabla_x u = \sum_{i=1}^k \frac{\partial u}{\partial z_i} \nabla_x z_i,$$

где частные производные $\partial u / \partial z_i$ вычисляются в точке \bar{x} , $\nabla_x z_i$ — градиент i -го критерия:

$$\nabla_x z_i = c_i.$$

Пусть $\partial u / \partial z_i > 0$. Введем положительные скаляры

$$w_i = \frac{\partial u / \partial z_i}{\partial u / \partial z_1},$$

вычисленные в точке \bar{x} . Тогда направление градиента $\nabla_x u$ в точке \bar{x} можно задать в виде

$$\nabla_x u = \sum_{i=1}^k w_i c_i.$$

Нормируя скаляры w_i , получаем

$$\nabla_x u = \sum_{i=1}^k \lambda_i c_i, \quad \text{где} \quad \lambda_i = \frac{w_i}{\sum_{i=1}^k w_i}.$$

Функция полезности u в общем случае нелинейна, и ее градиент будет изменяться от точки к точке. Рассмотрим касательную гиперплоскость к поверхности уровня функции u в точке \bar{x} :

$$\frac{\partial u}{\partial z_1}(\bar{z}_1 - c_1^T x) + \frac{\partial u}{\partial z_2}(\bar{z}_2 - c_2^T x) + \dots + \frac{\partial u}{\partial z_k}(\bar{z}_k - c_k^T x) = 0,$$

где $\bar{z}_i = c_i^T \bar{x}$, $c_i^T x = z_i$. Разделив это уравнение на $\partial u / \partial z_1 > 0$, получим

$$1 \cdot (\bar{z}_1 - z_1) + w_2(\bar{z}_2 - z_2) + \dots + w_k(\bar{z}_k - z_k) = 0.$$

Для оценки скаляров w_i , $i = 2, 3, \dots, k$, вводится произвольная величина $\Delta_1 > 0$, чтобы компенсировать изменения градиента, и в качестве оценки w_i используют величину Δ_1 / Δ_i , вычисленную в точке \bar{x} . При $\Delta_1 = 1$ получим

$$w_1 = 1, \quad w_2 = \frac{1}{\Delta_2}, \quad \dots, \quad w_k = \frac{1}{\Delta_k}.$$

Нормируя оценки скаляров w_i , получим λ_i . От этого метода оценки весовых коэффициентов не следует ждать большой точности.

В общем случае, когда при некотором $\lambda \in \Lambda_\theta$ существует точка, являющаяся решением составной задачи ЛП $\max\{\lambda^T Cx \mid x \in D\}$, значения оптимальных весовых векторов Λ_θ зависят не только от предпочтений ЛПР, но и от соотношения длин векторов-градиентов целевых функций и геометрии допустимой области, зависят также от степени корреляции критериев (от величины угла между градиентами целевых функций: чем меньше этот угол, тем больше корреляция между критериями). При сильной корреляции двух критериев, задав большой весовой коэффициент одному критерию, получим, что нет необходимости вводить какой-либо весовой коэффициент для другого критерия. При увеличении размерности задачи возрастает сложность оценки оптимальных весовых векторов.

Для облегчения процедуры нахождения оптимальных весовых векторов и решения задач МКЛП проводится масштабирование целевых функций путем применения множителей, выравнивающих диапазоны изменения критериев, а также путем выбора наиболее подходящего определения нормы.

§ 5.14. Сжатие множества допустимых решений

Рассмотрим *метод ϵ -ограничений*. В этом методе для нахождения оптимального решения задачи МКЛП выбирается только один из критериев, а остальные критерии ограничиваются снизу некоторыми числами e_j , т. е. переводятся в условия-ограничения, тем самым происходит сжатие области допустимых решений. Таким образом, вместо задачи МКЛП решается задача ЛП

$$\max \{c_i^T x = z_i\}$$

при

$$c_j x \geq e_j, \quad j \neq i, \quad j = 1, 2, \dots, k, \quad x \in D.$$

Какие из критериев необходимо перевести в ограничения и какие задать значения правых частей e_j , определяет пользователь. Например, неверный выбор e_j может привести к несовместной задаче ЛП. Решение принимают на основе анализа серии подобным образом построенных задач ЛП, т. е. процедура решения задачи МКЛП носит интерактивный характер.

Анализ почти оптимальности — другой способ сжатия области допустимых решений. Рассмотрим алгоритм анализа эффективной точки.

1. Решим задачу ЛП со взвешенными суммами

$$\max \{\lambda^T Cx = z\}, \quad x \in D,$$

и определим величину z^* .

2. Выбрав некоторое $z_1 < z^*$, решим задачу ЛП для сжатой области допустимых решений.

3. Вычислим все критериальные векторы $z_i \in \mathbb{R}^n$, соответствующие граничным точкам x_i сжатой области.

4. Выберем точку x_i , соответствующую самому предпочтительному вектору, в качестве окончательного решения МКЛП.

Трудности здесь могут возникнуть при нахождении всех альтернативных оптимумов для вычисления всех граничных точек.

Пусть D — множество допустимых решений, E — эффективное множество. В общем случае задачи МКЛП приводят к следующим взаимоисключающим и исчерпывающим ситуациям:

- 1) $D = \emptyset$, т. е. система ограничений несовместна;
- 2) $D \neq \emptyset$, $E = \emptyset$ и значение по крайней мере одного критерия ограничено;
- 3) $D = \emptyset$, $E = \emptyset$ и значение всех критериев неограничены;
- 4) $D \neq \emptyset$, E содержит только одну точку;
- 5) $D \neq \emptyset$, E ограничено и содержит бесконечное число точек;
- 6) $D \neq \emptyset$, E неограничено и значение по крайней мере одного критерия неограничено;
- 7) $D \neq \emptyset$, E неограничено и значения всех критериев неограничены.

В любом случае необходимо найти исходную эффективную крайнюю точку. Для этого можно использовать следующие методы:

- 1) взвешенных сумм;
- 2) взвешенных сумм с использованием подзадачи-теста;
- 3) лексикографической максимизации;
- 4) лексикографической максимизации с использованием подзадачи-теста;
- 5) метод Эккера — Куоды;
- 6) метод Бенсона.

В *методе взвешенных сумм* выбирается некоторый вектор

$$\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^k \mid \lambda_i > 0, \sum_{i=1}^k \lambda_i = 1 \right\}$$

и решается задача $\max \{ \lambda^T Cx \mid x \in D \}$.

Если область D ограничена и $D \neq \emptyset$, то эффективная точка будет найдена. Если область D не ограничена, метод не гарантирует нахождения эффективной точки.

При использовании *подзадачи-теста* еще до нахождения максимума целевой функции проверяют граничные точки в процессе работы метода, чем экономят время. При этом уменьшается вероятность неудачной реализации метода взвешенных сумм в случае неограниченной области D .

В методе лексикографической максимизации применяется следующая процедура сжатия допустимых областей:

$$D_0 = D, \quad D_1 = \{y \in D \mid c_{j_1}^T y = \max \{c_{j_1}^T x \mid x \in D_0\}\}, \dots$$

$$\dots, D_k = \{y \in D \mid c_{j_k}^T y = \max \{c_{j_k}^T x \mid x \in D_{k-1}\}\},$$

где D_h — область, полученная сжатием допустимой области после h максимизаций.

Процесс начинаем с максимизации целевой функции $c_{j_1}^T x$ в области D_0 . Затем, ограничиваясь выбором точек из D_0 , максимизируем j_1 -й критерий и получаем область D_1 . В области D_1 максимизируем второй критерий $c_{j_2}^T x$, ограничиваясь рассмотрением точек из D_1 , максимизирующих j_2 -й критерий; получаем область D_2 . Процесс продолжается до тех пор, пока не получим либо $D_k \neq \emptyset$, либо, начиная с некоторого $l, 1 \leq l \leq k, D_l, D_{l+1}, \dots, D_k = \emptyset$. Последнее будет иметь место, если все нерассмотренные целевые функции не ограничены на D_{l-1} .

При использовании подзадачи-теста быстрее находится эффективная точка, в том числе при неограниченности на D всех критериев. Подзадачу можно применять в начальной граничной точке и повторять проверку после каждого шага максимизации или проверять каждую граничную точку. При использовании этого метода задача не решается, если все критерии не ограничены на D и $E \neq \emptyset$.

Методом Эккера — Куоды решается вспомогательная задача

$$\max \{e^T s\}$$

при условиях

$$Cx = Is + Cx_0, \quad Ax = b, \quad 0 \leq x \in \mathbb{R}^n, \quad 0 \leq s \in \mathbb{R}^k.$$

Если (x^*, s^*) — оптимальное решение этой задачи, то $x^* \in E$, и если целевая функция $e^T s$ не ограничена сверху, то $E = \emptyset$. Здесь нет гарантии того, что обнаруженная эффективная точка будет точкой области D .

В методе Бенсона предлагается следующая процедура для определения исходной эффективной крайней точки.

1. Пусть $D \neq \emptyset$. Найти любую точку $x_0 \in D$.

2. Решить задачу

$$\min \{-z^T Cx_0 + u^T b\}$$

при условиях

$$z^T C - u^T A + w^T = -e^T C, \quad w, z \geq 0.$$

Если оптимальное решение этой задачи не существует, то задача МКЛП не имеет эффективных точек. Если существует оптимальное решение (z_0^*, u_0^*, w_0^*) , то переходим к шагу 3.

3. Положить $\bar{\lambda} = (z_0 + e)$ и решить задачу $\max \{\bar{\lambda}^T Cx \mid x \in D\}$. В результате найдем начальную эффективную крайнюю точку.

§ 5.15. Минимальные значения критериев на множестве эффективных точек

Для того чтобы определить минимальное значение i -го критерия на эффективном множестве E , необходимо решить следующую задачу:

$$\min \{c_i^T x = z_i \mid x \in E\}.$$

Поскольку эффективное множество E неизвестно в явном виде, то нельзя непосредственно решить эту задачу. Для большинства задач МКЛП множество E не является выпуклым.

Один из способов решения поставленной задачи базируется на использовании *таблицы выигрышей* (табл. 5.12). Строки таблицы выигрышей представляют собой критериальные векторы, полученные в результате максимизации отдельных критериев. В том случае, когда оптимум не единственный, необходимо принять специальные меры, чтобы все критериальные векторы, стоящие в строках, были недоминируемыми. Величины z_i^* , стоящие на главной диагонали, образуют вектор максимальных значений критериев на множестве эффективных точек. Минимальное значение в j -м столбце таблицы выигрышей — это некоторая оценка минимального значения j -го критерия на множестве E . Если минимальное по столбцу значение находится в строке, в которой стоит *доминируемый* критериальный вектор, то оно может оказаться меньше искомого минимума на множестве E . Если строка, содержащая минимальное значение, является *недоминируемым* критериальным вектором, то минимальное значение либо определяет минимальное значение критерия на множестве E , либо является его оценкой сверху. В целом, этот подход не всегда приводит к оптимальному решению.

Таблица 5.12

Таблица выигрышей

Критерий	z_1	z_2	...	z_k
z_1	z_1^*	z_{12}	...	z_{1k}
z_2	z_{21}	z_2^*	...	z_{2k}
...
z_k	z_{k1}	z_{k2}	...	z_k^*

Для определения минимального на множестве эффективных точек значения i -го критерия можно использовать симплекс-метод. Поочередно лексикографически максимизируя каждый из критериев, строим таблицу выигрышей. Пусть z_{mi} — минимальное значение критерия, находящееся в i -м столбце таблицы выигрышей. Добавим в условия-ограничения задачи еще одно ограничение $c_i^T x \leq z_{mi}$. Начнем счет с граничной точки, соответствующей z_{mi} . Исследуем грань $c_i^T x = z_{mi}$ нового (после дополнительного ограничения) множества допустимых точек с целью найти такую граничную точку, из которой исходит эффективное ребро с убывающим значением i -го критерия. Если такого ребра нет, то текущее значение z_{mi} и есть минимальное значение i -го критерия на E ; процесс окончен. Если такое ребро существует, то производим замещение переменной вдоль этого ребра методом Жордана и переходим в крайнюю точку на другом его конце, где значение i -го критерия равно z_{mi}^n . Вводим дополнительное ограничение $c_i^T x \leq z_{mi}^n$ и повторяем процедуру. Алгоритм заканчивает работу в точке минимума i -го критерия на множестве эффективных точек.

§ 5.16. Параметризация целевой функции

Рассмотрим алгоритм метода взвешенных сумм на примере параметризации целевой функции для задачи ЛП с одним критерием. Аналогичная задача рассматривалась в § 5.2.

Исходная задача имеет вид

$$\max \{c_1^T x = z\}, \quad x \in D.$$

Задан вектор $c_2 \in \mathbb{R}^n$, который определяет изменения координат целевой функции; вводится параметризованный (суммарный) градиент целевой функции $c^+ \in \mathbb{R}^n$:

$$c^+ = c_1 + Pc_2,$$

где $P \in [0, P_{\max}]$. Отсюда находится последовательность параметрически оптимальных граничных точек (и ребер) при изменении P от 0 до P_{\max} .

Точка $x \in D$ называется *параметрически оптимальной*, если она максимизирует величину $(c^+)^T x$, $x \in D$, для некоторого значения $P \in [0, P_{\max}]$.

При использовании метода взвешенных сумм вводится выпуклая комбинация векторов

$$\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^2 \mid \lambda_i \geq 0, \sum_{i=1}^2 \lambda_i = 1 \right\};$$

и тогда $c^+ = \lambda_1 c_1 + \lambda_2 c_2$.

Между рассмотренными подходами существует прямая связь:

$$P = \frac{\lambda_2}{1 - \lambda_2},$$

так как $c_1 + Pc_2 = \frac{1}{\lambda_1}(\lambda_1 c_1 + \lambda_2 c_2)$.

Однако в первом подходе вектор c^+ не достигает вектора c_2 (только стремится к нему), во втором — $c^+ = \lambda_2 c_2$ при $\lambda_1 = 0$.

В поставленной задаче требуется определить критические значения P или λ_1 и λ_2 , при которых новые базисы (крайние точки) становятся параметрически оптимальными (т. е. происходит смена базиса). Задача решается в три этапа.

I. Находим допустимую крайнюю точку из области D для решения симплекс-методом задачи ЛП

$$\max \{c_1^T x = z\}, \quad x \in D.$$

II. Решаем задачу ЛП $\max \{c_1^T x = z\}$ при $x \in D$, в результате получаем исходный параметрически оптимальный базис.

III. Заменяем c_1 (градиент целевой функции z_1) на $c^+ = \lambda_1 c_1 + \lambda_2 c_2$ и получаем остальные параметрически оптимальные базисы (крайние точки), варьируя значения λ_2 от 0 до 1. При этом строка $c_j^+ - z_j^+$ есть сумма по i , $i = 1, 2$, произведения весового коэффициента λ_i на строку $c_{ij} - z_{ij}$.

В процессе решения могут возникнуть следующие ситуации.

1. Все небазисные элементы $c_{2j} - z_{2j}$ не положительны. В этом случае исходная точка оптимальна.

2. Существуют небазисные положительные элементы $c_{2j} - z_{2j}$, т. е. найдется выпуклая комбинация, при которой $c_j^+ - z_j^+ > 0$; небазисную переменную, соответствующую этому элементу, переводим в базис. Берем тот элемент $c_j^+ - z_j^+$, который первым стал больше нуля при увеличении значения λ_2 . Ближайшим большим критическим значением λ_2 будет величина

$$\lambda_2 = \min_{j \in J} \left\{ \frac{-(c_{1j} - z_{1j})}{(c_{2j} - z_{2j}) - (c_{1j} - z_{1j})} \right\},$$

где $J = \{j \mid (c_{2j} - z_{2j}) > 0, (c_{1j} - z_{1j}) \leq 0\}$.

То значение j , при котором дробь минимизируется, указывает небазисную переменную, переводимую в базис. Далее продолжается параметризация по λ_2 .

Задача 1. Для $c_1 = (-3, -1)^T$ и $c_2 = (1, 2)^T$ рассмотрим задачу параметрического ЛП с ограничениями (рис. 5.19):

$$x_2 \leq 3, \quad 3x_1 - x_2 \leq 6, \quad x_1, x_2 \geq 0.$$

Решение. Оптимальную исходную симплекс-таблицу (табл. 5.13) задачи ЛП

$$\max \{c_1^T x \mid x \in D\}$$

дополним строкой

$$c_{2j} - z_{2j}, \quad z_j = c_b^T y_j,$$

где c_b — часть координат вектора c_i , стоящих в базисных столбцах; y_j — элементы j -го столбца матрицы. В рассматриваемом случае $c_{1b} = (0, 0)^T$, $c_{2b} = (0, 0)^T$ — два базисных вектора, s_3 и s_4 — слабые переменные.

Для небазисных переменных x_1 и x_2 имеем $c_{1j} - z_{1j} < 0$, а $c_{2j} - z_{2j} > 0$, т. е. множество J равно $\{1; 2\}$. Вычислим критические значения весовых коэффициентов:

$$\lambda_2 = \min \left\{ -\frac{-3}{1 - (-3)}; \frac{1}{2 + 1} \right\} = \frac{1}{3} \quad \text{при } j = 2; \quad \lambda_1 = \frac{2}{3}.$$

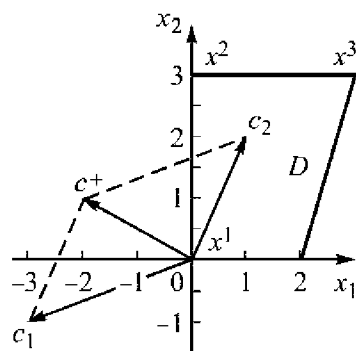


Рис. 5.19. Допустимое множество задачи 1

Переменную x_2 перенесем в базис (в табл. 5.13 обведен генеральный элемент). Вектор

$$c^+ = \frac{2}{3}c_1 + \frac{1}{3}c_2 = \left(-\frac{5}{3}, 0\right)^T$$

ортогонален ребру $\gamma(x^1, x^2)$; строка $c_j^+ - z_j^+$ имеет вид $(-5/3, 0, 0, 0)$.

Таблица 5.13

Оптимальная исходная симплекс-таблица

Базис	Свободный член	x_1	x_2	s_3	s_4
s_3	3	0	<u>1</u>	1	0
s_4	6	3	-1	0	1
c_1		-3	-1	0	0
c_2		1	2	0	0
$c_{1j} - z_{1j}$		-3	-1	0	0
$c_{2j} - z_{2j}$		1	2	0	0

В результате вычислений получим табл. 5.14, в которой $c_{1b} = (-1, 0)^T$, $c_{2b} = (2, 0)^T$. Только для столбца x_1 выполнены условия $c_{1j} - z_{1j} < 0$ и $c_{2j} - z_{2j} > 0$, т. е. $j = 1$. Новыми критическими значениями весовых коэффициентов являются

$$\lambda_2 = \min \left\{ \frac{3}{1+3} \right\} = \frac{3}{4} \quad \text{и} \quad \lambda_1 = \frac{1}{4}.$$

Таблица 5.14

Вторая итерация

Базис	Свободный член	x_1	x_2	s_3	s_4
x_2	3	0	1	1	0
s_4	9	<u>3</u>	0	1	1
$c_{1j} - z_{1j}$		-3	0	1	0
$c_{2j} - z_{2j}$		1	0	-2	0

Переменную x^1 переводим в базис. Вектор

$$c^+ = \frac{1}{4}c_1 + \frac{3}{4}c_2 = \left(0, \frac{5}{4}\right)^T$$

ортогонален ребру $\gamma(x^2, x^3)$; строка $c_j^+ - z_j^+$ имеет вид $(0, 0, -5/4, 0)$. Вводим в базис переменную x_1 . В результате вычислений получаем табл. 5.15.

Таблица 5.15

Последняя симплекс-таблица

Базис	Свободный член	x_1	x_2	s_3	s_4
x_2	3	0	1	1	0
x_1	3	1	0	1/3	1/3
$c_{1j} - z_{1j}$		0	0	2	1
$c_{2j} - z_{2j}$		0	0	-7/3	-1/3

В табл. 5.15 имеем $c_{1j} - z_{1j} > 0$, $c_{2j} - z_{2j} < 0$, т. е. $J = \emptyset$, следовательно, процесс завершен.

Получили следующие подмножества множества Λ , относящиеся к различным параметрически оптимальным крайним точкам и ребрам:

$$\Lambda_{x^1} = \left\{ \lambda \in \Lambda \mid \lambda_1 \in \left[\frac{2}{3}, 1 \right], \lambda_2 \in \left[0, \frac{1}{3} \right] \right\},$$

$$\Lambda_{\gamma(x^1, x^2)} = \left\{ \lambda \in \Lambda \mid \lambda_1 = \frac{2}{3}, \lambda_2 = \frac{1}{3} \right\},$$

$$\Lambda_{x^2} = \left\{ \lambda \in \Lambda \mid \lambda_1 \in \left[\frac{1}{4}, \frac{2}{3} \right], \lambda_2 \in \left[\frac{1}{3}, \frac{3}{4} \right] \right\},$$

$$\Lambda_{\gamma(x^2, x^3)} = \left\{ \lambda \in \Lambda \mid \lambda_1 = \frac{1}{4}, \lambda_2 = \frac{3}{4} \right\},$$

$$\Lambda_{x^3} = \left\{ \lambda \in \Lambda \mid \lambda_1 \in \left[0, \frac{1}{4} \right], \lambda_2 \in \left[\frac{3}{4}, 1 \right] \right\}.$$

Задача 2. С помощью метода взвешенных сумм провести анализ задачи МКЛП:

$$c_1 = (-1, 3)^T, \quad c_2 = (3, 3)^T, \quad c_3 = (1, 2)^T$$

при условиях

$$x_2 \leq 4, \quad x_1 + 2x_2 \leq 10, \quad 2x_1 + x_2 \leq 10, \quad x_1, x_2 \geq 0.$$

Область допустимых значений задачи приведена на рис. 5.20.

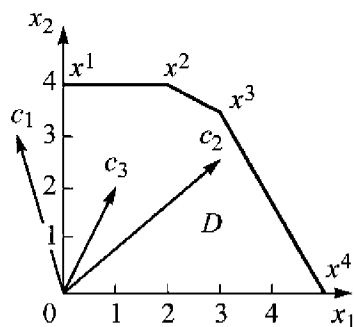


Рис. 5.20. Допустимое множество задачи 2

Решение. Параметризованный (суммарный) градиент целевой функции имеет вид

$$c^+ = \lambda_1 c_1 + \lambda_2 c_2 + \lambda_3 c_3,$$

где

$$\begin{aligned} \lambda &= (\lambda_1, \lambda_2, \lambda_3) \in \Lambda = \\ &= \left\{ \lambda \in \mathbb{R}^3 \mid \lambda_i \geq 0, \sum_{i=1}^3 \lambda_i = 1 \right\}. \end{aligned}$$

Найдем подмножества множества Λ , принадлежащие различным параметрически оптимальным крайним точкам и ребрам. Решим задачу ЛП $\max \{c_1^T x \mid x \in D\}$ и добавим в полученную оптимальную симплекс-таблицу строки для $c_{2j} - z_{2j}$ и $c_{3j} - z_{3j}$ (табл. 5.16). После решения задачи ЛП мы получим начальную параметрически оптимальную точку $x^1 = (0, 4)$ (см. рис. 5.20).

Таблица 5.16

Оптимальное решение задачи ЛП

Базис	Свободный член	x_1	x_2	s_3	s_4	s_5
x_2	4	0	1	1	0	0
s_4	2	<u>1</u>	0	-2	1	0
s_5	6	2	0	1	0	1
$c_{1j} - z_{1j}$		-1	0	2	0	0
$c_{2j} - z_{2j}$		3	0	-3	0	0
$c_{3j} - z_{3j}$		1	0	-2	0	0

Анализируем строку $c_j^+ - z_j^+$, равную сумме по i , $i=1, 2, 3$, произведения весового коэффициента λ_i на строку $c_{ij} - z_{ij}$. Поскольку небазисные элементы строки $c_j^+ - z_j^+$ должны быть неположительны в параметрически оптимальной крайней точке для всех λ из подмножества Λ_{x^1} , соответствующего точке x^1 , то имеем

$$-\lambda + 3\lambda_2 + \lambda_3 \leq 0, \quad -3\lambda_1 - 3\lambda_2 - 2\lambda_3 \leq 0,$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1, \quad \lambda \in \Lambda,$$

или, учитывая, что $\lambda_3 = 1 - \lambda_1 - \lambda_2$, получим

$$2\lambda_1 - 2\lambda_2 \geq 1, \quad -\lambda_1 - \lambda_2 \leq 2,$$

$$\lambda_1 + \lambda_2 \leq 1, \quad \lambda_1, \lambda_2 \geq 0.$$

Отсюда имеем $\lambda^1 = (1/2, 0, 1/2)$, $\lambda^2 = (3/4, 1/4, 0)$. Подмножество Λ_{x^1} изображено на рис. 5.21.

Переводим переменную x_1 в базис и переходим при использовании метода Жордана в точку $x^2 = (2, 4)$ (табл. 5.17).

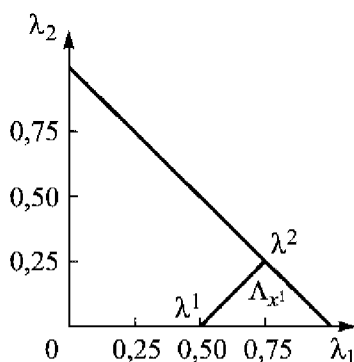


Рис. 5.21. Подмножество Λ_{x^1}

Таблица 5.17

Вторая итерация

Базис	Свободный член	x_1	x_2	s_3	s_4	s_5
x_2	4	0	1	1	0	0
x_1	2	1	0	-2	1	0
s_5	2	2	0	3	-2	1
$c_{1j} - z_{1j}$		0	0	-5	1	0
$c_{2j} - z_{2j}$		0	0	3	-3	0
$c_{3j} - z_{3j}$		0	0	0	-1	0

Для определения подмножества Λ_{x^2} аналогично получим систему

$$-5\lambda_1 + 3\lambda_2 \leq 0,$$

$$\lambda_1 - 3\lambda_2 - \lambda_3 \leq 0,$$

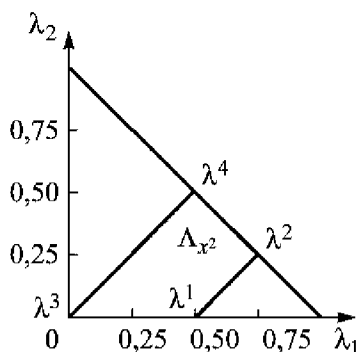
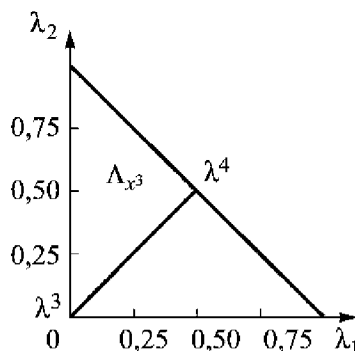
$$\lambda_1 + \lambda_2 + \lambda_3 = 1, \quad \lambda \in \Lambda,$$

или, исключая переменную λ_3 , имеем

$$-5\lambda_1 + 3\lambda_2 \leq 0, \quad 2\lambda_1 - 2\lambda_2 \leq 1,$$

$$\lambda_1 + \lambda_2 \leq 1, \quad \lambda_1, \lambda_2 \geq 0.$$

Отсюда получаем $\lambda^1 = (1/2, 0, 1/2)$, $\lambda^2 = (3/4, 1/4, 0)$, $\lambda^3 = (0, 0, 1)$, $\lambda^4 = (3/8, 5/8, 0)$. Подмножество Λ_{x^2} изображено на рис. 5.22.

Рис. 5.22. Подмножество Λ_{x^2} Рис. 5.23. Подмножество Λ_{x^3}

Далее переходим в точку $x^3 = (10/3, 10/3)$ (табл. 5.18).

Таблица 5.18

Последняя итерация

Базис	Свободный член	x_1	x_2	s_3	s_4	s_5
x_2	10/3	0	1	0	2/3	-1/3
x_1	10/3	1	0	0	-1/3	2/3
s_3	2/3	0	0	1	-2/3	1/3
$c_{1j} - z_{1j}$		0	0	-5	-7/3	5/3
$c_{2j} - z_{2j}$		0	0	3	-1	-1
$c_{3j} - z_{3j}$		0	0	0	-1	0

Из табл. 5.18 имеем систему

$$-\frac{7}{3}\lambda_1 - \lambda_2 - \lambda_3 \leq 0, \quad \frac{5}{3}\lambda_1 - \lambda_2 \leq 0,$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1, \quad \lambda \in \Lambda,$$

или

$$-\frac{4}{3}\lambda_1 \leq 1, \quad \frac{5}{3}\lambda_1 - \lambda_2 \leq 0,$$

$$\lambda_1 + \lambda_2 \leq 1, \quad \lambda_1, \lambda_2 \geq 0.$$

Подмножество Λ_{x^3} имеет вид, показанный на рис. 5.23.

§ 5.17. Целевое программирование

Целевое программирование (ЦП) возникло как приложение обычного линейного программирования. В настоящее время ЦП используют для решения задач многокритериальной оптимизации. В целевом программировании устанавливается некоторый уровень достижения целей по каждому критерию. От обычного линейного программирования ЦП отличается следующим:

- 1) интерпретацией критериев как целей;
- 2) приписыванием приоритетов и/или весовых коэффициентов функциям целей;
- 3) использованием переменных d_i^+ и d_i^- , являющихся мерой отклонения решения от целевых уровней t_i сверху и снизу соответственно;
- 4) минимизацией взвешенных сумм переменных отклонений, чтобы найти решения, наилучшим образом удовлетворяющие целям.

Обычно точка, удовлетворяющая сразу всем целям, не является допустимой. Стараются найти допустимую точку, в которой достигаются все цели «наилучшим» образом.

Для каждой цели (целевой функции) устанавливается значение критерия z_i , который должен быть достигнут (если возможно) согласно обусловленным числами t_i целям.

В задачах ЦП рассматриваются *утопическое множество D в пространстве решений* — множество тех точек из \mathbb{R}^n , в которых одновременно достигаются все цели, и *утопическое множество Z в пространстве критериев* — множество критериальных векторов в \mathbb{R}^k , которые одновременно удовлетворяют всем целям.

В ЦП используются две основные модели решения задач: *архимедова модель* и *модель с приоритетами*. В архимедовой модели точки — кандидаты в решение — генерируют путем определения тех точек из D , критериальные векторы которых являются ближайшими в смысле взвешенной метрики пространства L_1 к утопическому множеству в пространстве критериев. В модели с приоритетами генерируют решения, для которых критериальные векторы оказываются наиболее соответствующими в лексикографическом смысле точками утопического множества в пространстве критериев.

Пример. Рассмотрим задачу ЦП:

$$\text{цель } \{c_1^T x = z_1\}, \quad z_1 \geq t_1,$$

$$\text{цель } \{c_2^T x = z_2\}, \quad z_2 = t_2,$$

$$\text{цель } \{c_3^T x = z_3\}, \quad z_3 \in [t_3^H, t_3^B],$$

при $x \in D$.

Архимедова модель этой задачи записывается следующим образом:

$$\min \{w_1^- d_1^- + w_2^+ d_2^+ + w_2^- d_2^- + w_3^+ d_3^+ + w_3^- d_3^-\}$$

при целевых ограничениях

$$c_1^T x + d_1^- \geq t_1, \quad c_2^T x - d_2^+ + d_2^- = t_2, \quad c_3^T x + d_3^- \geq t_3^H,$$

$$c_3^T x - d_3^+ \leq t_3^B, \quad x \in D, \quad d_1^-, d_2^+, d_2^-, d_3^+, d_3^- \geq 0.$$

Переменные w_1 , w_2 , w_3 в целевой функции — положительные штрафные весовые коэффициенты; каждая цель порождает одно целевое ограничение, кроме случая, когда задан диапазон значений целевой функции и возникают два целевых ограничения. В формулировке задачи используются переменные отклонений d_1^- , d_2^+ , d_2^- , ..., которые соответствуют *нежелательным отклонениям*.

Архимедова целевая функция представляет собой взвешенную сумму переменных нежелательных отклонений. Переменные w_1 , w_2 , w_3 позволяют штрафовать нежелательные отклонения от цели с разной степенью жесткости. Целевые ограничения расширяют область допустимых решений D , переводя ее в пространство большей размерности и создавая таким образом архимедову область допустимых решений для задачи ЦП.

Архимедовы задачи ЦП можно решать, используя обычные методы линейного программирования. Но тогда мы можем получить только крайние точки допустимой области в пространстве решений для архимедовой задачи ЦП (т. е. крайние точки области D после ее усечения целевыми ограничениями). В процессе решения могут быть получены следующие варианты:

- 1) крайние точки области D ;
- 2) точки границы области D , не являющиеся крайними;
- 3) внутренние точки области D .

Если ЛПР предпочитает получить точку, не являющуюся крайней точкой допустимой области архимедовой задачи ЦП, то ее можно получить, используя процедуры изменения целевых показателей t_i .

Рассмотрим задачу ЦП с приоритетами. В приоритетном (лексикографическом) ЦП цели группируются по приоритетам. Цели с высшим уровнем приоритета считаются «бесконечно важными» по сравнению с целями со следующим уровнем приоритета, т. е. если решение задачи получено при некотором уровне приоритета, то цели нижних уровней не учитываются в дальнейшем. Рассмотрим задачу ЦП с приоритетами вида

$$\text{цель } \{c_1^T x = z_1\}, \quad R_1(z_1 \leq t_1),$$

$$\text{цель } \{c_2^T x = z_2\}, \quad R_2(z_2 \geq t_2),$$

$$\text{цель } \{c_3^T x = z_3\}, \quad R_3(z_3 = t_3)$$

при $x \in D$, в которой $j = 1, 2, 3$ указывают цели с уровнем приоритета j . Величины P_j служат и в качестве характеристик приоритетов, причем $P_j \gg P_{j+1}$ (т. е. P_j много больше P_{j+1}).

Запишем задачу ЦП с приоритетами в следующей лексикографической форме:

$$\text{lex min } \{d_1^+, d_2^-, (d_3^+ + d_3^-)\}$$

при условиях

$$c_1^T x - d_1^+ \leq t_1, \quad c_2^T x + d_2^- \geq t_2,$$

$$c_3^T x - d_3^+ + d_3^- = t_3, \quad x \in D, \quad d_1^+, d_2^-, d_3^+, d_3^- \geq 0.$$

Решается данная задача с помощью методов линейного программирования, при этом последовательно рассматриваются задачи с наибольшим приоритетом. На первом этапе решаем задачу целевого программирования с первым приоритетом:

$$\min \{d_1^+\}$$

при условиях

$$c_1^T x - d_1^+ \leq t_1, \quad x \in D \quad d_1^+ \geq 0.$$

Если в этой задаче есть альтернативные оптимумы (для небазисных элементов есть соответствующее значение $c_j - z_j = 0$), то

решаем задачу со вторым приоритетом, учитывая результаты, полученные на первом этапе:

$$\min \{d_2^-\}$$

при условиях

$$c_1^T x \leq t_1 + (d_1^+)_{\text{опт}}, \quad c_2^T x + d_2^- \geq t_2, \quad x \in D, \quad d_2^- \geq 0.$$

Здесь $(d_1^+)_{\text{опт}}$ — оптимальное значение переменной d_1^+ , найденное на первом этапе.

Если в задаче второго этапа есть альтернативные оптимумы, то решаем задачу третьего этапа с третьим приоритетом, учитывая результаты, полученные на первых двух этапах:

$$\min \{d_3^+ + d_3^-\}$$

при условиях

$$c_1^T x \leq t_1 + (d_1^+)_{\text{опт}}, \quad c_2^T x \geq t_2 - (d_2^-)_{\text{опт}},$$

$$c_3^T x - d_3^+ + d_3^- = t_3, \quad x \in D, \quad d_3^+, d_3^- \geq 0,$$

где $(d_2^-)_{\text{опт}}$ — оптимальное значение d_2^- , найденное на втором этапе.

Любое решение задачи третьего этапа определяет лексикографический минимум в задаче ЦП с приоритетами. Однако решение задачи прекращается, как только на каком-то этапе будет получено единственное решение, т. е. цели нижних уровней могут и не повлиять на решение.

Задачу ЦП с приоритетами можно решить в течение одного этапа при использовании лексикографического симплекс-метода.

Пример. Рассмотрим задачу ЦП:

$$\text{цель } \{x_2 = z_1\}, \quad R_1(z_1 \geq 5),$$

$$\text{цель } \{-x_1 - x_2 = z_2\}, \quad R_2(z_2 \geq 4),$$

$$\text{цель } \{x_3 = z_3\}, \quad R_3(z_3 \geq 3)$$

при условиях $x_2 \leq 2$, $x_3 \leq 3$, $x_1, x_2, x_3 \geq 0$.

Данная задача преобразуется к виду

$$\text{lex min } \{d_1^-, d_2^-, d_3^-\}$$

при условиях-ограничениях

$$x_2 + d_1^- \geq 5, \quad -x_1 - x_2 + d_2^- \geq 4, \quad x_3 + d_3^- \geq 3, \quad x_2 \leq 2, \quad x_3 \leq 3;$$

здесь все переменные положительные.

Введем дополнительные (слабые) переменные s_1, \dots, s_5 и заполним симплекс-таблицу (табл. 5.19), из которой удалим столбцы базисных переменных; $c_{ij} - z_{ij}$, $i = 1, 2, 3$, — строки относительных оценок целевой функции для каждого лексикографического уровня P_j , $j = 1, 2, 3$, (последние три строки симплекс-таблицы), в пустых клетках — нули.

Таблица 5.19

Симплекс-таблица первого этапа

Базис	Свободный член	x_1	x_3	s_1	s_2	s_3	s_4
d_1^-	3			-1			-1
d_2^-	6	-1			-1		
d_3^-	3		1			-1	
x_1	2						-1
s_5	3		1				
$c_{1j} - z_{1j}$	3			1			1
$c_{2j} - z_{2j}$	6	1			1		-1
$c_{3j} - z_{3j}$	3		-1			1	

Анализируя строки $c_{ij} - z_{ij}$, $i = 1, 2, 3$, видим, что переменную s_4 можно было бы перевести в базисные переменные. Тогда целевая функция второго лексикографического уровня может быть уменьшена, но при этом увеличится целевая функция первого лексикографического уровня, чего допустить нельзя. Таким образом, точка $(x_1, x_2, x_3) = (0, 2, 0)$ минимизирует целевые функции первого и второго этапов. Поскольку существуют альтернативные оптимумы, переходим к третьему этапу.

Вводим в базис переменную x_3 , так как в первой и второй строках $c_{ij} - z_{ij}$ над элементом -1 нет положительных элементов. Получим новую симплекс-таблицу (табл. 5.20) и оптимальное решение $(x_1, x_2, x_3) = (0, 2, 3)$, которое и является лексикографическим минимумом для рассматриваемой задачи. В точке оптимального решения для первой цели имеем $d_1^- = 3$; для второй цели — $d_2^- = 6$; для третьей цели $d_3^- = 0$, т. е. цель достигнута.

Таблица 5.20

Заключительная симплекс-таблица

Базис	Свободный член	x_1	s_1	s_2	s_3	s_4	s_5
d_1^-	3		-1			-1	
d_2^-	6	-1		-1		1	
d_3^-	0				-1		-1
x_1	2					1	
x_3	3						1
P_1	3		1			1	
P_2	6	1		1		-1	
P_3	0				1		1

Наилучшие результаты в решении задач ЦП получаются в интерактивном режиме, когда решаются одновременно и архимедова задача, и задача с приоритетами.

Полезно бывает использовать прием масштабирования целевых ограничений — записать отклонения от целей в процентах,

т. е. ввести вместо d_i выражение $\frac{t_i}{100} d_i$. Если дополнительно ми-

нимизировать новую переменную α и добавить условие, что отклонения d_i не будут превышать значения α , то такая процедура будет минимизировать максимальное отклонение. В этом случае число дополнительных ограничений равно числу переменных отклонения на рассматриваемом уровне приоритета.

Часть II

Статистические методы принятия решений

Всякая наука есть предвидение.

Герберт Спенсер

АНАЛИЗ МЕТОДОВ ПРИНЯТИЯ РЕШЕНИЙ И ПОСТАНОВКА ЗАДАЧИ УЧЕТА ПОГРЕШНОСТЕЙ ПРИЗНАКОВ

§ 6.1. Основные понятия и определения

Теория принятия решений и распознавания образов в последние годы приобретает все большее значение в различных областях знания (технике, экономике, медицине и др.) и проникает в области управления, исследования операций, радиолокации и т. д. Число статей и книг, посвященных этим вопросам, значительно возросло.

Алгоритмы распознавания образов, рассматриваемые в ч. 2 настоящей книги, по совокупности признаков различаются этапностью принятия решений, степенью и характером учета статистики признаков, помех, сигналов.

Различают алгоритмы *одноэтапного* и *многоэтапного* [14, 63, 65, 66, 80–85] принятия решений. Одноэтапное принятие решений предусматривает обязательное получение оценки принятия i -й гипотезы с приемлемой достоверностью. Многоэтапное принятие решений предусматривает отказ от выдачи решения на первом этапе (первых этапах) до получения дополнительного набора информативных признаков (последовательные алгоритмы Вальда), либо принятие приближенного решения до получения дополнительного набора информативных признаков, либо обобщение предварительных решений, полученных в различные моменты времени или от различных источников.

По степени учета статистических закономерностей различают *лингвистические* и *статистические* алгоритмы. По характеру учета статистических закономерностей из статистических алгоритмов выделяют *параметрические* (байесовские и небайесовские), *непараметрические* и *нейрокомпьютерные* алгоритмы.

Лингвистические алгоритмы [18, 62, 81, 84] не учитывают статистики признаков объектов. Вводимые признаки описывают объекты качественно, часто двоичными цифрами 0, 1. Описание признаков в терминах алгебры логики (языковое, кодовое, син-

таксическое) служит при этом основой распознавания изучаемых явлений.

Байесовские параметрические алгоритмы, в отличие от параметрических небайесовских, учитывают не только статистику помех, флуктуаций сигналов и признаков, но и определенные гипотезы об априорных вероятностях P_i различных элементов алфавита классов [28, 80, 85]. Структуры алгоритмов и работающих по ним устройств обработки сигналов определяются по математическим моделям, описывающим изучаемые явления. Статистика признаков сигналов, негауссова в общем случае, устанавливается путем эксперимента, математического или физического моделирования. Введение этой статистики можно трактовать как обучение распознаванию, адаптацию к конкретным условиям распознавания. Непараметрические алгоритмы синтезируются эвристически без явного принятия предположений о конкретных статистических распределениях [28, 52, 59, 62, 80, 85]. Их можно рассматривать в ряде случаев как эвристическое упрощение параметрических байесовских алгоритмов.

Нейрокомпьютерные алгоритмы отличаются своей заранее заданной универсальной структурой с большим числом неизвестных параметров, уточняемых в процессе адаптации (обучения) [60, 101, 107]. Возрастание вычислительных затрат как издержку универсализации компенсируют ростом производительности вычислительных средств.

К решению задач распознавания объектов (образов) привлекают также ряд математических теорий и методов, развитие которых связано с появлением экспертных систем [35, 74]: теорию нечетких множеств и теорию возможностей [29, 74, 82], теорию игр и другие математические методы [7, 8, 12, 60, 61].

В задачах распознавания образов и принятия решений важную роль играет тот факт, насколько строго учтены неопределенности исходных данных. Использование усредненных величин ведет к смещенным оценкам основных показателей, определяющих решение, и, как следствие, к неверным практическим выводам. Поэтому здесь особое внимание уделено строгому учету погрешностей измерений вектора признаков объекта.

Традиционные статистические методы распознавания образов часто не учитывают погрешности измерений наблюдаемых значений признаков, что приводит к следующему:

1) оценки функций условных плотностей распределения вероятностей признаков образов (если даже вид плотностей известен

априори), которые получают по результатам наблюдений, будут смещенными, а их интервальные оценки — неверными;

2) в процессе идентификации образов, имеющих близкие значения координат векторов признаков, вследствие влияния погрешностей на результаты наблюдений может быть принято неверное решение;

3) в традиционных методах принятия решений функции условных плотностей распределения вероятностей признаков считают в процедурах идентификации образов детерминированными (не учитываются их интервальные оценки), и по этой причине не рассматривают имеющие место зоны неопределенности (нулевые зоны) в принятии решений.

В отличие от традиционных статистических методов распознавания образов в настоящей книге рассматриваются методы, которые позволяют учесть погрешности наблюдаемых значений координат векторов признаков объектов, получить несмещенные точечные и интервальные оценки функций условных плотностей распределения вероятностей признаков, оценить «истинные» координаты вектора признаков, по которому ведется идентификация объектов, и включить эти данные в процедуру принятия решений.

В теории принятия решений используют методы математической статистики, которыми проводят проверку гипотез, при этом вводят в рассмотрение реальные потери от возможной ошибки при принятии гипотезы. Учет различных потерь для разных гипотез приводит к другим выводам, полученным методами теории принятия решений, по сравнению с выводами, сделанными с помощью статистической проверки гипотез. Выбор менее вероятной гипотезы может оказаться более предпочтительным, если потери в случае ошибочности такого выбора окажутся меньше потерь, вызванных ошибочностью выбора более вероятной конкурирующей гипотезы. Типичным примером такого случая является отбраковка изделий, выход из строя которых приводит к дорогостоящим последствиям. В этом случае более выгодно нести материальные потери на изготовление лишних изделий, чем допускать прием аварийно опасного изделия.

Вероятность ошибки, функция потерь, правила решений

В статистической теории решений, чтобы принять решение относительно некоторых гипотез (*состояний природы*) ω_1 и ω_2 по результатам наблюдений признаков x изучаемых явлений, необ-

ходимо каким-то образом получить *апостериорное* (на языке байесовских классификаторов) распределение $P(\omega_j | x)$, $j = 1, 2$. Если в последующем окажется, что при наблюдении x вероятность $P(\omega_1 | x)$ будет больше $P(\omega_2 | x)$, то следует все-таки выбрать решение, что состояние природы есть ω_1 . При этом совершается ошибка e , вероятность которой есть

$$P(e | x) = \begin{cases} P(\omega_1 | x), & \text{если принимается решение } \omega_2, \\ P(\omega_2 | x), & \text{если принимается решение } \omega_1. \end{cases}$$

Пусть l_{ij} — потери вследствие принятия решения ω_i при истинном состоянии природы (истинной гипотезе) ω_j . Ожидаемые потери называются *риском*, решение принимается по значению *условного риска*

$$R(\omega_i | x) = \sum_{j=1}^s l_{ij} P(\omega_j | x), \quad i = 1, 2, \dots, s,$$

где s — число состояний природы (гипотез), x — наблюдаемые значения признаков, $x \in X$.

Если окажется, что $R(\omega_i | x) < R(\omega_j | x)$, то выбирается гипотеза ω_i . Для случая двух состояний природы ω_1 и ω_2 последнее неравенство имеет вид

$$(l_{21} - l_{11})P(\omega_1 | x) > (l_{12} - l_{22})P(\omega_2 | x).$$

Рассмотрим для примера случай двух состояний природы, когда апостериорная вероятность известна и определена как произведение условной плотности распределения вероятностей $p(x | \omega_i)$ и априорной вероятности $P(\omega_i)$ (рис. 6.1).

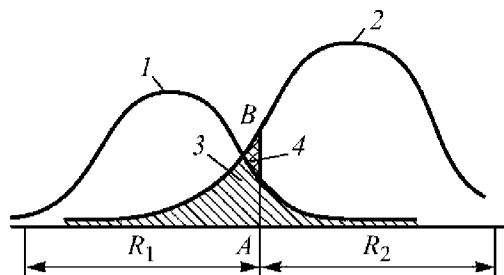


Рис. 6.1. Составляющие вероятности ошибки:

- 1 — апостериорная вероятность состояния природы ω_1 , равная $P(\omega_1)p(x | \omega_1)$;
- 2 — апостериорная вероятность состояния природы ω_2 , равная $P(\omega_2)p(x | \omega_2)$;
- 3 — область, определяющая вероятность ошибки (вся заштрихованная область);
- 4 — область (покрытая клетками), определяющая величину, за счет которой можно уменьшать вероятность ошибки

Возможны два типа ошибок классификации: когда наблюдаемое значение x попадает в область R_2 , в то время как истинным состоянием природы является ω_1 , либо когда x попадает в область R_1 , а истинным состоянием природы является ω_2 . Поскольку эти события несовместные и составляют полное множество событий, то вероятность ошибки e вычислим по формуле

$$P(e) = P(x \in R_2 | \omega_1) + P(x \in R_1 | \omega_2) = \\ = \int_{R_2} p(x | \omega_1) P(\omega_1) dx + \int_{R_1} p(x | \omega_2) P(\omega_2) dx.$$

Если объект относится к классу ω_1 , а его считают объектом класса ω_2 , то совершается *ошибка первого рода*

$$\alpha = \int_{R_2} p(x | \omega_1) P(\omega_1) dx.$$

Наоборот, если объект относится к классу ω_2 , а его считают объектом класса ω_1 , то совершается *ошибка второго рода*

$$\beta = \int_{R_1} p(x | \omega_2) P(\omega_2) dx.$$

Разность $1 - \beta$ называется *мощностью критерия*. Решение следует выбирать так, чтобы мощность критерия была максимальной.

Пусть заданы функции потерь, образующие платежную матрицу

$$\|c\| = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix},$$

где c_{11} и c_{22} — потери, связанные с правильным выбором решения, а c_{12} и c_{21} — потери, связанные с ошибками первого и второго рода соответственно. Тогда средние потери (средний риск) ρ при многократном распознавании неизвестных объектов определяются по формуле

$$\rho = P(\omega_1) \left[c_{11} \int_{R_1} p(x | \omega_1) dx + c_{12} \int_{R_2} p(x | \omega_1) dx \right] + \\ + P(\omega_2) \left[c_{22} \int_{R_2} p(x | \omega_2) dx + c_{21} \int_{R_1} p(x | \omega_2) dx \right].$$

Чтобы найти значение x , при котором средний риск минимален, продифференцируем ρ по x , приравняем производную нулю и получим

$$\frac{p(x|\omega_2)}{p(x|\omega_1)} = \frac{P(\omega_1)(c_{12} - c_{11})}{P(\omega_2)(c_{21} - c_{22})}.$$

Отношение условных плотностей распределения

$$\frac{p(x|\omega_2)}{p(x|\omega_1)} = \lambda(x)$$

называется *коэффициентом правдоподобия*, или *отношением правдоподобия*.

Пусть границей области решений является вертикальная прямая AB (см. рис. 6.1). Ясно, что, смещая границу области решений AB влево, можно уменьшить вероятность ошибки за счет сужения области 4. Именно таким образом уменьшают ошибку распознавания при разработке оптимальных критериев принятия решений.

В теории принятия решений рассматриваются три различных пространства:

1) *пространство наблюдений* X , содержащее все возможные наблюдения $x = (x_1, x_2, \dots, x_n)$;

2) *пространство параметров* Ω , содержащее все возможные значения параметров $\omega = (\omega_1, \omega_2, \dots, \omega_p)$ (возможные значения ω часто называют состояниями природы);

3) *пространство решений* D , которое содержит всевозможные значения решений d .

Правило решения δ (или *процедура решения*, или *решающая функция*) указывает, какое решение d необходимо принять, если получены наблюдения $x = (x_1, x_2, \dots, x_n)$, т. е. $d = \delta(x)$.

Для выбора правила решений вводится *функция потерь* $L(\omega, d)$, значение которой равно величине потери, связанной с выбором решения d , а ω рассматривается как истинное значение параметра. Значения функции потерь $L(\omega, \delta(x))$ являются случайными величинами и зависят от переменной x . Именно учет потерь отличает теорию принятия решений от теории статистической проверки гипотез.

Решение принимают по усредненным величинам. К ним, в частности, относится функция риска.

Функция риска $\rho_\delta(\omega)$ для правила решения δ определяется следующим образом:

$$\rho_\delta(\omega) = M\{L(\omega, \delta(x))\},$$

где $M(k)$ — математическое ожидание случайной величины k , т. е. функция риска $\rho_\delta(\omega)$ определяет среднюю потерю по всем возможным наблюдениям.

Согласно теории Байеса, когда любой параметр ω является случайной величиной с плотностью вероятности $\pi(\omega)$, ожидаемый риск, усредненный по всем значениям ω :

$$\rho_\pi(\delta) = \int_{\Omega} \rho_\delta(\omega) \pi(\omega) d\omega,$$

называется *апостериорным риском* использования правила решения δ при заданной априорной плотности $\pi(\omega)$. Апостериорный риск $\rho_\pi(\delta)$ можно записать в виде

$$\rho_\pi(\delta) = M_\omega\{M_x\{L(\omega, \delta(x)) | \omega\}\} = M_x\{M_\omega\{L(\omega, \delta(x)) | x\}\}.$$

Индекс оператора математического ожидания указывает переменную, по которой производится усреднение. Величина

$$M_\omega\{L(\omega, \delta(x)) | x\}$$

называется *апостериорной потерей* при заданных наблюдениях x . Она соответствует среднему значению потерь, связанному с принятием решения $\delta(x)$. Усреднение проводится по апостериорной плотности $p(\omega | x)$.

Классический подход к выбору правила решения δ основан на использовании функции риска $\rho_\delta(\omega)$. Наилучшим правилом решения является решение δ , которое минимизирует функцию риска для всех ω . Правило решения δ' называется *допустимым*, если не существует правила δ такого, что неравенство $\rho_\delta(\omega) < \rho_{\delta'}(\omega)$ справедливо для всех значений ω .

Байесовский подход к выбору правила решения δ для известной априорной плотности $\pi(\omega)$ основан на использовании функции апостериорного риска $\rho_\pi(\delta)$. Наилучшим правилом решения в таком случае является то правило δ , которое дает наименьший апостериорный риск, т. е. $\rho_\pi(\delta) \leq \rho_\pi(\delta')$ для любого δ' . При довольно общих предположениях можно доказать, что все допустимые решения являются байесовскими решениями, т. е. если δ — допустимое правило решения, то существует некоторая априорная

плотность $\pi(\omega)$ такая, что правило δ является байесовским решением для $\pi(\omega)$. Если задано правило решения, то существует байесовское правило, которое эквивалентно ему или является более предпочтительным.

Класс байесовских правил является *полным* классом. Если δ_B — байесовское правило, то не существует правила, которое лучше, чем δ_B , для всех ω . В то же время δ_B не всегда может быть допустимым (например, если соответствующее априорное распределение $\pi(\omega)$ равно нулю для некоторых ω).

Для выбора правила решения используется *метод минимакса Неймана*, согласно которому необходимо выбирать правило, минимизирующее максимальный риск. Метод минимакса Неймана приводит к наиболее пессимистическому решению.

Пусть D — пространство решений d ; R — пространство доходов r , которые можно получить в результате решения d и исхода эксперимента ω ; Ω — пространство возможных исходов эксперимента ω . Считаем заданной функцию распределения вероятности $P(\omega)$ на пространстве исходов Ω . На множестве R задана *функция полезности* u . Тогда для любой функции распределения вероятности $P_d(r)$, для которой функция u интегрируема, средняя полезность вычисляется по формуле

$$M\{u | P_d\} = \int_R u(r) dP_d(r) = \int_{\Omega} u(\omega, d) dP(\omega).$$

Следует выбрать решение d , максимизирующее $M\{u | P_d\}$. Обычно в задачах решения каждому доходу $r \in R$ принято сопоставлять не полезность, а *ущерб*, имеющий смысл отрицательной полезности: для всех исходов $\omega \in \Omega$ и всех решений $d \in D$ ущерб (потери) равен по величине функции потерь $L(\omega, d)$, т. е. $L(\omega, d) = -u(\omega, d)$. Вещественная функция потерь $L(\omega, d)$ задается на произведении $\Omega \times D$ пространств. При любом $(\omega, d) \in \Omega \times D$ значение $L(\omega, d)$ представляет собой ущерб от принятия решения d в случае исхода ω .

Пусть $P(\omega)$ — заданная функция распределения вероятности параметра ω . При всяком решении $d \in D$ средний ущерб $\rho(P, d)$, называемый *риском*, определяется формулой

$$\rho(P, d) = \int_{\Omega} L(\omega, d) dP(\omega).$$

Выбирается такое решение d , при котором минимизируется функция риска $\rho(P, d)$.

Пусть Ω — параметрическое пространство с параметром, принимающим значения ω . Для всякого распределения $P(\omega)$ параметра ω байесовский риск $\rho^*(P)$ определяется как точная нижняя грань рисков $\rho(P(\omega), d)$ по всем решениям $d \in D$:

$$\rho^*(P) = \inf_{d \in D} \rho(P(\omega), d).$$

Каждое решение d^* , риск которого равен байесовскому риску, называется *байесовским решением* при распределении $P(\omega)$, т. е. $\rho^*(P) = \rho(P(\omega), d^*)$. В ряде случаев байесовское решение может не достигаться.

Во многих задачах удобно использовать *неотрицательные функции потерь*. Оказывается, любую функцию потерь можно заменить неотрицательным ее аналогом. Рассмотрим новую функцию потерь $L_0(\omega, d)$, определяемую по начальной функции потерь следующим образом:

$$L_0(\omega, d) = \alpha L(\omega, d) + \lambda(\omega), \quad \omega \in \Omega, \quad d \in D,$$

где α — числовой коэффициент, $\lambda(\omega)$ — некоторая функция.

Исходной функции $L(\omega, d)$ соответствует риск $\rho(P(\omega), d)$, а функции $L_0(\omega, d)$ — риск $\rho_0(P(\omega), d)$. Тогда для любых значений $d_1 \in D$ и $d_2 \in D$ соотношения $\rho_0(P(\omega), d_1) < \rho_0(P(\omega), d_2)$ и $\rho(P(\omega), d_1) < \rho(P(\omega), d_2)$ равносильны. В частности, решение d^* тогда и только тогда является байесовским при распределении $P(\omega)$ для исходной задачи, когда оно является байесовским решением при $P(\omega)$ для новой задачи с функцией потерь $L_0(\omega, d)$. Выбирая функцию $\lambda(\omega)$ и ее знак, можно получить $L_0(\omega, d) \geq 0$ при всех $\omega \in \Omega$, $d \in D$. Тогда $\inf L(\omega, d) = 0$.

В любой задаче принятия решения байесовский риск $\rho^*(P(\omega))$ является вогнутой функцией от распределения $P(\omega)$ параметра ω , т. е. для любых распределений $P_1(\omega)$ и $P_2(\omega)$ параметра ω и для любого числа α такого, что $0 \leq \alpha \leq 1$, выполнено неравенство

$$\rho^*[\alpha P_1(\omega) + (1 - \alpha)P_2(\omega)] \geq \alpha \rho^*(P_1(\omega)) + (1 - \alpha) \rho^*(P_2(\omega)).$$

В общем случае байесовский риск мало чувствителен к ошибке (приращению) в выборе значения распределения $P(\omega)$ параметра ω ; если функция $\rho^*(P(\omega))$ кусочно линейна, то приращение

$\Delta \rho^*(P(\omega))$ равно нулю, когда приращение $\Delta P(\omega)$ содержится в интервале линейности функции $\rho^*(P(\omega))$.

Критерии выбора стратегии решений

Критерий Байеса — правило, в соответствии с которым стратегия решений выбирается таким образом, чтобы обеспечить минимум среднего риска. Стратегию, основанную на этом правиле, называют *байесовской стратегией*, а минимальный средний риск — *байесовским риском*.

Байесовский подход состоит в вычислении условных апостериорных вероятностей и принятии решений на основе их сравнения.

Если число классов равно m , а значение признака, полученное в опыте, рассматриваемого объекта равно x_0 , то апостериорная вероятность события, состоящего в том, что объект относится к классу ω_i , вычисляется по формуле

$$P(\omega_i | x = x_0) = \frac{P(\omega_i) p(x_0 | \omega_i)}{\sum_{i=1}^m P(\omega_i) p(x_0 | \omega_i)}.$$

Или, в другой трактовке, при $c_{11} = c_{22} = 0$ объект относится к классу ω_1 , если

$$\frac{p(x | \omega_2)}{p(x | \omega_1)} > \frac{c_{12} P(\omega_1)}{c_{21} P(\omega_2)}.$$

Минимаксный критерий используется, если априорные вероятности появления объектов ω_i , $i = 1, 2, \dots, m$, неизвестны. Минимаксная стратегия состоит в том, что решение о принадлежности неизвестного объекта соответствующему классу ω_i принимается на основе байесовской стратегии, соответствующей такому значению $P(\omega_i)$, при котором средний риск минимален. При наличии классов ω_1 и ω_2 средний риск с учетом того, что

$$P(\omega_2) = 1 - P(\omega_1), \quad c_{11} = c_{22} = 0,$$

определяется по формуле

$$\rho = P(\omega_1) c_{12} \int_{R_2} p(x | \omega_1) dx + [1 - P(\omega_1)] c_{21} \int_{R_1} p(x | \omega_2) dx.$$

Приравнявая производную $\frac{\partial \rho}{\partial [P(\omega_1)]}$ нулю, в некоторой точке x_0 получим

$$c_{12} \int_{R_2} p(x | \omega_1) dx = c_{21} \int_{R_1} p(x | \omega_2) dx.$$

Если измеренное значение признака x меньше x_0 , то объект принадлежит классу ω_1 , или если $x > x_0$, то объект принадлежит классу ω_2 . Данный подход приводит к следующему пороговому значению коэффициента правдоподобия:

$$\lambda'_0 = \frac{c_{12}}{c_{21}} \frac{R(\omega_1)}{1 - R(\omega_1)},$$

где $R(\omega_1)$ — точка максимума функции $\rho = \rho(P(\omega_1))$.

Тогда объект принадлежит классу ω_1 , если $\lambda(x) < \lambda'_0$, и объект принадлежит классу ω_2 , если $\lambda(x) > \lambda'_0$.

Критерий Неймана — Пирсона используется, если неизвестны априорные вероятности появления объектов соответствующих классов и платежная матрица $\|c\|$.

Для построения алгоритма классификации задается допустимое значение условной вероятности ошибки первого рода α , затем определяется такая граница между классами, придерживаясь которой, удастся добиться минимума условной вероятности ошибки второго рода β .

Пусть принято, что допустимая условная вероятность ошибки первого рода не должна превышать постоянной величины A , т. е. $\alpha \leq A$. Требуется определить решение x_0 задачи

$$\min_x \beta = \min_x \int_{-\infty}^x p(z | \omega_2) dz$$

при ограничении вида

$$\alpha = \int_x^{+\infty} p(z | \omega_1) dz < A.$$

Очевидно, что решение x_0 удовлетворяет уравнению

$$\int_{x_0}^{+\infty} p(x | \omega_1) dx = A.$$

Для решения задачи строят *рабочую характеристику* — функцию $1 - \beta$, зависящую от α . Если $\alpha = 0$, то $\beta = 1$ и $1 - \beta = 0$; если $\alpha = 1$, то $\beta = 0$ и $1 - \beta = 1$. Вычислим

$$\frac{\partial(1 - \beta)}{\partial \alpha} = \frac{\partial(1 - \beta)/\partial x_0}{\partial \alpha / \partial x_0} = \frac{p(x_0 | \omega_2)}{p(x_0 | \omega_1)} = \lambda_0.$$

Поскольку тангенс угла наклона касательной к рабочей характеристике равен λ_0 , то для определения α и $1 - \beta$ найдем точку на рабочей характеристике, в которой

$$\lambda_0 = \frac{c_{12}P(\omega_1)}{c_{21}P(\omega_2)}.$$

Ордината этой точки определяет условную вероятность правильного решения, а абсцисса — условную вероятность ошибки первого рода. При этом надо учесть, что производная от среднего риска ρ по априорной вероятности $P(\omega_1)$ в точке его максимума равна нулю, т. е.

$$\frac{\partial \rho}{\partial [P(\omega_1)]} = c_{11}(1 - \alpha) + c_{12}\alpha - c_{22}(1 - \beta) - c_{21}\beta = 0.$$

В координатах α и $1 - \beta$ это уравнение прямой линии. При $c_{11} = c_{22}$ уравнение прямой имеет вид

$$1 - \beta = \alpha \frac{c_{11} - c_{12}}{c_{21} - c_{22}} + 1$$

с угловым коэффициентом

$$k = \frac{c_{11} - c_{12}}{c_{21} - c_{22}}.$$

Координаты точки пересечения этой прямой с рабочей характеристикой определяют вероятности α и $1 - \beta$ в условиях применения минимаксного критерия. Тангенс угла наклона касательной в точке пересечения равен λ_0 .

Связь между статистической проверкой гипотез, теорией принятия решений и математическим программированием

На простом примере покажем связь между статистической проверкой гипотез, теорией принятия решений и математическим программированием. Пусть необходимо проверить выполнимость

гипотезы H_0 при конкурирующей гипотезе H_1 относительно значений параметра ω . В математической статистике для проверки гипотез задается уровень значимости α (вероятность события, состоящего в выборе гипотезы H_1 , когда верна гипотеза H_0), определяется значение β (вероятность события, состоящего в выборе гипотезы H_0 , когда верна гипотеза H_1) и находится такое решение, при котором значение β минимально. В теории принятия решений для каждого правила решения δ существует конечная вероятность $\alpha(\delta)$ выбора гипотезы H_1 , когда верна гипотеза H_0 , и вероятность $\beta(\delta)$ выбора гипотезы H_0 , когда верна гипотеза H_1 (табл. 6.1).

Для выбора решения определим функцию потерь, полагая, что она равна нулю для правильного решения (табл. 6.2).

Таблица 6.1

Вероятности $P(H_0 | \omega)$ и $P(H_1 | \omega)$

Гипотеза	Решение	
	H_0	H_1
H_0	$1 - \alpha(\delta)$	$\alpha(\delta)$
H_1	$\beta(\delta)$	$1 - \beta(\delta)$

Таблица 6.2

Функция потерь

Гипотеза	Решение	
	H_0	H_1
H_0	0	l_0
H_1	l_1	0

В рамках байесовского подхода каждой из гипотез H_0 и H_1 зададим априорные вероятности μ и $1 - \mu$ соответственно. Теперь необходимо выбрать такое правило принятия решения δ , которое минимизирует функцию

$$r_\mu(\delta) = \alpha(\delta)l_0\mu + \beta(\delta)l_1(1 - \mu).$$

Таким образом получена задача математического программирования: минимизировать линейную целевую функцию $r_\mu(\delta)$ на области допустимых значений $[\alpha(\delta), \beta(\delta)]$. Можно доказать, что область допустимых значений $[\alpha(\delta), \beta(\delta)]$ выпукла, и тогда минимум функции $r_\mu(\delta)$ определяется координатами точки касания прямой $r = l_0\mu\alpha + l_1(1 - \mu)\beta$ с нижней границей допустимой области $[\alpha(\delta), \beta(\delta)]$.

Оптимальное решение получается, если рассмотреть апостериорную потерю при заданных наблюдениях x и соответственно

условных вероятностях $p(x|H_0)$ и $p(x|H_1)$. Ожидаемая потеря при выборе гипотезы H_0 вычисляется по формуле

$$l_1 P(H_1|x) = l_1(1-\mu)p(x|H_1),$$

ожидаемая потеря при выборе гипотезы H_1 — по формуле

$$l_0 P(H_0|x) = l_0\mu p(x|H_0).$$

Здесь $P(H_0|x)$ и $P(H_1|x)$ — соответственно апостериорные вероятности того, что верны гипотезы H_0 и H_1 при наблюдаемых значениях x .

Таким образом, правило решения с минимальным риском следующее: выбирается гипотеза H_0 , если выполнено неравенство

$$l_1(1-\mu)p(x|H_1) < l_0\mu p(x|H_0),$$

или неравенство

$$\frac{p(x|H_1)}{p(x|H_0)} < \frac{l_0\mu}{l_1(1-\mu)},$$

в противном случае выбирается гипотеза H_1 .

§ 6.2. Статистические задачи решения с наблюдениями

Перед тем как выбрать решение из множества D , наблюдается значение случайной величины или случайного вектора x , которые связаны с параметром ω . Наблюдение случайной величины x помогает принять рациональное решение. Предполагается, что для всех $\omega \in \Omega$ задано условное распределение значений случайной величины x при известном ω . Подобные задачи называются *статистическими задачами принятия решения*. Основными элементами статистической задачи принятия решения являются параметрическое пространство Ω , пространство решений D , функция потерь $L(\omega, d)$ и семейство условных обобщенных вероятностных плотностей (о. в. п.) $f(\cdot|\omega)$, $\omega \in \Omega$, значений случайной величины x , наблюдаемой до принятия решения. Пусть S — выборочное пространство возможных значений наблюдения x . Для принятия решения требуется знать *решающую функцию* δ , заданную для любого возможного значения $x \in S$ решения $d(x) \in D$. Класс всех решающих функций δ обозначим Δ .

Для любой о. в. п. $\xi(\omega)$ параметра ω и любой решающей функции $\delta \in \Delta$ функция риска $\rho(\xi, \delta)$ определяется соотношением

$$\begin{aligned} \rho(\xi, \delta) &= M\{L(\omega, \delta(x))\} = \\ &= \iint_{\Omega S} L(\omega, \delta(x)) f(x|\omega) \xi(\omega) d\mu(x) d\nu(\omega). \end{aligned} \quad (6.1)$$

Предполагается, что при всех $\omega \in \Omega$ функция $L(\omega, \delta(\cdot))$ измерима и интегрируема на множестве S . Через $d\mu(x)$ и $d\nu(x)$ обозначены меры, указывающие на то, что каждый из интегралов может быть как обычным интегралом от о. в. п., так и суммой значений дискретной функции вероятностей. Функция риска определяет здесь средний ущерб (потери).

Для каждого решения $d \in D$ функция риска $\rho(\xi, \delta)$ при о. в. п. $\xi(\omega)$ определяется формулой

$$\rho(\xi, d) = \int_{\Omega} L(\omega, d) \xi(\omega) d\nu(\omega),$$

для каждого значения $\omega \in \Omega$ функция риска $\rho(\omega, \delta)$, соответствующая решающей функции $\delta \in \Delta$, — формулой

$$\rho(\omega, \delta) = \int_S L(\omega, \delta(x)) f(x|\omega) d\mu(x). \quad (6.2)$$

Из соотношений (6.1) и (6.2) имеем

$$\rho(\xi, \delta) = \int_{\Omega} \rho(\omega, \delta) \xi(\omega) d\nu(\omega).$$

Пусть $\delta^* \in \Delta$ — такая решающая функция, что

$$\rho(\xi, \delta^*) = \inf_{\delta \in \Delta} \rho(\xi, \delta) = \rho^*(\xi);$$

тогда δ^* называется *байесовской решающей функцией* при заданной о. в. п. $\xi(\omega)$, а $\rho^*(\xi)$ называется *байесовским риском*.

При заданной о. в. п. $\xi(\omega)$ параметра ω надо найти решающую функцию δ , минимизирующую функцию риска $\rho(\xi, \delta)$ (см. (6.1)). Если функция $L(\omega, d)$ неотрицательна или ограничена, то в равенстве (6.1) можно изменить порядок интегрирования:

$$\rho(\xi, \delta) = \int_{\Omega} \int_S L(\omega, \delta(x)) f(x|\omega) \xi(\omega) d\nu(\omega) d\mu(x).$$

Поэтому решающую функцию δ , минимизирующую риск, при каждом значении $x \in S$ можно определить из условия минимизации внутреннего интеграла, т. е. байесовская решающая функция имеет вид $\delta^*(x) = d^*$, где d^* — решение из D , минимизирующее интеграл

$$\int_{\Omega} L(\omega, d) f(x|\omega) \xi(\omega) d\nu(\omega). \quad (6.3)$$

Вместо того чтобы искать минимизирующее решение d^* для интеграла (6.3), можно найти то же самое значение d^* из условия минимума интеграла

$$\int_{\Omega} L(\omega, d) \left[\frac{f(x|\omega) \xi(\omega)}{f_1(x)} \right] d\nu(\omega),$$

где $f_1(x) = \int_{\Omega} f(x|\omega) \xi(\omega) d\nu(\omega)$.

Поскольку дробь, заключенная в квадратные скобки, является о. в. п. случайной величины ω , то значение интеграла равно условному математическому ожиданию $M\{L(\omega, d) | x\}$. Маргинальное распределение случайной величины ω называется *априорным* распределением, оно задает распределение случайной величины ω до проведения наблюдений над x . Условное распределение случайной величины ω при известном значении x называется *апостериорным* распределением, оно задает распределение ω после наблюдения x .

Пример 1. Пространство Ω содержит только точки 0 и 1. Пространство решений D состоит из чисел d , $d \in [0, 1]$. Функция потерь определена для $\omega \in \Omega$ и $d \in D$ формулой $L(\omega, d) = |\omega - d|$. Заданы значения вероятностей $P(\omega)$ параметра ω :

$$P(0) = \frac{3}{4}, \quad P(1) = \frac{1}{4}.$$

Тогда для любого решения $d \in D$

$$\rho(P, d) = L(0, d)P(0) + L(1, d)P(1) = \frac{3}{4}d + \frac{1}{4}(1-d) = \frac{1}{2}d + \frac{1}{4}.$$

Отсюда получаем

$$\inf_d \rho(P, d) = \rho(P, 0) = \frac{1}{4},$$

т. е. $\rho^*(P) = 1/4$ — единственное байесовское решение.

Если предположить, что пространство решений — полуоткрытый интервал $(0, 1]$, то байесовский риск будет по-прежнему равен $1/4$, но ни одно решение из D не будет байесовским.

Если решение принимается без предварительных наблюдений, то оптимальным является байесовское решение при априорном распределении случайной величины ω . Если же было предварительное наблюдение, то априорное распределение случайной величины ω заменяется на апостериорное. Отсюда видно, что решение $d^*(x_0)$, задаваемое байесовской решающей функцией $\delta^*(x)$ для наблюдаемого значения x_0 , можно найти без вычисления $\delta^*(x)$, и $\delta^*(x)$ при о. в. п. $\xi(\omega)$ можно определить без расчета байесовского риска $\rho^*(\xi)$.

Рассмотрим влияние цены наблюдения на принимаемое решение. Пусть $c(\omega, x)$ — цена наблюдения значения x при заданном ω . Тогда если $\xi(\omega)$ — о. в. п. случайной величины (параметра) ω , то средняя цена наблюдения

$$M\{c(\omega, x)\} = \int_{\Omega} \int_S c(\omega, x) f(x|\omega) \xi(\omega) d\mu(x) dv(\omega)$$

может быть такой, что выигрыш от проведенного наблюдения не окупит стоимости измерения.

Общим риском, зависящим от наблюдения x и принятой решающей функции δ , называется сумма риска $\rho(\xi, \delta)$ и средней цены наблюдения $M\{c(\omega, x)\}$. Выбирается наблюдение x и соответствующая байесовская решающая функция δ , минимизирующая общий риск.

Пример 2. Пусть $\Omega = \{\omega_1, \omega_2\}$, $D = \{d_1, d_2\}$ и функция потерь задается значениями, приведенными в табл. 6.3. Случайная величина x принимает значения 0 и 1 со следующими условными вероятностями:

$$\begin{aligned} P(x=1|\omega_1) &= \frac{3}{4}, & P(x=0|\omega_1) &= \frac{1}{4}, \\ P(x=0|\omega_2) &= \frac{2}{3}, & P(x=1|\omega_2) &= \frac{1}{3}. \end{aligned}$$

Таблица 6.3

Функция потерь

Ω	D	
	d_1	d_2
ω_1	0	5
ω_2	10	0

Априорное распределение параметра ω следующее:

$$P(\omega_1) = p, \quad P(\omega_2) = 1 - p, \quad 0 \leq p \leq 1;$$

здесь априорная вероятность p задана. Построить байесовскую решающую функцию.

Решение. Пусть $\xi(x)$ — апостериорная вероятность события ω_1 ; если x — наблюдаемое значение, то

$$\xi(x) = P(\omega_1 | x).$$

Пусть $x = 1$; тогда

$$\xi(1) = \frac{\frac{3}{4}p}{\frac{3}{4}p + \frac{1}{3}(1-p)}.$$

Для $x = 0$ имеем

$$\xi(0) = \frac{\frac{1}{4}p}{\frac{1}{4}p + \frac{2}{3}(1-p)}.$$

После наблюдения $x = x_1 = 1$ риск от принятия решения d_1 равен $0 \cdot p + 10(1 - \xi(x))$, а от принятия решения d_2 равен $5\xi(x)$. Решение d_2 будет байесовским, если $10(1 - \xi(x)) > 5\xi(x)$ и $\xi(x) < 2/3$; решение d_1 будет байесовским, если $\xi(x) > 2/3$. При $\xi(x) = 2/3$ и d_1 , и d_2 — байесовское решение.

Если наблюдается значение $x = 1$, то для байесовской решающей функции δ^* имеем: $\delta^*(1) = d_2$ при $\xi(1) < 2/3$ или $p < 8/17$ и $\delta^*(1) = d_1$ при $\xi(1) > 2/3$ или $p > 8/17$; при $p = 8/17$ оба решения d_1 и d_2 являются байесовскими. Если наблюдается значение $x = 0$, то $\delta^*(0) = d_2$ при $\xi(0) < 2/3$ или $p < 16/19$ и $\delta^*(0) = d_1$ при $p > 16/19$; при $p = 16/19$ оба решения d_1 и d_2 являются байесовскими.

Вычислим значения байесовского риска $\rho^*(p)$ для произвольной априорной вероятности p . Если $0 \leq p \leq 8/17$, то решение d_2 будет байесовским независимо от наблюдаемых значений x ; для таких p байесовский риск будет $\rho^*(p) = 5p$. При $8/17 < p < 16/19$ имеем $\delta^*(0) = d_2$, $\delta^*(1) = d_1$. Поэтому

$$\begin{aligned} \rho^*(p) &= p\rho(\omega_1, \delta^*) + (1-p)\rho(\omega_2, \delta^*) = p\left(0 \cdot \frac{3}{4} + 5 \cdot \frac{1}{4}\right) + \\ &+ (1-p)\left(10 \cdot \frac{1}{3} + 0 \cdot \frac{2}{3}\right) = \frac{5}{4}p + \frac{10}{3}(1-p) = \frac{10}{3} - \frac{25}{12}p. \end{aligned}$$

Если $16/19 \leq p \leq 1$, то решение d_1 будет байесовским для любого значения x ; $\rho^*(p) = 10(1 - p)$.

Таким образом, мы провели *экстенсивный вид анализа* — построение байесовской решающей функции (рис. 6.2). В данном примере не учитывалась *цена наблюдения*. Если в рассмотренном примере потребуется принять решение до проведения измерения x , то минимальный риск $\rho_0(p)$ (по априорной информации) будет вычисляться по формуле

$$\rho_0(p) = \begin{cases} 5p & \text{при } 0 \leq p \leq 2/3, \\ 10(1 - p) & \text{при } 2/3 < p \leq 1. \end{cases}$$

Функция $\rho_0(p)$ представлена на рис. 6.3, где обозначена сплошными и пунктирными линиями, образующими треугольник; на нее наложена байесовская функция $\rho^*(p)$, построенная на рис. 6.2. Если $p \leq 8/17$ или $p \geq 16/19$, то тот же риск может быть достигнут без наблюдения x . Если же $8/17 < p < 16/19$, то $\rho^*(p) < \rho_0(p)$. За возможность наблюдения x может быть уплачена цена $c < \rho_0(p) - \rho^*(p)$. Эта разность максимальна при $p = 2/3$ и равна $25/18$.

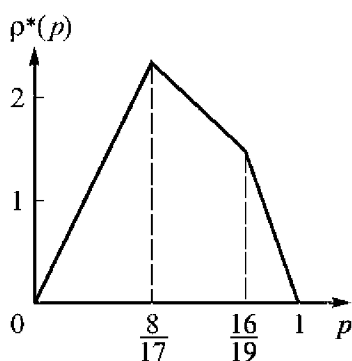


Рис. 6.2. Байесовская решающая функция

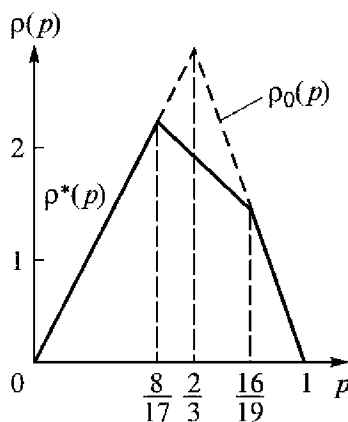


Рис. 6.3. Зависимость минимального риска от p

Если в задаче x — случайный вектор, т. е. набор нескольких случайных величин, то эти величины могут наблюдаться одновременно или в несколько этапов; тогда апостериорное распределение можно вычислять на каждом этапе, взяв в качестве априорного распределения апостериорное распределение, полученное на предыдущем этапе. В итоге получим то же апостериорное рас-

пределение, что и при одновременном учете всех случайных величин.

Например, пусть $f(x, y | \omega)$ — совместная (многомерная) условная о. в. п. случайных величин x и y при $\omega \in \Omega$. Апостериорная о. в. п. $\xi(\omega | x, y)$ параметра ω при наблюдениях x, y имеет вид

$$\xi(\omega | x, y) = \frac{f(x, y | \omega)p(\omega)}{\int_{\Omega} f(x, y | \omega)p(\omega)d\nu(\omega)}. \quad (6.4)$$

Пусть наблюдаем случайную величину x , затем y ; функция $g(x | \omega)$ означает условную о. в. п. x при заданном ω . После наблюдения x апостериорная о. в. п. $\xi(\omega | x)$ для ω имеет вид

$$\xi(\omega | x) = \frac{g(x | \omega)p(\omega)}{\int_{\Omega} g(x | \omega)p(\omega)d\nu(\omega)}. \quad (6.5)$$

Условная о. в. п. $h(y | \omega, x)$ для случайной величины y при заданных ω и x определяется по формуле

$$h(y | \omega, x) = \frac{f(x, y | \omega)}{g(x | \omega)}. \quad (6.6)$$

Апостериорная о. в. п. $\xi(\omega | x, y)$ параметра ω при заданном y запишется в виде

$$\xi(\omega | x, y) = \frac{h(y | \omega, x)\xi(\omega | x)}{\int_{\Omega} h(y | \omega, x)\xi(\omega | x)d\nu(\omega)}. \quad (6.7)$$

Подставляя выражения (6.5) и (6.6) в равенство (6.7), получим соотношение (6.4).

§ 6.3. Статистическая классификация при фиксированном объеме выборки

Пусть результаты измерений x_i , $i = 1, 2, \dots, N$, — случайные величины. Считаем, что для каждого класса образов ω_j , $j = 1, 2, \dots, m$, известны многомерная (N -мерная) функция условной плотности вероятности $p(x | \omega_j)$ (или условного распределения) вектора признаков x и вероятность $P(\omega_j)$ появления образа

ω_j , $j=1, 2, \dots, m$. Классификация образов проводится путем минимизации вероятности ошибочного распознавания с помощью определения решающей функции $\delta(x)$, где равенство $\delta(x) = \delta_j$ означает, что принимается гипотеза $H_j: x \sim \omega_j$. Пусть принятие решения d_j , когда в действительности реальный образ принадлежит ω_i , приводит к потере $L(\omega_i, d_j)$. Величина условных потерь (условный риск) для $x \sim \omega_i$ составит

$$\rho(\omega_i, d) = \int_{X_\Omega} L(\omega_i, d) p(x | \omega_i) dx,$$

где X_Ω — множество значений x , по которым определяется образ ω . Для данного множества априорных вероятностей $P = \{P(\omega_i)\}$ средние потери (средний риск) определяется по формуле

$$\rho(P, d) = \sum_{j=1}^m P(\omega_j) \rho(\omega_j, d),$$

где $\rho(\omega_j, d)$ — условный риск, когда решение d приводит к гипотезе $x \sim \omega_j$, или

$$\rho(P, d) = \int_{X_\Omega} P(x) \rho_x(P, d) dx, \quad (6.8)$$

где $\rho_x(P, d)$ — апостериорный условный средний риск решения d при данных замерах признаков x :

$$\rho_x(P, d) = \frac{\sum_{j=1}^m L(\omega_j, d) P(\omega_j) p(x | \omega_j)}{P(x)}.$$

Необходимо найти такое решение d_j , $j=1, 2, \dots, m$, которое минимизирует средний риск $\rho(P, d)$ или минимизирует максимум условного риска $\rho(\omega_j, d)$ (критерий минимакса).

Оптимальное решающее правило минимизации среднего риска называется *байесовским правилом*. Из соотношения (6.8) следует, что достаточно рассмотреть каждый вектор x в отдельности и минимизировать функцию $\rho_x(P, d)$. Если d^* — оптимальное решение в смысле минимума среднего риска, то

$$\rho_x(P, d^*) \leq \rho_x(P, d),$$

т. е.

$$\sum_{j=1}^m L(\omega_j, d^*) P(\omega_j) p(x | \omega_j) \leq \sum_{j=1}^m L(\omega_j, d) P(\omega_j) p(x | \omega_j).$$

Для функции потерь вида

$$L(\omega_j, d_i) = 1 - \delta_{ji} = \begin{cases} 0 & \text{при } i = j, \\ 1 & \text{при } i \neq j \end{cases}$$

средний риск является также вероятностью ложного распознавания и байесовское решающее правило приводит к гипотезе $x \sim \omega_i$ при $d_1 = d^*$, если

$$P(\omega_i) p(x | \omega_i) \geq P(\omega_j) p(x | \omega_j) \quad (6.9)$$

для всех $j = 1, 2, \dots, m$.

Определим *отношение правдоподобия* между классами следующим образом:

$$\lambda = \frac{p(x | \omega_i)}{p(x | \omega_j)}.$$

Тогда неравенство (6.9) примет вид $d^* = d_i$, если $\lambda \geq \frac{P(\omega_j)}{P(\omega_i)}$ для всех $j = 1, 2, \dots, m$.

Если информация об априорных вероятностях $P(\omega_j)$ отсутствует, то классификация строится на основе минимаксного критерия по отношению к наименее благоприятному априорному распределению. Из условия (6.9) получаем *разделяющую* функцию

$$D_i(X) = P(\omega_i) p(x | \omega_i), \quad i = 1, 2, \dots, m,$$

или эквивалентную ей функцию

$$D_i(X) = \ln(P(\omega_i) p(x | \omega_i)), \quad i = 1, 2, \dots, m.$$

Решающая граница между областями в Ω , разделяющая образы ω_i и ω_j , определяется условием

$$P(\omega_i) p(x | \omega_i) - P(\omega_j) p(x | \omega_j) = 0,$$

или

$$\ln \frac{P(\omega_i) p(x | \omega_i)}{P(\omega_j) p(x | \omega_j)} = 0. \quad (6.10)$$

Пример 1. Рассмотрим статическую задачу принятия решения в случае, когда пространства Ω и D состоят из двух точек.

Таблица 6.4

Функция потерь

Ω	D	
	d_1	d_2
ω_1	0	l_1
ω_2	l_2	0

Пусть $\Omega = \{\omega_1, \omega_2\}$, $D = \{d_1, d_2\}$, функция потерь $L(\omega, d)$ задана в табл. 6.4, $\alpha(\delta)$ — условная вероятность для любой решающей функции δ принятия решения d_2 при истинности ω_1 , $\beta(\delta)$ — условная вероятность принятия решения d_1 при истинности ω_2 , $l_1, l_2 > 0$.

Другими словами, $\alpha(\delta)$ и $\beta(\delta)$ — вероятности того, что δ предписывает неправильные решения в случаях ω_1 и ω_2 соответственно. Пусть априорное распределение параметра ω задано: $P(\omega_1) = p$, где $0 < p < 1$. С учетом функции потерь $L(\omega, d)$ риск $\rho(p, \delta)$ решающей функции δ запишется в виде

$$\rho(p, \delta) = l_1 p \alpha(\delta) + l_2 (1 - p) \beta(\delta).$$

В каждой конкретной задаче необходимо минимизировать эту комбинацию. Согласно лемме Неймана — Пирсона решающая функция, доставляющая минимум линейной комбинации $l_1 p \alpha(\delta) + l_2 (1 - p) \beta(\delta)$, определяется отношением $f_2(x | \omega_2) / f_1(x | \omega_1)$. Тогда $\delta^*(x) = d_1$, если $a f_1(x | \omega_1) > b f_2(x | \omega_2)$, и $\delta^*(x) = d_2$, если $a f_1(x | \omega_1) < b f_2(x | \omega_2)$. Здесь $a = l_1 p$, $b = l_2 p$.

Если $p(x | \omega_i)$ — плотность многомерного гауссова распределения со средним вектором M_i и ковариационной матрицей K_i , $i = 1, 2, \dots, m$:

$$p(x | \omega_i) = \frac{\exp \left\{ -\frac{1}{2} (x - M_i)^T K_i^{-1} (x - M_i) \right\}}{(2\pi)^{N/2} |K_i|^{1/2}},$$

то уравнение решающей границы согласно соотношению (6.10) имеет вид

$$\begin{aligned} & \ln \frac{P(\omega_i)}{P(\omega_j)} - \frac{1}{2} \ln \frac{|K_i|}{|K_j|} - \\ & - \frac{1}{2} \left[(x - M_i)^T K_i^{-1} (x - M_i) - (x - M_j)^T K_j^{-1} (x - M_j) \right] = 0, \\ & i, j = 1, 2, \dots, m. \end{aligned}$$

При $K_i = K_j = K$ имеем

$$x^T K^{-1}(M_i - M_j) - \frac{1}{2}(M_i + M_j)^T K^{-1}(M_i - M_j) + \ln \frac{P(\omega_i)}{P(\omega_j)} = 0.$$

Получили уравнение гиперплоскости. При $P(\omega_i) = P(\omega_j)$ имеем

$$x^T K^{-1}(M_i - M_j) - \frac{1}{2}(M_i + M_j)^T K^{-1}(M_i - M_j) = 0.$$

В действительности должны получить «полосу», так как x , M и K определяются в процессе наблюдений с погрешностями.

Рассмотрим решающую процедуру с фиксированным объемом выборки.

Пример 2. Найти решающую границу для ω_1 и ω_2 , если

$$p(x | \omega_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp \left\{ -\frac{1}{2} \frac{(x - M_i)^2}{\sigma_i^2} \right\}, \quad i = 1, 2.$$

В общем случае (при $\sigma_1 \neq \sigma_2$) согласно соотношению (6.10) решающая граница определяется из условия

$$\ln \frac{P(\omega_1)}{P(\omega_2)} + \ln \frac{p(x | \omega_1)}{p(x | \omega_2)} = 0,$$

или

$$\ln \frac{P(\omega_1)}{P(\omega_2)} + \ln \frac{\sigma_2}{\sigma_1} - \frac{(x - M_1)^2}{2\sigma_1^2} + \frac{(x - M_2)^2}{2\sigma_2^2} = 0.$$

При $\sigma_1 = \sigma_2 = \sigma$ получим

$$\ln \frac{P(\omega_1)}{P(\omega_2)} - \frac{M_1 - M_2}{2\sigma^2} (M_1 - M_2 - 2x) = 0,$$

откуда решающая граница для ω_1 и ω_2 определяется следующим образом:

$$x = \frac{\sigma^2 \ln \frac{P(\omega_1)}{P(\omega_2)}}{M_2 - M_1} + \frac{M_1 + M_2}{2}.$$

Пример 3. Найти решающую границу для плотности двумерного гауссова распределения $p(x | \omega_i)$, $i = 1, 2$, со средними векторами (m_{ix}, m_{iy}) , $i = 1, 2$, и с общей ковариационной матрицей

$K = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$ при $P(\omega_1) = P(\omega_2)$. Разделяющая граница в данном случае будет прямой линией, уравнение которой имеет вид

$$(x, y) \begin{pmatrix} \sigma_1^{-2} & 0 \\ 0 & \sigma_2^{-2} \end{pmatrix} \begin{pmatrix} m_{1x} - m_{2x} \\ m_{1y} - m_{2y} \end{pmatrix} - \\ - \frac{1}{2} \begin{pmatrix} m_{1x} + m_{2x} \\ m_{1y} + m_{2y} \end{pmatrix}^T \begin{pmatrix} \sigma_1^{-2} & 0 \\ 0 & \sigma_2^{-2} \end{pmatrix} \begin{pmatrix} m_{1x} - m_{2x} \\ m_{1y} - m_{2y} \end{pmatrix} = 0,$$

или

$$x \cdot \frac{m_{1x} - m_{2x}}{\sigma_1^2} + y \cdot \frac{m_{1y} - m_{2y}}{\sigma_2^2} - \frac{1}{2} \left[\frac{m_{1x}^2 - m_{2x}^2}{\sigma_1^2} + \frac{m_{1y}^2 - m_{2y}^2}{\sigma_2^2} \right] = 0.$$

§ 6.4. Методы детерминистской классификации

Концепцию классификации образов можно выразить в терминах разбиения пространства признаков или отображения пространства признаков в пространство решений. Пусть для каждого входного образа измеряется N признаков, каждое множество из N признаков называют *вектором признаков* x или точкой в N -мерном пространстве признаков X_Ω .

Задача классификации образов заключается в распределении всех возможных векторов (или точек) в пространстве признаков по соответствующим классам образов: пространство признаков разделяется на взаимно непересекающиеся области, каждая из которых соответствует некоторому классу образов. Математическая задача классификации образов может быть сформулирована при использовании разделяющей функции.

Пусть имеется m возможных классов образов $\omega_j, j = 1, 2, \dots, m$, $x = (x_1, \dots, x_N)^T$ — вектор замеров признаков. Тогда *разделяющая функция* $D_j(x)$, относящаяся к классу образов $\omega_j, j = 1, 2, \dots, m$, такова, что если входной образ, представленный вектором признаков x , принадлежит классу ω_i , то величина $D_i(x)$ должна быть наибольшей ($x \sim \omega_i$). Для всех значений признака $x \sim \omega_i$ имеем

$$D_i(x) > D_j(x), \quad i, j = 1, 2, \dots, m, \quad i \neq j. \quad (6.11)$$

В пространстве признаков X_Ω граница разбиений, называемая *решающей границей*, между областями, относящимися соответственно к классам ω_i и ω_j , выражается уравнением $D_i(x) - D_j(x) = 0$. Можно предложить много различных форм для записи функций $D_i(x)$, удовлетворяющих неравенству (6.11).

1. *Линейные разделяющие функции.* В качестве $D_i(x)$ берется линейная комбинация измеренных признаков x_1, x_2, \dots, x_N , т. е.

$$D_i(x) = \sum_{k=1}^N w_{ki} x_k + w_{i,N+1}, \quad i = 1, 2, \dots, m.$$

Решающая граница между областями ω_i и ω_j в пространстве Ω имеет вид

$$D_i(x) - D_j(x) = \sum_{k=1}^N w_k x_k + w_{N+1} = 0, \quad (6.12)$$

где $w_k = w_{ik} - w_{jk}$, $w_{N+1} = w_{i,N+1} - w_{j,N+1}$.

Уравнение (6.12) является уравнением гиперплоскости. При $N = 2$ получаем уравнение прямой линии.

2. *Полиномиальные разделяющие функции.* Полиномиальная разделяющая функция r -й степени имеет вид

$$D_i(x) = w_{i1} f_1(x) + w_{i2} f_2(x) + \dots + w_{il} f_l(x) + w_i, \quad i = 1, 2, \dots, m,$$

где $f_j(x)$, $j = 1, 2, \dots, l$, являются формами

$$x_{k_1}^{n_1} x_{k_2}^{n_2} \dots x_{k_r}^{n_r}, \quad k_1, k_2, \dots, k_r = 1, 2, \dots, N, \quad n_1, n_2, \dots, n_r \in \{0; 1\}.$$

При $r = 2$ разделяющая функция называется *квадратичной*. Тогда

$$f_j(x) = x_{k_1}^{n_1} x_{k_2}^{n_2}, \quad k_1, k_2 = 1, 2, \dots, N, \quad n_1, n_2 \in \{0; 1\},$$

и разделяющая функция D_i имеет вид

$$D_i(x) = \sum_{k=1}^N w_{ki} x_k^2 + \sum_{j=1}^{N-1} \sum_{k=j+1}^N b_{jki} x_j x_k + \sum_{j=1}^N w_{ji} x_j + w_{l+1,i},$$

где $l = \frac{1}{2} N(N+3)$.

В общем случае границей для квадратичных разделяющих функций является поверхность второго порядка. В частных случаях это будет гиперсфера, гиперэллипсоид и гиперэллипсоидальный цилиндр.

3. *Классификатор по минимальному расстоянию*. Пусть задано m опорных векторов R_i , $i = 1, 2, \dots, m$, где $R_i \sim \omega_i$. Входной сигнал x предполагают принадлежащим ω_i , если расстояние между x и R_i , равное $\|x - R_i\|$, минимально. Это расстояние можно определить, например, следующим образом:

$$\|x - R_i\| = [(x - R_i)^T(x - R_i)]^{1/2},$$

или

$$\|x - R_i\|^2 = x^T x - x^T R_i - x R_i^T + R_i^T R_i.$$

Поскольку произведение $x^T x$ не зависит от i , то разделяющая функция для классификатора по минимальному расстоянию имеет вид

$$D_i(x) = x^T R_i + x R_i^T - R_i^T R_i, \quad i = 1, 2, \dots, m.$$

Это линейная функция (линейный классификатор).

4. *Обучение в линейном классификаторе*. Введем дополнительный вектор $y = (x_1, \dots, x_N, 1)^T = (x, 1)^T$. Пусть в результате наблюдений получены два множества дополнительных векторов, принадлежащих ω_1 и ω_2 соответственно. Для разделимых множеств

$$y^T w > 0, \quad y \sim \omega_1,$$

$$y^T w < 0, \quad y \sim \omega_2, \quad w = (w_1, \dots, w_N, w_{N+1})^T.$$

Если классификатор дает ошибочный или неопределенный результат при выбранных w , то берут новый вектор весов $w' = w \pm \alpha y$, $\alpha > 0$ (поправочное дополнение). Вначале w выбирается произвольно. В результате итерационного процесса получаем значение w' , которое считается истинным.

§ 6.5. Последовательная решающая модель для классификации образов

Ранее была рассмотрена решающая процедура с фиксированным объемом выборки. Если стоимость измерений признаков высока либо требуется сложное оборудование, значительное время или сложные и рискованные операции, уместно применить *последовательную* решающую процедуру. Последовательный процесс измерений признаков заканчивается (принимается решение),

когда достигнута достаточная или необходимая точность классификации.

Естественно, признаки должны быть расположены в таком порядке, чтобы измерения давали окончательное решение как можно раньше. Задача упорядочения признаков является специальной задачей в системах последовательного распознавания.

В случае двух классов образов можно применить *последовательный критерий отношения вероятностей Вальда* (п. к. о. в.). Он построен для решения задачи о выборе между двумя простыми гипотезами. Пусть случайная величина x обладает функцией плотности $p(x, \theta)$, где θ — испытываемый параметр. Задача заключается в проверке гипотезы H_0 о том, что $\theta = \theta_0$, против гипотезы H_1 , состоящей в том, что $\theta = \theta_1$. Критерий выбирает гипотезу H_0 или H_1 на основе наблюдений x_1, x_2, \dots, x_n . Допустим, что если гипотеза H_0 истинна, мы хотим получить решение в пользу выбора гипотезы H_0 с вероятностью, не меньшей $1 - \alpha$, а если истинна гипотеза H_1 , то решение в пользу выбора гипотезы H_1 должно иметь вероятность, не меньшую $1 - \beta$.

Для последовательного критерия с постоянным объемом выборки оптимальное решение этой задачи можно получить, используя критерий Неймана — Пирсона: при данном числе наблюдений n критерий, обеспечивающий наименьшее β (наиболее мощный критерий), определяется отношением правдоподобия λ_n , имеющим вид

$$\lambda_n = \prod_{i=1}^n \frac{p(x_i | H_0)}{p(x_i | H_1)} = \frac{p_n(x | H_0)}{p_n(x | H_1)}. \quad (6.13)$$

Гипотеза H_0 принимается или отвергается, когда λ_n соответственно меньше или больше некоторой постоянной. Значение этой постоянной может быть выбрано так, чтобы критерий давал нужную величину α (ошибку 1-го рода). Однако можно выбрать и так, чтобы критерий имел заданную мощность $1 - \beta$, где β — ошибка 2-го рода.

Последовательный критерий отношения вероятностей аналогичен изложенному критерию и обладает аналогичными оптимальными свойствами: наблюдения производят до тех пор, пока выполняется условие $B < \lambda_n < A$, где A и B — соответственно верхняя и нижняя границы (числа). Наблюдения прекращаются и

принимается решение в пользу выбора гипотезы H_0 , как только будет выполнено неравенство $\lambda_n \geq A$; решение принимается в пользу выбора гипотезы H_1 , когда $\lambda_n \leq B$. Постоянные A и B называются соответственно *верхним* и *нижним* порогами (*останавливающими границами*). Они могут быть выбраны так, чтобы приблизительно получить заданные вероятности ошибок α и β .

Пусть на n -м шаге оказалось, что

$$\lambda_n = A. \quad (6.14)$$

Это указывает на окончательное решение о принятии гипотезы H_0 . Из условий (6.13) и (6.14) получим

$$p(x | H_0) = A p(x | H_1),$$

что эквивалентно равенству

$$\int p(x | H_0) dx = A \int p(x | H_1) dx. \quad (6.15)$$

Интегралы берутся по области, которая содержит все замеры, приводящие к выбору гипотезы H_0 . Из условия (6.15) получим $1 - \alpha = A\beta$.

Пусть теперь $\lambda_n = B$; тогда $\alpha = B(1 - \beta)$. Отсюда имеем

$$A = \frac{1 - \alpha}{\beta}, \quad B = \frac{\alpha}{1 - \beta}.$$

Эти соотношения строго выполняются для непрерывных замеров; для дискретных замеров могут быть получены вероятности ошибок, отличающиеся от α и β вследствие некоторого превышения порогов. Это отличие несущественно. Тем не менее, в общем случае

$$A \leq \frac{1 - \alpha}{\beta}, \quad B \geq \frac{\alpha}{1 - \beta}.$$

Рассмотрим пример того, как изменяются решающие границы при n замерах признаков.

Пример. Пусть x_1, x_2, \dots, x_n — независимые замеры признаков с одномерной гауссовой функцией плотности $p(x_j | \omega_i)$, $j = 1, 2, \dots, n$, $i = 1, 2$, со средним значением m_i и дисперсией σ^2 . Для простоты будем вычислять $\ln \lambda_n$. После получения признака x_1 имеем

$$\begin{aligned}\ln \lambda_1 &= \ln \frac{p(x_1 | \omega_1)}{p(x_1 | \omega_2)} = \ln \frac{(\sigma\sqrt{2\pi})^{-1} \exp\left\{-\frac{(x_1 - m_1)^2}{2\sigma^2}\right\}}{(\sigma\sqrt{2\pi})^{-1} \exp\left\{-\frac{(x_1 - m_2)^2}{2\sigma^2}\right\}} = \\ &= \frac{1}{\sigma^2} \left[(m_1 - m_2)x_1 - \frac{1}{2}(m_1^2 - m_2^2) \right].\end{aligned}$$

Сравниваем $\ln \lambda_1$ с $\ln A$ и $\ln B$. Если

$$x_1 > \frac{\sigma^2}{m_1 - m_2} \ln A + \frac{1}{2}(m_1 + m_2),$$

то $x \sim \omega_1$. Если

$$x_1 \leq \frac{\sigma^2}{m_1 - m_2} \ln B + \frac{1}{2}(m_1 + m_2),$$

то $x \sim \omega_2$. Если

$$\frac{\sigma^2}{m_1 - m_2} \ln B + \frac{1}{2}(m_1 + m_2) < x_1 < \frac{\sigma^2}{m_1 - m_2} \ln A + \frac{1}{2}(m_1 + m_2),$$

то следует продолжить измерения. После второго измерения получим

$$\ln \lambda_2 = \ln \frac{p(x_1 | \omega_1)}{p(x_1 | \omega_2)} + \ln \frac{p(x_2 | \omega_1)}{p(x_2 | \omega_2)} = \frac{m_1 - m_2}{\sigma^2} [x_1 + x_2 - (m_1 + m_2)].$$

Если

$$x_1 + x_2 > \frac{\sigma^2}{m_1 - m_2} \ln A + m_1 + m_2,$$

то $x \sim \omega_1$. Если

$$x_1 + x_2 \leq \frac{\sigma^2}{m_1 - m_2} \ln B + m_1 + m_2,$$

то $x \sim \omega_2$. Если

$$\frac{\sigma^2}{m_1 - m_2} \ln B + m_1 + m_2 < x_1 + x_2 < \frac{\sigma^2}{m_1 - m_2} \ln A + m_1 + m_2,$$

то проводится новое измерение x_3 и т. д. На n -м шаге процесса имеем

$$\ln \lambda_n = \sum_{i=1}^n \ln \frac{p(x_i | \omega_1)}{p(x_i | \omega_2)} = \frac{m_1 - m_2}{\sigma^2} \sum_{i=1}^n \left[x_i - \frac{1}{2}(m_1 + m_2) \right].$$

Процедура классификации образов ω_1 и ω_2 будет следующей. Если

$$\sum_{i=1}^n x_i > \frac{\sigma^2}{m_1 - m_2} \ln A + \frac{n}{2}(m_1 + m_2), \quad (6.16)$$

то $x \sim \omega_1$. Если

$$\sum_{i=1}^n x_i \leq \frac{\sigma^2}{m_1 - m_2} \ln B + \frac{n}{2}(m_1 + m_2), \quad (6.17)$$

то $x \sim \omega_2$. Если

$$\frac{\sigma^2}{m_1 - m_2} \ln B + \frac{n}{2}(m_1 + m_2) < \sum_{i=1}^n x_i < \frac{\sigma^2}{m_1 - m_2} \ln A + \frac{n}{2}(m_1 + m_2),$$

то измеряется x_{n+1} .

Решающие границы, определяемые формулами (6.16), (6.17), при знаках равенства представляют собой две параллельные гиперплоскости в пространстве признаков. Ширина области неопределенности равна $\frac{\sigma^2}{m_1 - m_2} \ln \frac{A}{B}$, т. е. пропорциональна $\frac{\sigma^2}{m_1 - m_2}$.

При заданных вероятностях ошибок α и β среднее число измерений, необходимое для завершения процесса, пропорционально σ^2 и обратно пропорционально разности $m_1 - m_2$.

Последовательный критерий отношения вероятностей дает минимум среднего числа наблюдений; математическое ожидание среднего числа наблюдений запишется в виде

$$M_1(n) = \frac{(1 - \alpha) \ln A + \alpha \ln B}{M_1(z)},$$

когда H_0 истинно, и

$$M_2(n) = \frac{\beta \ln A + (1 - \beta) \ln B}{M_2(z)},$$

когда H_1 истинно, где $z = \ln \frac{p_n(x | H_0)}{p_n(x | H_1)}$.

Процедура п. к. о. в. по существу не зависит от априорных вероятностей $P(H_i)$, $i=0,1$, хотя вероятность ошибок зависит от априорных данных.

Иногда необходимо прервать процедуру наблюдений и принять решение при $n = N$. Это может быть сделано усечением последовательного процесса при $n = N$: выполняется обычная процедура п. к. о. в. либо до получения решения, либо до N -го шага. Если на N -м шаге решение не получено, принимается гипотеза H_0 при $\lambda_N > 1$ или принимается гипотеза H_1 при $\lambda_N < 1$.

Для большого числа гипотез предложен *обобщенный последовательный критерий отношения вероятностей* (о. п. к. о. в.). Пусть имеется m гипотез. На n -м шаге обобщенные последовательные отношения вероятностей для каждой гипотезы определяются следующим образом:

$$U_n(x | H_i) = \frac{p_n(x | H_i)}{\left[\prod_{q=1}^m p_n(x | H_q) \right]^{1/m}}, \quad i = 1, 2, \dots, m.$$

Решающее правило о. п. к. о. в. состоит в следующем. Отношение $U_n(x | H_i)$ сравнивается с останавливающей границей $A(H_i)$ для гипотезы H_i . Гипотеза H_i исключается из рассмотрения, если $U_n(x | H_i) < A(H_i)$, $i = 1, 2, \dots, m$. Останавливающая граница для гипотезы H_i определяется соотношением

$$A(H_i) = \frac{1 - e_{ii}}{\left[\prod_{q=1}^m (1 - e_{iq}) \right]^{1/m}}, \quad i = 1, 2, \dots, m,$$

где e_{iq} — вероятность принятия гипотезы H_i , когда в действительности истинна гипотеза H_q .

После исключения гипотезы H_i общее число гипотез становится на единицу меньше и составляется новое множество обобщенных последовательных отношений вероятностей. Последовательно исключая гипотезы, получим одну, которая и принимается за истинную. При $m = 2$ о. п. к. о. в. эквивалентен п. к. о. в. и сохраняет его свойство оптимальности: при заданных α и β не существует другой процедуры, которая обладает меньшими значени-

ями вероятностей ошибок или среднего риска и дает выигрыш в среднем числе измерений признаков по сравнению с последовательной процедурой классификации.

Между границами A и B заключена область неопределенности, в которой не может быть принято окончательное решение (нулевая область).

В классификации образов на основе о. п. к. о. в. при $n = N$ входной образ полагают принадлежащим тому классу, обобщенное последовательное отношение вероятностей для которого имеет наибольшее значение.

В последовательном анализе при повышении верхнего порога A и понижении нижнего порога B по меньшей мере одна из вероятностей ошибок, α или β , уменьшается, если новый п. к. о. в. не оказывается эквивалентен старому. Поэтому можно начинать процедуру с относительно больших значений порогов, и, постепенно уменьшая их, получим среднее значение испытаний не таким большим, как в случае, когда в течение всего процесса используется малая величина порога.

§ 6.6. Байесовская последовательная решающая процедура

Используем здесь определения, данные в § 6.3.

При фиксированном объеме выборки и испытании m статистических гипотез оптимальное решение d^* выбирается так, чтобы минимизировать величину средних потерь

$$\rho(P, d) = \sum_{i=1}^m P(H_i) \rho(H_i, d),$$

где $\rho(H_i, d) = \int_{\Omega} L(H_i, d) p(x | H_i) d\omega$ — условные потери (условный риск). Другими словами, требуется найти

$$\rho(P, d^*) = \min_i \rho(P, d_i).$$

При

$$L(H_i, d_j) = 1 - \delta_{ij} = \begin{cases} 0, & \text{если } i = j, \\ 1, & \text{если } i \neq j, \end{cases}$$

байесовским решением будет $d^* = d_i$, если выполнены условия

$$P(H_i) p(x | H_i) \geq P(H_j) p(x | H_j), \quad j = 1, 2, \dots, m.$$

Если $\lambda = \frac{p(x|H_i)}{p(x|H_j)}$, имеем решение $d^* = d_i$ при $\lambda \geq \frac{P(H_j)}{P(H_i)}$ для $j = 1, 2, \dots, m$.

На каждом шаге, используя предварительную информацию, делают выбор: принять окончательное решение или провести следующее наблюдение. Проблема состоит в стоимости наблюдений. На каждом шаге приходится соизмерять стоимость выполнения будущих наблюдений со средним выигрышем, даваемым принятым решением.

Пусть имеем m гипотез с априорными вероятностями $P(H_i)$, $i = 1, 2, \dots, m$. Для любого варианта последовательной выборки $s_j \in S$, $j = 1, \dots, N$, определим риск:

$$\rho(H_i, d) = \sum_{i=1}^m P(H_i) \sum_{j=1}^N \int [c_j(x) + L(H_i, d_j(x))] p(x|H_i) dx,$$

где $c_j(x)$ — стоимость наблюдений x_1, x_2, \dots, x_j , $d_j(x)$ — решающая функция, основанная на замерах x_1, x_2, \dots, x_j .

Далее рассмотрим последовательные решения для двухточечной модели. Пусть проводятся поочередно наблюдения x_1, x_2, \dots, x_n над неизвестным объектом. Наблюдения подчиняются некоторым распределениям, зависящим от неизвестного параметра ω . Получают *последовательную* выборку. После каждого наблюдения x_i можно оценить информацию об ω , полученную на основе наблюдений x_1, x_2, \dots, x_i . Каждое наблюдение имеет некоторую стоимость. Предположим, что можно проводить наблюдения последовательно, делая после каждого наблюдения вывод о том, следует ли принять решение $d \in D$ или провести очередное наблюдение. Последовательный выбор может дать, например, выигрыш при проверке большой партии изделий путем извлечения m изделий, из которых допускается k бракованных изделий. Последовательный отбор прекращается, если будет отобрано k бракованных изделий или $m - k$ годных изделий. Как правило, это происходит до отбора всех m изделий. В первом случае вся партия будет забракована, во втором — принята.

Рассмотрим задачу последовательного принятия решения при параметрическом пространстве $\Omega = \{\omega_1, \omega_2\}$ и пространстве решений $D = \{d_1, d_2\}$.

Функция потерь задана в табл. 6.5, где $\lambda_i > 0$, $i=1,2$. Мы можем наблюдать последовательную выборку x_1, x_2, \dots ; каждое наблюдение стоит c усл. ед. Пусть $f_i(x|\omega_i)$ — условная о. в. п. наблюдения x при ω_i . Поскольку параметр ω может принимать всего два значения, то его распределение на каждом шаге процесса выбора задается одним числом $P(\omega_1) = p$, $0 \leq p \leq 1$.

Таблица 6.5

Функция потерь

Ω	D	
	d_1	d_2
ω_1	0	λ_1
ω_2	λ_2	0

Риск принятия решения до наблюдения находим по формуле

$$\rho_0(p) = \min \{ \lambda_1 p; \lambda_2 (1-p) \}.$$

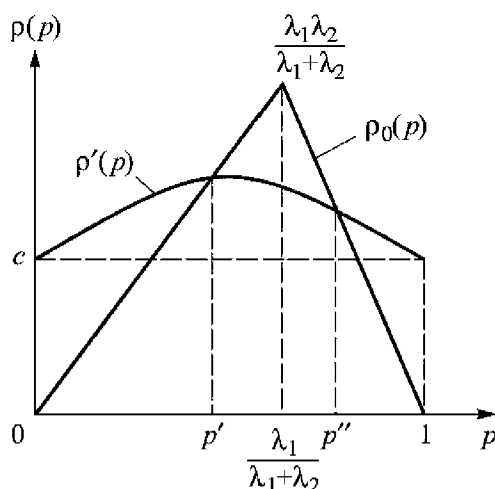
Обозначим через Δ класс всех процедур последовательного решения δ , требующих хотя бы одного наблюдения, и пусть

$$\rho'(p) = \inf_{\delta \in \Delta} \rho(p, \delta).$$

Таким образом, байесовский риск $\rho^*(p)$ удовлетворяет соотношению

$$\rho^*(p) = \min \{ \rho_0(p); \rho'(p) \}.$$

Ранее указывалось, что $\rho'(p)$ — вогнутая непрерывная функция, заданная на отрезке $[0, 1]$. Поскольку в формулу риска входит цена c , то $\rho'(0) = c$ и $\rho'(p) \geq c$ при всех значениях $p \in [0, 1]$ (рис. 6.4).

Рис. 6.4. Связь минимального риска $\rho(p)$ и цены наблюдения

Пусть Σ^* — множество тех значений p , для которых процесс заканчивается, т. е. $\Sigma^* = \{p: \rho(p) \leq \rho'(p)\}$. Предположим, что

$$\rho'\left(\frac{\lambda_1}{\lambda_1 + \lambda_2}\right) < \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2};$$

тогда Σ^* является объединением отрезков $[0, p']$ и $[p'', 1]$, причем p' и p'' удовлетворяют уравнениям

$$\lambda_1 p' = \rho(p'), \quad \lambda_2 (1 - p'') = \rho(p'').$$

Если это условие не выполнено, то Σ^* совпадает со всем отрезком $[0, 1]$. В этом случае проводить наблюдения не имеет смысла. Множество Σ^* задает байесовскую процедуру последовательного решения. Трудность состоит в получении информации о функции $\rho'(p)$, достаточной для того, чтобы явным образом определить p' и p'' .

Предположим, что априорная вероятность p удовлетворяет условию $p' < p < p''$. В этом случае целесообразно провести первое наблюдение. Пусть $\xi(x_1, x_2, \dots, x_n)$ — априорная вероятность того, что $\omega = \omega_i$ при проведенных x_i , $i = 1, 2, \dots, n$, наблюдениях; тогда дальнейшие наблюдения необходимо проводить, если выполнено соотношение $p' < \xi(x_1, x_2, \dots, x_n) < p''$. Если нарушается первое неравенство, то принимается решение d_2 , если второе — принимается решение d_1 .

Апостериорную вероятность можно записать следующим образом:

$$\xi(x_1, \dots, x_n) = \left[1 + \frac{1-p}{p} \frac{f_2(x_1) \dots f_2(x_n)}{f_1(x_1) \dots f_1(x_n)} \right]^{-1}.$$

Введем постоянные A и B :

$$A = \frac{p(1-p'')}{(1-p)p''}, \quad B = \frac{p(1-p')}{(1-p)p'}.$$

Поскольку $p' < p < p''$, то $A < 1$, $B > 1$. Дальнейшие наблюдения следует проводить, если

$$A < \frac{f_2(x_1) \dots f_2(x_n)}{f_1(x_1) \dots f_1(x_n)} < B.$$

При нарушении неравенств принимаются соответственно решения d_1 или d_2 .

Процедура последовательного принятия решения такого вида называется *последовательным критерием отношения вероятностей*.

Вычислим вероятность принятия решений d_1 и d_2 и найдем среднюю стоимость выбора для последовательной процедуры. В большинстве задач точные формулы получить не удастся, но существуют простые приближенные формулы. Для последовательности наблюдений x_1, \dots, x_n , приводящих к решению d_1 , выполняется соотношение

$$\prod_{i=1}^n f_2(x_i) \leq A \prod_{i=1}^n f_1(x_i) \quad \text{или} \quad P(d_1 | \omega_2) \leq AP(d_1 | \omega_1).$$

Для последовательности наблюдений, приводящих к решению d_2 , получим

$$\prod_{i=1}^n f_2(x_i) \geq B \prod_{i=1}^n f_1(x_i) \quad \text{или} \quad P(d_2 | \omega_2) \geq BP(d_2 | \omega_1).$$

Однако $P(d_1 | \omega_2) + P(d_2 | \omega_1) = 1$, т. е. точка $(P(d_2 | \omega_1), P(d_1 | \omega_2))$ лежит в области Φ (рис. 6.5). Можно считать приведенные соотношения приближенными равенствами, т. е.

$$P(d_1 | \omega_1) = 1 - P(d_2 | \omega_1) \approx \frac{B-1}{B-A},$$

$$P(d_1 | \omega_2) = 1 - P(d_2 | \omega_2) \approx \frac{A(B-1)}{B-A}.$$

Эти приближенные значения являются координатами точки $K(P(d_2 | \omega_1), P(d_1 | \omega_2))$ (см. рис. 6.5).

Введем случайную величину

$$z_i = \ln \frac{f_2(x_i)}{f_1(x_i)}, \quad i = 1, 2, \dots, n,$$

и положим $\ln A = a < 0$, $\ln B = b > 0$; тогда согласно последовательному критерию отношения вероятностей необходимо продолжать наблюдения при выполнении условия $a < \sum_{i=1}^n z_i < b$.

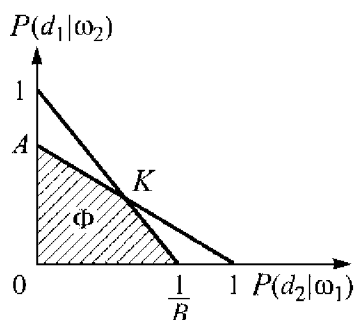


Рис. 6.5. Вероятности принятия решений d_1 и d_2

При фиксированном значении ω_i , $i = 1, 2$, случайные величины x_1, x_2, \dots, x_n независимы и одинаково распределены. Следовательно, этими же свойствами обладают и случайные величины z_1, z_2, \dots, z_n . Для любых заданных значений a и b все моменты распределения случайного числа наблюдений N конечны и процесс выбора заканчивается с вероятностью 1, т. е. $P(N < \infty) = 1$, $M(N^k) < \infty$ для $k = 1, 2, \dots$

Если z_1, z_2, \dots, z_n — последовательность независимых одинаково распределенных случайных величин, для которых $M(z_i) = m$, $i = 1, 2, \dots, N$, то для всякой последовательной процедуры с моментом $M(N) < \infty$ имеет место равенство

$$M\left(\sum_{i=1}^N z_i\right) = m M(N).$$

В общем случае границы a и b можно найти, минимизируя функцию риска $\rho(p, \delta)$. Как правило, это сделать сложно. При малой цене c оптимальная процедура обычно предписывает проведение большого числа наблюдений, величины границ a и b являются большими числами. Тогда получают следующие приближения:

$$P(d_2 | \omega_1) \approx e^{-b}, \quad P(d_1 | \omega_2) \approx e^{-a},$$

$$M(N | \omega_1) \approx \frac{a}{M(z | \omega_1)}, \quad M(N | \omega_2) \approx \frac{b}{M(z | \omega_2)},$$

$$a \approx \ln c - \ln \frac{I_1 \lambda_2 (1-p)}{p}, \quad b \approx \ln \frac{1}{c} + \ln \frac{I_2 \lambda_1 p}{1-p},$$

где $I_1 = -M(z | \omega_1)$, $I_2 = M(z | \omega_2)$; I_1 и I_2 называют *информационными числами*.

§ 6.7. Байесовские методы обучения

Перед тем как система может быть использована для принятия решений (для классификации классов или объектов) по результатам наблюдений, она должна быть «обучена»: должны быть получены (оценены) все ее характеристики вплоть до разделяющих функций. Причем в обучающей выборке результаты наблюдений или заранее отождествлены с конкретными классами — *обучение*

с учителем (с поощрением), или не относятся к конкретным классам — обучение без учителя (без поощрения).

Обучение с поощрением

Проводится серия наблюдений, причем известен класс, к которому относится каждое наблюдение. Известен также общий вид функции плотности. Параметры известной функции плотности $p(x|\omega_i)$, могут быть оценены путем итерационного применения теоремы Байеса (далее параметр ω_i в записи функции плотности не используем).

Пусть существует априорная функция плотности $p_0(\theta)$ с неизвестным параметром θ , отражающая первоначальную информацию о параметре θ . Когда наблюдается последовательность независимых и одинаково распределенных векторов признаков x_1, x_2, \dots, x_n , относящихся к *одному* классу образов, функция стремится к апостериорной плотности $p(\theta|x_1, \dots, x_n)$. После первого наблюдения априорная функция плотности для следующего наблюдения будет иметь вид

$$p(\theta|x_1) = \frac{p(x_1|\theta)p_0(\theta)}{p(x_1)}.$$

После наблюдения x_2 получим

$$p(\theta|x_1, x_2) = \frac{p(x_2|x_1, \theta)p(\theta|x_1)}{p(x_2|x_1)}.$$

В общем случае имеем

$$p(\theta|x_1, \dots, x_n) = \frac{p(x_n|x_{n-1}, \dots, x_1, \theta)p(\theta|x_1, \dots, x_{n-1})}{p(x_n|x_1, \dots, x_{n-1})}. \quad (6.18)$$

Искомая функция плотности может быть вычислена по формуле

$$\begin{aligned} & p(x_{n+1}|x_1, \dots, x_n, \omega_i) = \\ & = \int p(x_{n+1}|x_1, \dots, x_n, \omega_i, \theta)p(\theta|x_1, \dots, x_n, \omega_i)d\theta, \quad n=1, 2, \dots, \end{aligned}$$

где член $p(x_{n+1}|x_1, \dots, x_n, \omega_i, \theta)$ известен, а $p(\theta|x_1, \dots, x_n)$ получается из соотношения (6.18). Известно, что в среднем апостериорная функция плотности улучшает оценку параметра, которая сходится к истинному значению, если только оно не исключено из априорной функции плотности.

Пример. Имеем n обучающих наблюдений x_1, \dots, x_n , проведенных на двух классах образов ω_1 и ω_2 , которые имеют одномерные гауссовы распределения с неизвестными параметрами. Оптимальная решающая граница, минимизирующая вероятность ошибочного распознавания, является функцией априорных вероятностей, средних значений наблюдений и дисперсий.

Совместное распределение в этом случае имеет вид

$$P(x) = \sum_{i=1}^2 P(\omega_i) p(x | \omega_i) = \\ = \frac{1}{2} \frac{1}{\sqrt{2\pi} \cdot \sigma} \left(\exp \left\{ -\frac{(x - m_1)^2}{2\sigma^2} \right\} + \exp \left\{ -\frac{(x - m_2)^2}{2\sigma^2} \right\} \right).$$

Тогда оптимальная решающая граница определяется условием

$$m_n = \frac{1}{n} \sum_{i=1}^n x_i.$$

Обучение без поощрения

Поскольку невозможно точно отнести каждое наблюдение к правильному классу образов, то задача определения решающей границы формируется как *задача оценки параметров общего распределения*. Обучающие наблюдения считают принадлежащими этому совместному распределению, его составляющие могут быть распределениями каждого класса образов или распределениями, соответствующими различным разбиениям пространства наблюдений (признаков). Пусть множество случайных наблюдений x_1, x_2, \dots, x_n может быть разбито на m ячеек z_1^n, \dots, z_m^n .

Совместное распределение определяется следующим образом:

$$P(x) = \sum_{i=1}^m p(x | z_i^n) P(z_i^n), \quad x = (x_1, x_2, \dots, x_n), \quad (6.19)$$

где $p(x | z_i^n)$ — условная плотность вероятности для i -го разбиения, $P(z_i^n)$ — параметр i -го разбиения.

Условная плотность совместного распределения $p(x | \theta)$ может быть построена на основе семейства условных плотностей распределения $p(x | \theta_i, x_i^n)$, $i = 1, 2, \dots, m$, с двумя множествами параметров $\theta = \{\theta_1, \theta_2, \dots, \theta_m\}$ и $P = \{P(z_1^n), P(z_2^n), \dots, P(z_m^n)\}$.

Основное уравнение (6.19) примет вид

$$p(x|\theta) = \sum_{i=1}^m p(x|\theta_i, z_i^n) P(z_i^n).$$

Известно, что класс плотностей совместных распределений, для которых существует единственное решение для θ и $P(z_i^n)$, ограничен. Метод оценки параметров θ и $P(z_i^n)$ зависит от конкретных условий задачи.

Рассмотрим два метода оценки параметров совместных распределений.

Пусть ω_1 и ω_2 — два класса образов. Вид функции плотности $p(x|\omega_i)$, $i = 1, 2$, известен, неизвестным параметром является θ . По наблюдениям x_1, \dots, x_n необходимо найти оценку θ . Если последовательность x_1, x_2, \dots, x_n разбить на всевозможные комбинации по классам ω_1 и ω_2 , то получим 2^n комбинаций. Пусть z_i^n — i -е разбиение последовательности x_1, x_2, \dots, x_n . Тогда апостериорная плотность вероятности будет

$$p(\theta | x_1, \dots, x_n) = \sum_{i=1}^{2^n} p(\theta | x_1, \dots, x_n, z_i^n) P(z_i^n | x_1, \dots, x_n).$$

Задача свелась к обучению с поощрением для каждого из 2^n разбиений. Оценка параметров производится по взвешенной сумме результатов для каждого разбиения с весами, равными

$$P(z_i^n | x_1, \dots, x_n), \quad i = 1, 2, \dots, 2^n.$$

Объем вычислений при этом растет экспоненциально.

Рассмотрим применение другого метода. По теореме Байеса имеем

$$p(\theta | x_1, \dots, x_n) = \frac{p(x_n | \theta, x_1, \dots, x_{n-1}) p(\theta | x_1, \dots, x_{n-1})}{p(x_n | x_1, \dots, x_{n-1})}. \quad (6.20)$$

Пусть обучающие наблюдения условно независимы, т. е. $p(x_{n+1} | \theta, x_1, \dots, x_n) = p(x_{n+1} | \theta)$. Тогда

$$p(x_n | \theta, x_1, \dots, x_{n-1}) = P(\omega_1) p(x_n | \theta, \omega_1) + P(\omega_2) p(x_n | \theta, \omega_2). \quad (6.21)$$

Подставляя равенство (6.21) в формулу (6.20), получим рекуррентное соотношение для оценки θ :

$$\begin{aligned} p(\theta | x_1, \dots, x_n) &= \\ &= p(\theta | x_1, \dots, x_{n-1}) \left[\frac{P(\omega_1) p(x_n | \theta, \omega_1)}{p(x_n | x_1, \dots, x_{n-1})} + \frac{P(\omega_2) p(x_n | \theta, \omega_2)}{p(x_n | x_1, \dots, x_{n-1})} \right]. \end{aligned}$$

Если $p(\theta | x_1, \dots, x_{n-1})$, $p(x_n | \theta, \omega_1)$, $p(x_n | \theta, \omega_2)$, $P(\omega_1)$, $P(\omega_2)$ известны, то можно получить $p(\theta | x_1, \dots, x_n)$. Пусть $P(\omega_1)$ и $P(\omega_2)$ известны. Тогда для нахождения $p(x_n | \theta, \omega_1)$ и $p(\theta | x_1, \dots, x_n)$ при всех значениях θ следует предположить, что к величине θ можно применить конечное квантование, чтобы объем вычислений был конечным.

Для m классов и ряда неизвестных параметров θ_i , относящихся соответственно к ω_i , полагая условную независимость обучающих наблюдений и значений параметров θ_i , получим рекуррентное соотношение для оценки θ_i :

$$p(\theta_i | x_1, \dots, x_n) = \frac{P(\omega_i)p(x_n | \theta_i, \omega_i) + \sum_{j=1}^m P(\omega_j)p(x_n | \omega_j, x_1, \dots, x_{n-1})}{\sum_{j=1}^m P(\omega_j)p(x_n | \omega_j, x_1, \dots, x_{n-1})},$$

$$i = 1, 2, \dots, m.$$

В общем случае начальные оценки либо $P_0(\theta_i)$, либо $P_0(\omega_i)$, $i = 1, 2, \dots, m$, должны быть различны, иначе для всех θ_i будет оцениваться одно и то же значение (так как вычисляется одна величина) и система в целом ничему не будет обучена.

§ 6.8. Обучение с помощью стохастической аппроксимации

Поставим перед собой следующую задачу: найти оценки параметров функций плотностей распределения вероятностей. Здесь так же возможны два случая: в первом наблюдения отождествлены с конкретными классами (обучение с поощрением), во втором рассматриваются совместные распределения.

Обучение с поощрением

Имеем наблюдения x_1, x_2, \dots, x_n , n_i — число наблюдений, соответствующих ω_i , $i = 1, 2, \dots, m$, $n = \sum_{i=1}^m n_i$. Необходимо оценить неизвестную вероятность $P(\omega_i)$, если $\sum_{i=1}^m P(\omega_i) = 1$. Имеем (задаем)

начальную оценку вероятности $P_0(\omega_i)$, $\sum_{i=1}^m P_0(\omega_i) = 1$, и полагаем, что $P(\omega_i) = P_0(\omega_i)$. Последовательные оценки имеют вид

$$P_{k+1}(\omega_i) = P_k(\omega_i) + \gamma_{k+1} \left[\frac{n_i}{n} - P_k(\omega_i) \right], \quad k = 0, 1, 2, \dots$$

Оценим неизвестную функцию плотности $p(x)$ по наблюдениям x_1, x_2, \dots, x_n , предварительно разложив ее в ряд:

$$p(x) = \sum_{i=1}^m c_i \varphi_i(x),$$

$$\int \varphi_i(x) \varphi_j(x) dx = \begin{cases} 0 & \text{при } i \neq j, \\ 1 & \text{при } i = j, \end{cases}$$

$$c_{i,k+1} = c_{i,k} + \gamma_k [\varphi_i(x_k) - c_{i,k}], \quad i = 1, 2, \dots, m, \quad k = 0, 1, 2, \dots,$$

$$1 > \gamma_k > 0, \quad \sum_{k=1}^{\infty} \gamma_k = \infty, \quad \sum_{k=1}^{\infty} \gamma_k^2 < \infty.$$

В частном случае можно найти оценки неизвестных параметров функции плотности по наблюдениям x_1, x_2, \dots, x_n .

Обучение без подощрения

Рассматривается задача оценки параметров в совместном распределении с помощью стохастической аппроксимации.

Предполагается, что:

1) существуют m классов распределений вероятностей, соответствующих m классам образов, априорные вероятности $P(\omega_i)$ которых фиксированы, но неизвестны, $\sum_{i=1}^m P(\omega_i) = 1$;

2) функции распределения вероятностей (или плотности) каждого класса ω_i характеризуются некоторым множеством параметров θ_i ;

3) обучающие наблюдения взяты из совместного распределения, построенного из составляющих распределений:

$$p(x | \theta, \omega) = \sum_{i=1}^m p(x | \omega_i, \theta_i) P(\omega_i);$$

здесь параметры θ и $P(\omega_i)$ неизвестны;

4) существуют несмещенные оценки некоторых статистик $H = \{H(x)\}$, например моментов; известно для каждого шага процесса обучения функциональное отношение $F(H, \theta, p_1) = 0$ между H и множествами параметров θ и p_1 ;

5) имеются дополнительные соотношения $G(\theta, p_1) = 0$, обеспечивающие получение единственного решения для неизвестных параметров θ и p_1 .

Из равенств $F(H, \theta, p_1) = 0$ и $G(\theta, p_1) = 0$ получают оценки θ и p_1 ; $F(\cdot)$ может быть найдено из последовательных оценок $\{H(x)\}$, а $G(\cdot)$ задано априори или с помощью вспомогательных процедур оценки.

Пример 1. Пусть имеем два класса m_1 и m_2 , $P(\omega_1) = p_1$, $P(\omega_2) = 1 - p_1$ — априорные вероятности. Найти оценки параметров функций плотности вероятности.

Каждая функция плотности характеризуется средним значением m_i и дисперсией σ_i^2 , т. е.

$$p(x | \omega_i) \sim p(x | m_i, \sigma_i^2, \omega_i), \quad i = 1, 2.$$

Совместная функция плотности характеризуется параметрами $\theta = \{m_1; m_2; \sigma_1^2; \sigma_2^2\}$, и p_1 задано следующим образом:

$$p(x | \theta, p_1) = p_1 p(x | m_1, \sigma_1^2, \omega_1) + (1 - p_1) p(x | m_2, \sigma_2^2, \omega_2). \quad (6.22)$$

Необходимо найти оценки параметров θ и p_1 по классифицированным обучающим наблюдениям x_1, x_2, \dots, x_n , имеющим плотность распределения $p(x | \theta, p_1)$.

Учитывая условие (6.22), аналитически определим моменты:

$$\begin{aligned} M(x) &= p_1 m_1 + (1 - p_1) m_2 = p_1 (m_1 - m_2) + m_2, \\ M(x^2) &= p_1 (m_1^2 + \sigma_1^2) + (1 - p_1) (m_2^2 + \sigma_2^2), \\ M(x^3) &= p_1 (m_1^3 + 3m_1 \sigma_1^2) + (1 - p_1) (m_2^3 + 3m_2 \sigma_2^2). \end{aligned} \quad (6.23)$$

Теперь те же значения моментов вычислим по результатам наблюдений x_1, x_2, \dots, x_n , подставим их в систему (6.23), из которой найдем искомые параметры. Как правило, параметры последовательно оцениваются посредством оценок моментов совместных распределений. В случае, когда σ_1^2 и σ_2^2 неизвестны и не равны друг другу, необходимо знать моменты смешанного распределе-

ния более высокого порядка для получения достаточных функциональных связей между неизвестными параметрами.

Обычно неоднозначность решения следует ожидать при совместном решении нелинейных уравнений, полученных в результате применения метода моментов. Единственное решение может быть получено только в случае, когда имеется дополнительная информация о параметрах.

Пример 2. Пусть в примере 1 $m_2 = 0$, $\sigma_1 = \sigma_2 = \sigma$. Требуется оценить p_1 , σ , m_1 .

Из системы (6.23) имеем

$$M(x) = p_1 m_1, \quad M(x^2) = p_1 m_1^2 + \sigma^2, \quad M(x^3) = p_1 (m_1^3 + 3m_1 \sigma^2),$$

откуда относительно m_1 получим квадратное уравнение

$$m_1^2 - 3M(x)m_1 + 3M(x^2) - \frac{M(x^3)}{M(x)} = 0.$$

Единственное решение этого уравнения будет при $p_1 = 2/3$, когда дискриминант

$$9M^2(x) - 12M(x^2) + 4\frac{M(x^3)}{M(x)} = m_1^2(3p_1 - 2)^2.$$

равен нулю. В других случаях решение будет не единственно.

§ 6.9. Математическая постановка задачи учета погрешности признаков

Как показано в предыдущих параграфах, в традиционных методах (алгоритмах) решения задач распознавания образов не в полной мере учитывается статистическая природа наблюдений (рис. 6.6). В частности, в процедурах классификации образов функции условных плотностей вероятности векторов признаков и результаты наблюдений, по которым ведется классификация образов, являются соответственно детерминированными функциями и детерминированными величинами.

Когда функции и величины содержат ошибки, получаем смещенные оценки, например, разделяющих функций и, как следствие, увеличение числа неверных решений. Полный учет статистических характеристик наблюдаемых векторов признаков в задачах принятия решения приведет к уменьшению числа ложных решений.

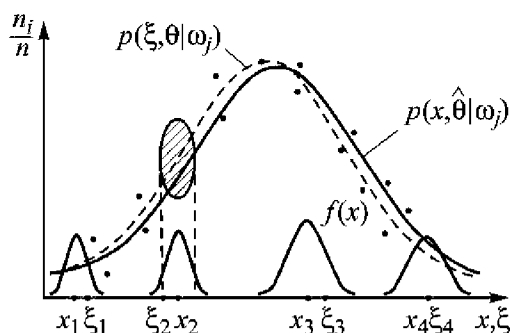


Рис. 6.6. Оценки условной плотности вероятности признака ξ при плотности вероятности $f(x)$ наблюдаемых значений $x_i = \xi_i + \delta_i$, $n_i/n = p(\xi_i, \theta | \omega_j) + \epsilon_i$

Пусть D — пространство решений d , Ω — пространство образов ω (или пространство параметров в статистических задачах принятия решения), R — пространство потерь, которые могут быть получены в результате решения d и исхода (образа) ω . Каждый образ $\omega \in \Omega$ характеризуется вектором признаков ξ , а также считаются известными для каждого класса образов ω_j , $j = 1, 2, \dots, m$, многомерные условные функции плотностей вероятностей векторов признаков $p(\xi | \omega_j)$, которые оцениваются по результатам наблюдений векторов признаков ξ . Координаты векторов ξ в силу различных случайных помех не наблюдаются, а наблюдаются случайные величины x , связанные с ξ следующим соотношением: $x = \xi + \delta$, где δ — случайная аддитивная помеха с известным законом распределения. В частности, во многих случаях можно считать, что δ подчиняется многомерному нормальному закону распределения с нулевым математическим ожиданием и известной ковариационной матрицей $\delta^2 I$ (I — единичная матрица).

В рассмотренных традиционных методах решения статистических задач распознавания образов (статистических задач принятия решения) с фиксированным объемом выборки можно выделить два этапа: на первом этапе по результатам наблюдений случайного вектора признаков x_1, x_2, \dots, x_n получают оценки плотностей условных распределений в виде $p(x | \omega_j)$ и функции распределения образов $P(\omega_j)$, $j = 1, 2, \dots, m$; на втором этапе, считая известными $p(x | \omega_j)$ и $P(\omega_j)$, $j = 1, 2, \dots, m$, наблюдаемый вектор x относят к некоторому классу ω_j , $j = 1, 2, \dots, m$ (рис. 6.7).

На рис. 6.7 $L(d, \omega_j)$ — функция ущерба от принятия решения d , когда выбран объект ω_j ; $P(\omega_j)$ — априорная вероятность по-

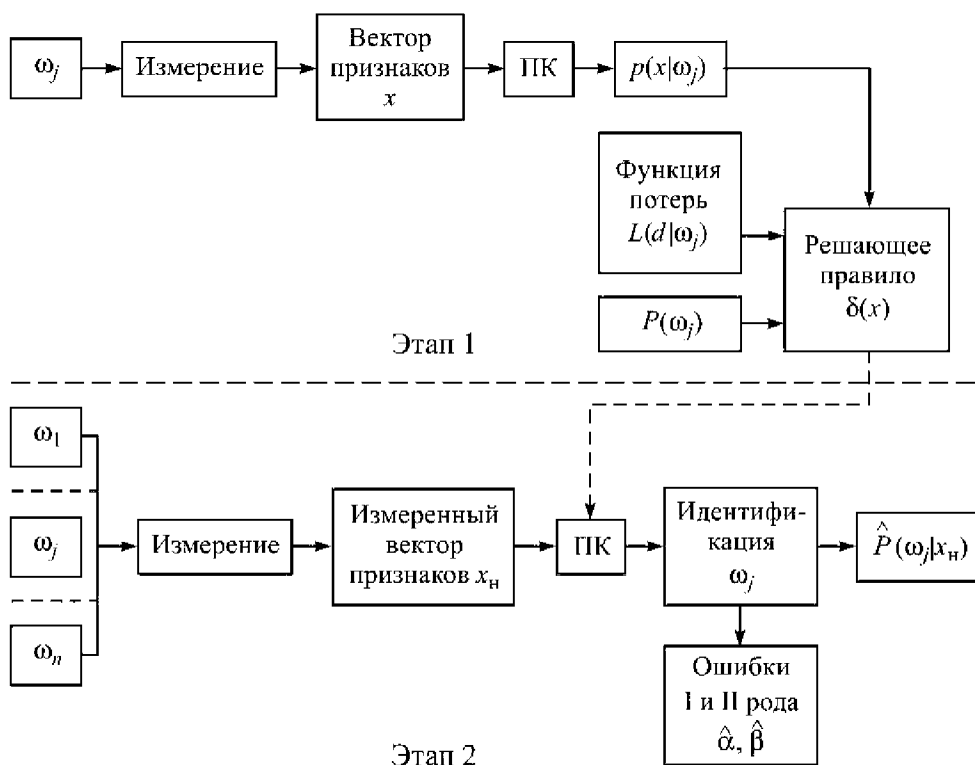


Рис. 6.7. Блок-схема традиционной процедуры обучения системы и распознавания образов

явления объекта ω_j . Классификация проводится путем минимизации вероятности ошибочного распознавания, в процессе которой получается решающее правило $\delta(x)$.

В традиционных методах на втором этапе условные распределения $p(x|\omega_j)$, $j = 1, 2, \dots, m$, являются детерминированными функциями, а случайная величина (вектор) x — детерминированной величиной (вектором). В силу этого получаемые разделяющие функции также являются детерминированными. В действительности следует оценить «истинные значения» признаков ξ и функций $p(\xi|\omega_j)$, $j = 1, 2, \dots, m$, которые в процессе обработки результатов наблюдений x_1, x_2, \dots, x_n должны иметь интервальные оценки, а если вид функций $p(\xi|\omega_j)$ был выбран априори, то должны быть указаны интервальные оценки их параметров и самих функций. Соответственно, должны быть представлены интервальные оценки разделяющих функций и наблюдавшейся перед классификацией случайной величины x .

Задача классификации объектов с учетом интервальных оценок исходных данных усложняется, но уменьшается число ложных распознаваний. Учет интервальных оценок разделяющих функций и наблюдавшихся перед классификацией величин чрезвычайно важен в практических приложениях, когда координаты вектора признаков достаточно близки для различных объектов, а точность измерения этих признаков соизмерима с разностью значений признаков. Например, при радиолокационных измерениях характеристик различных типов самолетов точность измерения соизмерима с разностью значений этих характеристик.

В общем случае байесовский риск (который определяет разделяющую функцию) относительно мало чувствителен к ошибке (приращению) в выборе значения априорного распределения $P(\omega_j)$, $j = 1, 2, \dots, m$. Если функция байесовского риска кусочно линейна, то приращение байесовского риска равно нулю, когда приращение распределения $P(\omega_i)$ содержится в интервале линейности функции байесовского риска. Поэтому далее априорное распределение $P(\omega_i)$ мы часто рассматривать не будем.

В действительности измеренные значения вектора признаков имеют две компоненты: истинное значение ξ и помеху δ , т. е. $x = \xi + \delta$. Неучет помехи δ приводит к смещенным оценкам всех характеристик, в расчете которых участвует значение x . Кроме этого, необходимо учесть интервальные оценки всех рассчитываемых функций и величин: $p(\xi|\omega_j)$ и ошибок первого и второго рода (в традиционных методах этого не делают), что приводит к появлению зон неопределенности в принятии решений даже при фиксированном объеме выборки. Наблюдаемые значения признаков x_n в процедуре идентификации также содержат помеху, и идентификацию объектов следует проводить по оценке истинного значения ξ_n , а не по наблюдаемому значению x_i . Определить оценку величины ξ_n можно, используя только всю информацию, полученную на первом этапе. Таким образом задача идентификации практически не разделяется на независимые этапы, а имеет общую схему (рис. 6.8).

На рис. 6.7 и 6.8 прямоугольниками с надписью ПК обозначены процедура обработки результатов наблюдений и ее алгоритм; на рис. 6.8 утолщенной обводкой прямоугольников обозначены блоки, которых либо нет в традиционных методах, либо эти блоки претерпевают существенные изменения по сравнению с традиционным подходом.

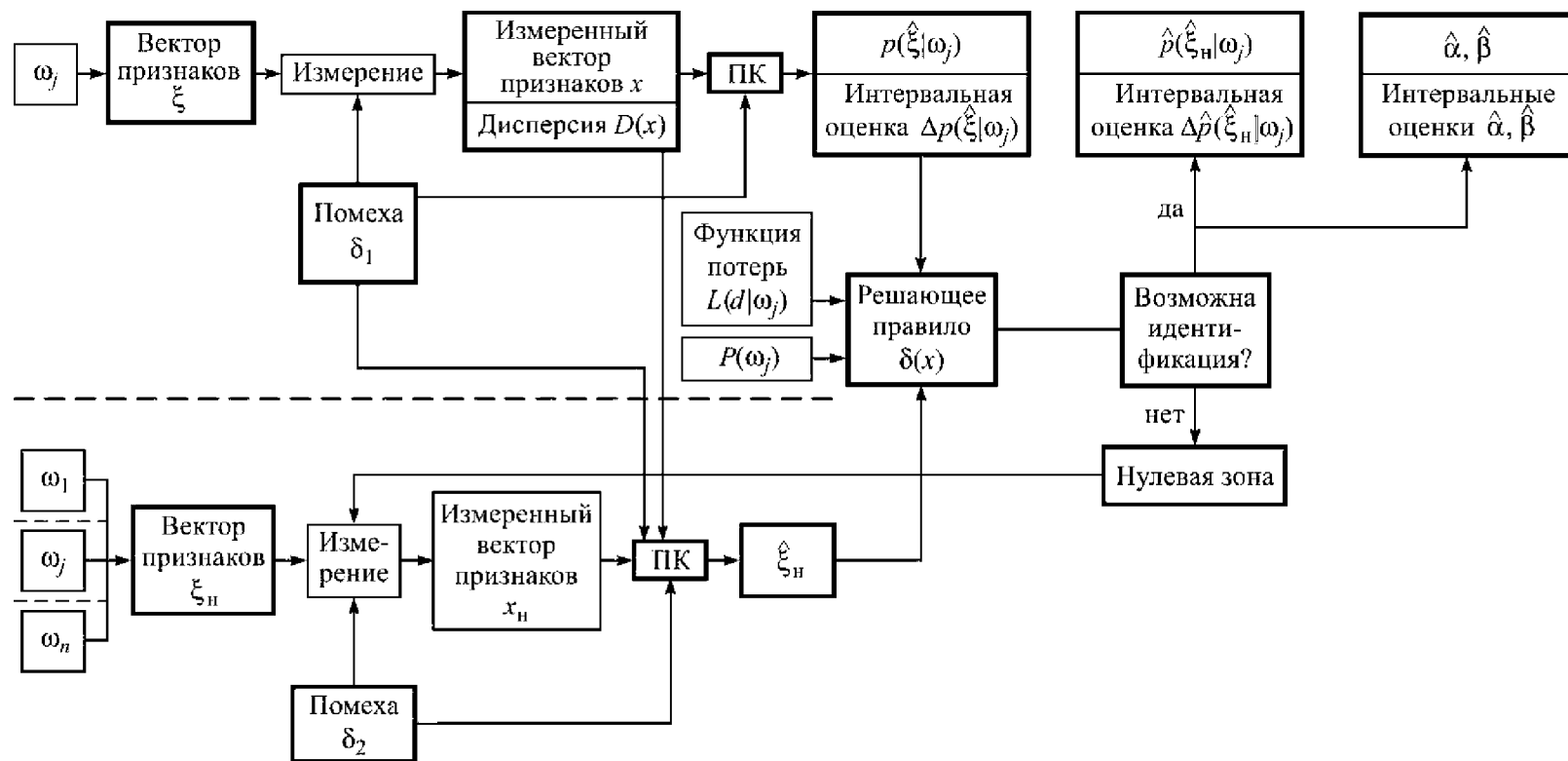


Рис. 6.8. Блок-схема единой процедуры обучения системы и распознавания образов

Из алгоритмов распознавания образов, представленных в блок-схеме, приведенной на рис. 6.8, видно, что изменяются конкретные значения оцениваемых функций и параметров по сравнению с традиционными алгоритмами (а это приводит и к иным практическим выводам) и появляется область неопределенности, когда идентификацию объектов, в принципе, произвести невозможно.

Ниже будет показано, что в процессах принятия решений даже при фиксированном объеме выборки появляются зоны неопределенности, в которых необходимо применять процедуры методов последовательного принятия решений. Естественно, что учет погрешности наблюдаемых значений признаков приведет к изменению верхних и нижних останавливающих границ и среднего числа наблюдений до принятия решений и в последовательных методах принятия решений.

Отсюда приходим к математической постановке задачи.

Введем некоторые обозначения. Каждый объект в задаче распознавания образов характеризуется вектором признаков ξ . В эксперименте из-за помех вместо величины ξ наблюдается случайная величина $x = \xi + \delta$. Для величины x может быть задана как плотность распределения вероятности признаков (для непрерывных случайных величин), так и функция вероятности (для дискретных случайных величин).

Пусть до принятия решения d , принадлежащего множеству решений D , задано параметрическое пространство исходов Ω и для всех $\omega \in \Omega$ задана обобщенная плотность распределения вероятности $p(\omega)$ (где Ω — пространство образов ω или пространство параметров ω в статических задачах решения). Задана вещественная функция полезности $u(\omega, d)$ или функция потерь $L(\omega, d) = -u(\omega, d)$ на произведении $\Omega \times D$ пространств.

Пусть S — выборочное пространство возможных значений наблюдаемого признака ξ исхода ω , когда значение признака не наблюдается, а наблюдается случайная величина x , связанная со значением признака ξ соотношением $x = \xi + \delta$, где δ — аддитивная помеха, плотность распределения $f(x)$ которой и числовые характеристики плотности распределения предполагают известными, в частности это может быть нормальный закон распределения с математическим ожиданием $M(\delta) = 0$ и дисперсионной матрицей $D(\delta) = \sigma^2 I$, где I — единичная матрица.

Пусть задано с точностью до значений параметров параметрическое семейство условных обобщенных вероятностных плотностей (о. в. п.) $p(\xi, \theta | \omega)$, $\omega \in \Omega$. При неизвестных о. в. п. $p(\xi, \theta | \omega)$ должно быть задано параметрическое семейство разделяющих функций $h(\xi, \theta)$.

Пусть выбрана из класса решающих функций Δ функция $\varphi(\xi)$, определяющая для любого возможного значения $\xi \in S$ решение $d(\xi) \in D$ и минимизирующая функцию среднего риска

$$\begin{aligned} \rho(\omega, d(\xi)) &= M\{L(\omega, d(\xi))\} = \\ &= \int_{\Omega} \int_S L(\omega, d(\xi)) p(\xi, \theta | \omega) p(\omega) d\mu(\xi) d\nu(\omega), \end{aligned} \quad (6.24)$$

где для всех $\omega \in \Omega$ функция $L(\omega, d(\xi))$ измерима и интегрируема на множестве S .

Требуется по выборке фиксированного объема результатов наблюдений x или в методах последовательного принятия решений найти:

1) точечные и интервальные оценки параметров θ условных о. в. п. $p(\xi, \theta | \omega)$, а также точечные и интервальные оценки самих условных о. в. п. $p(\xi, \theta | \omega)$; при неизвестных о. в. п. $p(\xi, \theta | \omega)$ — точечные и интервальные оценки параметров θ и разделяющих функций $h(\xi, \theta)$;

2) точечные и интервальные оценки решающих функций $\varphi(\xi)$;

3) оценки $\hat{\xi}$ «истинных» значений вектора признаков ξ , по которым впоследствии принимается решение $d \in D$.

После этого необходимо произвести идентификацию наблюдаемого объекта, т. е. поставить в соответствие наблюдаемому значению вектора признаков x некоторый образ ω_j и определить при этом точечные и интервальные оценки суммарной ошибки и ошибки первого и второго рода, возникающих при принятии решения $d \in D$.

В соотношении (6.24) запись выражений $d\mu(\xi)$ и $d\nu(\xi)$ означает, что интеграл $\rho(\omega, d(\xi))$ понимается как обычный интеграл для непрерывных случайных величин ξ и ω и как сумма для дискретных величин. При любом наборе $(\omega, d) \in \Omega \times D$ функция $L(\omega, d)$ представляет собой потери для лица, принимающего решение d при исходе ω .

В отличие от этой постановки в традиционной постановке задачи распознавания образов не учитывают помехи δ , величины ξ и x отождествляют, не определяют интервальные оценки функций $p(\xi, \theta | \omega)$, $h(\xi, \theta)$ и $\varphi(\xi)$, не определяют оценки $\hat{\xi}$ и интервальные оценки ошибок первого и второго рода (α и β). Отсутствие учета помех δ и оценок $\hat{\xi}$ приводит к смещенным точечным оценкам $p(\xi, \theta | \omega)$, $h(\xi, \theta)$ и $\varphi(\xi)$, к неверным интервальным оценкам и к ложным практическим выводам.

МЕТОДЫ РЕГРЕССИОННОГО И КОНФЛЮЭНТНОГО АНАЛИЗА КАК ИНСТРУМЕНТ В ПРОЦЕДУРАХ ПРИНЯТИЯ РЕШЕНИЙ

§ 7.1. Понятие регрессии. Основные определения

При определении вида функции плотности распределения вероятности распознаваемых классов или разделяющих (решающих) функций в качестве исходных данных используют выборки, полученные в результате конкретных экспериментов. Как любые экспериментальные значения, исходные данные содержат ошибки, которые существенно влияют на результаты решений задач. Поскольку рассматриваемые алгоритмы учета погрешностей исходных данных будут применяться к различным функциям: условным плотностям вероятности, разделяющим (решающим) функциям, то измеренные значения признаков обозначим переменной x , а значения оцениваемых функций — $y(x)$. Будем считать, что общий вид оцениваемых функций априори известен, но надлежит найти точечные и интервальные оценки свободных параметров этих функций, по которым можно определить точечные и интервальные оценки самих функций.

Модели, позволяющие учитывать ошибки в значениях функций и аргументов, рассматриваются в [8, 9, 16, 20–24, 30, 40, 87, 94–96, 99, 100, 102–104, 106, 108, 109]. Наиболее часто [2, 14, 45, 76, 80, 93] встречается постановка задачи определения оценок вектора параметров θ модели

$$y_i = f(x_i, \theta) + \varepsilon_i, \quad i = 1, 2, \dots, n,$$

где ε_i — случайная ошибка, имеющая нормальное распределение с параметрами $M(\varepsilon_i) = 0$, $D(\varepsilon_i) = \sigma_i^2$, $D(\varepsilon_i, \varepsilon_j) = 0$, $i, j = 1, 2, \dots, n$. Данная постановка является классической регрессионной задачей, которая решается *методом максимума правдоподобия* (ММП) или *методом наименьших квадратов* (МНК).

Если дисперсии наблюдаемых значений равны: $\sigma_i^2 = \sigma^2$, $i = 1, \dots, n$, и значение σ^2 не задано, то оценку параметра σ можно найти по формуле

$$\hat{\sigma}^2 = \frac{S}{n - p},$$

где $S = \sum_{i=1}^n (y_i - f(x_i, \hat{\theta}))^2$, p — число оцениваемых параметров, n — число точек наблюдений.

При решении регрессионных задач предполагают, что переменные x являются детерминированными, т. е. измеренными с погрешностью много меньшей, чем погрешность значений функции. На практике это требование часто не выполняется, поэтому возникает необходимость учета погрешностей аргумента x . Если погрешности аргумента не учитываются, то оценки искомых параметров будут *смещенными*.

Рассмотрим пассивный эксперимент и алгоритм определения оценок параметров функции $\eta = \psi(\xi, \theta)$, где ξ — аргумент функции, θ — неизвестный вектор параметров. В процессе измерений получают не значения функции η и аргумента ξ , а наборы значений $\{y_i\}$ и $\{x_i\}$:

$$y_i = \eta_i + \varepsilon_i, \quad x_i = \xi_i + \delta_i, \quad i = 1, 2, \dots, n,$$

где ε_i и δ_i — ошибки измеренных значений функции и аргумента.

Предположим, что ошибки измерений ε_i и δ_i — нормально распределенные случайные величины с нулевыми средними значениями, с дисперсиями $\sigma^2(y_i)$ и $\sigma^2(x_i)$ соответственно и коэффициентом корреляции $\rho_i = 0$.

Рассмотрим простейший случай. Пусть ξ , η — две случайные величины, которые имеют совместное распределение непрерывного типа с плотностью вероятности $f(x, y)$. В процессе наблюдений случайная величина ξ принимает значения x_i , а случайная величина η — значения y_i , $i = 1, 2, \dots, n$. Будем рассматривать ξ как независимую переменную. Тогда фиксированному значению $\xi = x$ соответствует распределение вероятностей зависимой переменной η с плотностью вероятности

$$f(y|x) = \frac{f(x,y)}{f_1(x)} = \frac{f(x,y)}{\int_{-\infty}^{+\infty} f(x,y)dy},$$

где $f_1(x)$ — плотность вероятности частного распределения случайной величины ξ .

Можно найти числовые характеристики условного распределения вероятностей переменной η : среднее значение, моду, медиану и другие, которые будут зависеть от x . Обозначим через \bar{y}_x выбранную числовую характеристику. Если x изменяется, то точки (x, \bar{y}_x) опишут некоторую кривую (рис. 7.1). По виду этой кривой можно судить о поведении условного η -распределения для различных значений ξ . Такая кривая называется *кривой регрессии* (говорят, что она описывает *регрессию η на ξ*). Далее будем полагать, что \bar{y}_x — условное среднее величины η , задаваемое соотношением

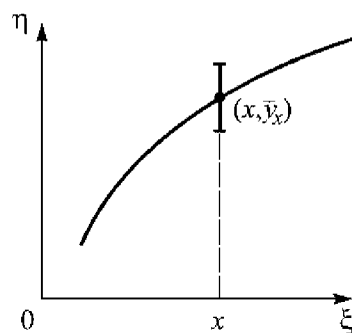


Рис. 7.1. Регрессия η на ξ

$$\bar{y}_x = M(\eta|\xi = x) = \frac{\int_{-\infty}^{+\infty} y f(x,y) dy}{\int_{-\infty}^{+\infty} f(x,y) dy}.$$

Получим кривую регрессии для условного среднего значения η .

Если рассматривать η как независимую переменную, то условная плотность вероятности зависимой переменной ξ при фиксированном значении $\eta = y$ определяется по формуле

$$f(x|y) = \frac{f(x,y)}{f_2(y)} = \frac{f(x,y)}{\int_{-\infty}^{+\infty} f(x,y)dx} = \frac{f_1(x)f(y|x)}{f_2(y)},$$

где $f_2(y)$ — плотность вероятности частного распределения случайной величины η .

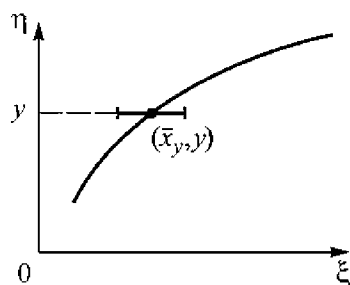


Рис. 7.2. Регрессия ξ на η

Значения любой числовой характеристики \bar{x}_y условного распределения величины ξ описывают кривую *регрессии ξ на η* (рис. 7.2). Кривая регрессии для условного среднего значения ξ задается уравнением

$$\bar{x}_y = M(\xi | \eta = y).$$

Две кривые регрессии \bar{y}_x и \bar{x}_y в общем случае не будут совпадать. Кривые регрессии выбирают таким образом, чтобы они отвечали свойству минимальности: среди множества всех функций $g(\xi)$ необходимо найти такую, которая даст возможно лучшее представление о случайной величине η . Например, если термином «возможно лучшее представление» определить минимум выражения

$$\begin{aligned} M\{[\eta - g(\xi)]^2\} &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [y - g(x)]^2 f(x, y) dx dy = \\ &= \int_{-\infty}^{+\infty} f_1(x) dx \int_{-\infty}^{+\infty} [y - g(x)]^2 f(y | x) dy, \end{aligned} \quad (7.1)$$

то среди всех возможных значений $g(\xi)$ необходимо взять $g(\xi) = \bar{y}_x$. Аналогично, выражение $M\{[\xi - h(\eta)]^2\}$ достигает минимума при $h(\eta) = \bar{x}_y$.

На практике наиболее часто применяется другая постановка задачи регрессии. Рассмотрим функцию $g(\xi)$ или $h(\eta)$, принадлежащую конкретному классу (например, классу линейных функций), и будем путем подбора значений свободных параметров этой функции искать такую, которая даст возможно лучшее представление величины ξ или η . В общем случае это будет другая кривая регрессии.

Наиболее простой является линейная регрессия $\eta = g(\xi) = \theta_1 + \theta_2 \xi$. Подставим это выражение $g(\xi)$ в (7.1) вместо $g(x)$. Если оценки параметров θ_1 и θ_2 находятся из условия (7.1), то говорят о *линейной средней квадратической регрессии*.

Рассмотрим различные определения возможно лучшего представления зависимой переменной. Для этого используется концепция метода наименьших квадратов. Здесь можно отметить два подхода. Из курса линейной алгебры известно, что представление

некоторого элемента (функции) y , принадлежащего пространству L_2 (пространству функций с интегрируемым квадратом), в виде линейной комбинации заданных элементов (функций) $\phi_i(x)$ сводится к нахождению минимума по α_i , $i=1, 2, \dots, n$, следующего выражения:

$$F = \int_X \left(y - \sum_{i=1}^m \alpha_i \phi_i(x) \right)^2 dx, \quad (7.2)$$

где X — множество допустимых значений x , α_i — коэффициенты разложения.

Получим этот же функционал вторым способом. Пусть случайная величина η имеет нормальное распределение, т. е. $\eta \in N\left(\sum_i \alpha_i \phi_i(x); \sigma^2\right)$. Предположим, что дисперсия σ^2 известна и постоянна для любого x . Рассмотрим независимые наблюдения x_j , $j=1, 2, \dots, n$, каждому из которых отвечает значение y_j . Функция правдоподобия для этих наблюдений имеет вид

$$L(y, \alpha) = \frac{1}{(2\pi)^{n/2} \sigma^n} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{j=1}^n \left(y_j - \sum_{i=1}^m \alpha_i \phi_i(x_j) \right)^2 \right\}$$

и будет максимальна, если достигается минимум по α_i , $i=1, 2, \dots, m$, выражения

$$F = \frac{1}{2\sigma^2} \sum_{j=1}^n \left(y_j - \sum_{i=1}^m \alpha_i \phi_i(x_j) \right)^2, \quad (7.3)$$

эквивалентного выражению (7.2).

Используя статистический функционал (7.3), можем не только получить оценки коэффициентов α_i , но и определить закон их распределения и интервальные оценки.

Заметим, что использование функции правдоподобия приводит к методу наименьших квадратов только в случае нормального закона распределения случайной величины (зависимой переменной). Например, если случайная величина η подчиняется закону распределения Лапласа

$$\psi(x) = \frac{1}{2\beta} \exp \left\{ -\frac{1}{\beta} \left| y - \sum_i \alpha_i \phi_i(x) \right| \right\},$$

то функция правдоподобия для независимых наблюдений имеет вид

$$L(y, \alpha) = \frac{1}{2^n \beta^n} \prod_{j=1}^n \exp \left\{ -\frac{1}{\beta} \left| y_j - \sum_i \alpha_i \varphi_i(x_j) \right| \right\},$$

а функционал, аналогичный функционалу (7.3), — вид

$$F = \frac{1}{\beta} \sum_{j=1}^n \left| y_j - \sum_i \alpha_i \varphi_i(x_j) \right|. \quad (7.4)$$

Из анализа функционалов (7.3) и (7.4) следует, что то или иное определение «возможно лучшего представления» зависит от вида законов распределения исходных данных.

Здесь мы рассмотрели функционалы для регрессии η на ξ . Аналогично получаются выражения для функционалов, определяющих регрессию ξ на η . Заметим, что в обоих случаях независимые переменные принимают фиксированные (детерминированные, неслучайные) значения. Случайной величиной остается только значение зависимой переменной.

§ 7.2. Линейные регрессии η на ξ и ξ на η

Пусть уравнение регрессии η на ξ имеет вид

$$\bar{y}_x = \theta_1 + \theta_2 x.$$

Получим уравнения линий регрессии в этом широко распространенном случае. Используя определение линейной средней квадратической регрессии, найдем оценки параметров θ_1 и θ_2 и выразим уравнение регрессии через данные наблюдений. Для этого продифференцируем функционал

$$F = \sum_i (\bar{y}_{x_i} - \theta_1 - \theta_2 x_i)^2 \quad (7.5)$$

по θ_1 и θ_2 и приравняем производные нулю. Отсюда получим оценки

$$\hat{\theta}_2 = \rho \frac{\sigma_y}{\sigma_x}, \quad \hat{\theta}_1 = \bar{y} - \hat{\theta}_2 \bar{x},$$

где $\hat{\theta}_2$ — коэффициент регрессии η на ξ ; \bar{x} , \bar{y} — соответственно средние значения наборов $\{x_i\}$ и $\{y_i\}$; ρ — коэффициент корреляции случайных величин ξ и η :

$$\rho = \frac{1}{n\sigma_x\sigma_y} \left(\sum_i \sum_j x_i y_j - n\bar{x}\bar{y} \right),$$

где n — число точек (x_i, y_i) ; σ_x и σ_y — соответственно средние квадратические отклонения вариационных рядов x и y (не путать с величиной σ^2 — дисперсией нормального закона распределения случайной величины η). Отсюда уравнение средней квадратической регрессии η на ξ имеет вид

$$\frac{\bar{y}_x - \bar{y}}{\sigma_y} = \rho \frac{x - \bar{x}}{\sigma_x}. \quad (7.6)$$

Подставив найденные оценки $\hat{\theta}_1$ и $\hat{\theta}_2$ в соотношение (7.5), найдем минимальное значение функционала F :

$$F_{\min} = n\sigma_y^2(1 - \rho^2).$$

Выражение (7.5) определяет сумму квадратов расстояний по вертикали (вдоль оси η) между точками (x_i, \bar{y}_{x_i}) и прямой $\bar{y}_x = \theta_1 + \theta_2 x$.

В случае регрессии ξ на η , т. е. $\bar{x}_y = \alpha + \beta\eta$, необходимо найти оценки параметров α и β , при которых можно решить задачу

$$M[(\xi - \alpha - \beta\eta)^2] \rightarrow \min_{\alpha, \beta},$$

или задачу

$$\Phi = \sum_i (x_{y_i} - \alpha - \beta y_i)^2 \rightarrow \min_{\alpha, \beta},$$

где значения y_i фиксированы. Здесь расстояние между точками (y_i, \bar{x}_{y_i}) и прямой $\bar{x}_y = \alpha + \beta y$ измеряется по оси ξ .

Уравнение средней квадратической регрессии ξ на η имеет вид

$$\frac{\bar{x}_y - \bar{x}}{\sigma_x} = \rho \frac{y - \bar{y}}{\sigma_y}. \quad (7.7)$$

В данном случае коэффициент регрессии и минимальное значение функционала определяются по формулам

$$\hat{\beta} = \rho \frac{\sigma_x}{\sigma_y}, \quad \Phi_{\min} = n\sigma_x^2(1 - \rho^2).$$

§ 7.3. Регрессионный парадокс

По одним и тем же исходным данным (если не учитывать строго их погрешности) согласно уравнениям (7.6) и (7.7) мы получим различные прямые, а следовательно, и различные выводы из решенной задачи. Прямые регрессии совпадают только при $\rho = \pm 1$; при $\rho = 0$ (когда ξ и η — независимые случайные величины) имеем соответственно уравнения

$$\bar{y}_x = \bar{y}, \quad \bar{x}_y = \bar{x}$$

двух взаимно перпендикулярных прямых.

Если $\bar{x} = \bar{y} = 0$, $\sigma_x = \sigma_y = 1$, то линия регрессии η на ξ описывается уравнением $\bar{y}_x = \rho x$; а линия регрессии ξ на η — уравнением $y = \bar{x}_y / \rho$.

В качестве примера рассмотрим линии регрессии, характеризующие зависимость роста сыновей от роста отцов. Пусть переменная ξ характеризует рост отца, а переменная η — рост сына (рис. 7.3). Уравнение регрессии η на ξ имеет вид

$$\frac{\bar{y}_x - \bar{y}}{\sigma_y} = \rho \frac{x - \bar{x}}{\sigma_x}, \quad |\rho| \leq 1.$$

Положим $\sigma_y = \sigma_x = 1$ и перенесем начало координат в точку (\bar{x}, \bar{y}) , т. е. в уравнениях регрессии положим $\bar{x} = \bar{y} = 0$. Получим

$$\bar{y}_x = \rho x \leq x.$$

Это означает, что в среднем сыновья высоких отцов не так высоки, как их отцы. Рост сыновей имеет тенденцию к усреднению.

Уравнение регрессии ξ на η имеет вид

$$\frac{\bar{x}_y - \bar{x}}{\sigma_x} = \rho \frac{y - \bar{y}}{\sigma_y},$$

и при сделанных предположениях имеем $\bar{x}_y = \rho y \leq y$.

Это означает, что в среднем отцы высоких детей не так высоки, как их дети. Рост поколения отцов имеет тенденцию к усреднению. Причем оба утверждения имеют место одновременно, что

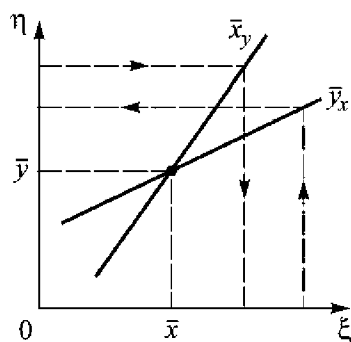


Рис. 7.3. Регрессионный парадокс

является регрессионным парадоксом. Чтобы устранить этот парадокс, необходимо при определении уравнения регрессии учесть случайный характер как переменной ξ , так и переменной η .

§ 7.4. Ортогональная регрессия

Учесть одновременно случайный характер переменных ξ и η позволяет ортогональная регрессия. Пусть известны набор случайных точек $\{x_i, y_i\}$, $i=1, 2, \dots, n$, и закон распределения каждого наблюдения. Область неопределенности каждой точки (x_i, y_i) теперь, в отличие от регрессий η на ξ и ξ на η , является некоторой окрестностью (рис. 7.4), форма которой зависит от закона распределения результатов наблюдений. Линия регрессии должна возможно «лучшим» образом пройти по областям неопределенности.

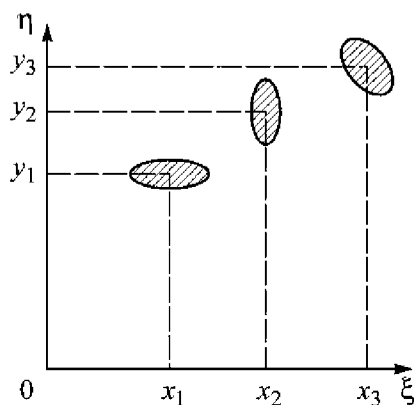


Рис. 7.4. Области неопределенности

Пусть результаты наблюдений $x_i = \xi_i + \delta_i$ и $y_i = \eta_i + \varepsilon_i$, $\eta = f(\xi, \theta)$, подчиняются нормальному закону распределения соответственно с математическими ожиданиями ξ_i и η_i , дисперсиями $\sigma^2(x_i)$ и $\sigma^2(y_i)$ и коэффициентами корреляции ρ_i . Тогда функция правдоподобия имеет вид

$$L(x, y; \theta) = \prod_{i=1}^n \frac{1}{2\pi\sigma(x_i)\sigma(y_i)\sqrt{1-\rho_i^2}} \times \exp\left\{-\frac{1}{2(1-\rho_i^2)}\left(\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} - 2\rho_i \frac{(x_i - \xi_i)(y_i - \eta_i)}{\sigma(x_i)\sigma(y_i)} + \frac{(y_i - \eta_i)^2}{\sigma^2(y_i)}\right)\right\}, \quad (7.8)$$

а соответствующий минимизируемый функционал — вид

$$F = \frac{1}{2} \sum_{i=1}^n \frac{1}{1-\rho_i^2} \left[\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} - 2\rho_i \frac{(x_i - \xi_i)(y_i - \eta_i)}{\sigma(x_i)\sigma(y_i)} + \frac{(y_i - \eta_i)^2}{\sigma^2(y_i)} \right]. \quad (7.9)$$

Для функционала (7.9) при $\rho = 0$ найдем точку минимума по θ :

$$F = \frac{1}{2} \sum_{i=1}^n \left[\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} + \frac{(y_i - \eta_i)^2}{\sigma^2(y_i)} \right] \rightarrow \min_{\theta}. \quad (7.10)$$

Геометрически (при $\sigma(x_i) = \sigma(y_i) = 1$) это означает, что минимизируется сумма квадратов расстояний между точками (x_i, y_i) и соответствующими точками (ξ_i, η_i) кривой регрессии.

Чтобы найти минимум функционала F по θ , продифференцируем его по θ_j , $j = 1, 2, \dots, m$. Получим систему уравнений

$$\frac{\partial F}{\partial \theta_j} = \sum_{i=1}^n \frac{y_i - \eta_i}{\sigma^2(y_i)} \frac{\partial \eta_i}{\partial \theta_j} \bigg|_{\theta_j = \hat{\theta}_j} = 0, \quad j = 1, 2, \dots, m, \quad (7.11)$$

которую решить невозможно, поскольку не известны истинные значения ξ_i , входящие в выражения для η_i , т. е. нельзя найти оценки параметров θ_j .

Доопределим условие задачи (7.10). В качестве оценок $\hat{\xi}_i$ возьмем те значения ξ_i , которые обратят в нуль частные производные $\frac{\partial F}{\partial \xi_i}$:

$$\frac{\partial F}{\partial \xi_i} = \frac{x_i - \xi_i}{\sigma^2(x_i)} + \frac{y_i - \eta_i}{\sigma^2(y_i)} \frac{\partial \eta_i}{\partial \xi_i} = 0, \quad i = 1, 2, \dots, n. \quad (7.12)$$

Условие (7.12) при $\sigma(x_i) = \sigma(y_i) = 1$ является условием ортогональности векторов $\{(x_i - \hat{\xi}_i); (y_i - \hat{\eta}_i)\}$ и $\{1; \partial \eta_i / \partial \xi_i\}$, причем последний вектор направлен по касательной к кривой $\eta = f(\xi, \theta)$ или вдоль прямой при линейной регрессии (рис. 7.5). Условия (7.11), (7.12) определяют третью линию регрессии — ортогональную, не совпадающую ни с кривой регрессии η на ξ ни с кривой регрессии ξ на η , но одновременно учитывают случайный характер ξ , и η .

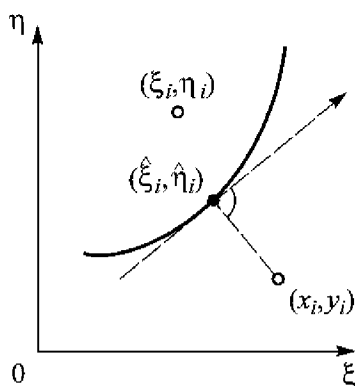


Рис. 7.5. Наблюдаемые и оцениваемые координаты переменных

Приведенные результаты справедливы для пассивного эксперимента, когда используются текущие значения x_i и соответствующие им значения y_i .

К результатам активного эксперимента, когда заранее устанавливаются значения x_i , в общем случае выводы пассивного эксперимента не применимы.

§ 7.5. Метод наименьших квадратов. Оценка свободных параметров функций, линейных по параметрам

Примером функций, линейных по параметрам θ , могут быть функции

$$\eta = \sum_{i=0}^k \theta_i x^i, \quad \eta = \sum_{i=0}^k \theta_i f_i(x)$$

и т. д. Их часто используют для описания разделяющих функций, для приближенного описания законов распределения и т. п.

Здесь независимая переменная x измеряется без ошибок (детерминирована), т. е. рассматривается регрессия y на x . Результаты наблюдений случайной величины η дают значения y , соответствующие значениям x . Каждому значению x_i может отвечать либо единственное значение y_i , либо множество значений $y_{i1}, y_{i2}, \dots, y_{im}$.

Предположим, что каждое наблюдение y_i содержит ошибку ε_i .

Пусть n -мерный вектор наблюдений (вектор откликов) $y = (y_1, y_2, \dots, y_n)^T$ порождается моделью

$$y = X\theta + \varepsilon,$$

где X — известная $(n \times p)$ -матрица измерений (матрица плана), построенная либо по значениям вектора x , либо по значениям функций $f_i(x)$, $i = 1, 2, \dots, k$; $\theta = (\theta_1, \theta_2, \dots, \theta_p)^T$ — неизвестный p -мерный вектор параметров; $(\varepsilon - n)$ -мерный случайный вектор ошибок, удовлетворяющий условиям $M(\varepsilon) = 0$, $D(\varepsilon) = D(\varepsilon^T \varepsilon) = \sigma^2 I$. Здесь σ^2 — неизвестный скалярный параметр, I — единичная матрица. Линейные модели наблюдений могут быть полного и неполного ранга. Если ранг матрицы измерений, $\text{rang } X$, равен p , то модель наблюдений называется *моделью полного ранга*, если $\text{rang } X < p$, то — *моделью неполного ранга*. Другими словами, если число точек наблюдений меньше числа искомых параметров θ , то имеет место модель наблюдений

неполного ранга. В этом случае некоторые параметры θ являются свободными. Модель неполного ранга здесь не рассматривается.

А. Простейший случай: $\sigma^2 = 1$. Используя метод наименьших квадратов, найдем точку минимума квадратичной формы

$$Q(\theta) = \varepsilon^T \varepsilon = (y - X\theta)^T (y - X\theta) = y^T y - 2\theta^T X^T y + \theta^T X^T X \theta.$$

Дифференцируя $Q(\theta)$ по θ , получим систему нормальных уравнений

$$2X^T y + 2X^T X \theta = 0.$$

Отсюда имеем

$$\theta = (X^T X)^{-1} X^T y.$$

Минимальное значение S квадратичной формы $Q(\theta)$ при $\theta = \hat{\theta}$ вычислим по формуле

$$\begin{aligned} S &= y^T y - 2[(X^T X)^{-1}(X^T y)]X^T y + \\ &+ [(X^T X)^{-1}(X^T y)]^T X^T X (X^T X)^{-1} X^T y = \\ &= y^T y - 2y^T X [(X^T X)^{-1}(X^T y)]^T + y^T X (X^T X)^{-1} X^T X (X^T X)^{-1} X^T y = \\ &= y^T y - y^T X (X^T X)^{-1} X^T y, \end{aligned}$$

откуда получим

$$S = y^T y - y^T X \hat{\theta}.$$

Здесь использованы соотношения $(AB)^T = B^T A^T$, $(AB)^{-1} = B^{-1} A^{-1}$.

Итак, при нахождении точечной оценки параметров методом наименьших квадратов не требуется предположения о нормальном законе распределения исходных данных. Однако это предположение необходимо при построении доверительных интервалов и для проверки гипотез о параметрах θ .

Рассмотрим свойства оценок, полученных методом наименьших квадратов.

Теорема. Оценка $\hat{\theta}$ является несмещенной оценкой параметра θ .

Доказательство. Найдем математическое ожидание оценок $\hat{\theta}$:

$$M(\hat{\theta}) = M[(X^T X)^{-1} X^T y] = (X^T X)^{-1} X^T M(y).$$

Поскольку $M(y) = X\theta$, то $M(\hat{\theta}) = (X^T X)^{-1} X^T X\theta = \theta$. Теорема доказана.

Теорема (Гаусса — Маркова). Среди класса оценок θ^* величины θ , которые являются несмещенными оценками и представляют собой линейные комбинации исходных данных y , с помощью метода наименьших квадратов можно найти такую оценку $\hat{\theta}$, что $D(\hat{\theta}) \leq D(\theta^*)$, т. е. $\hat{\theta}$ — наиболее точная оценка величины θ из всех возможных, принадлежащих данному классу (эффективная оценка).

Доказательство теоремы можно найти в [40, 49, 89].

Получим формулу для $D(\hat{\theta})$. Пусть дисперсия $D(\varepsilon_j) = \sigma^2$, $j = 1, 2, \dots, n$, т. е. $D(y) = \sigma^2 I$. В общем случае имеет место равенство $D(Wy) = WD(y)W^T$. При $\hat{\theta} = (X^T X)^{-1} X^T y$ имеем

$$\begin{aligned} D(\hat{\theta}) &= (X^T X)^{-1} X^T D(y) [(X^T X)^{-1} X^T]^T = \\ &= (X^T X)^{-1} X^T \sigma^2 I X (X^T X)^{-1} = \\ &= \sigma^2 (X^T X)^{-1} X^T I X (X^T X)^{-1} = \sigma^2 (X^T X)^{-1}. \end{aligned}$$

Из формулы $D(\hat{\theta}) = \sigma^2 (X^T X)^{-1}$ следует, что дисперсии оценок могут быть получены без знания самих оценок, причем оценка неизвестного параметра σ^2 имеет вид

$$\hat{\sigma}^2 = \frac{S}{n - p},$$

где n — число точек наблюдения, p — число оцениваемых компонентов вектора θ .

Замечание. Приведенными формулами для вычисления дисперсии можно пользоваться, несколько их изменив, если каждое наблюдение имеет свой вес ω_j . Пусть $D(\varepsilon_j) = \sigma_j^2$; тогда можно положить $D(\varepsilon_j) = \sigma^2 / \omega_j$, где ω_j — вес j -го наблюдения.

Вместо разности $\varepsilon_j = y_j - \sum_{i=1}^p \hat{\theta}_i x_i$ в этом случае следует использовать величины $\varepsilon_j^* = \varepsilon_j \sqrt{\omega_j}$. Тогда

$$D(\epsilon_j^*) = D(\epsilon_j \sqrt{\omega_j}) = \omega_j D(\epsilon_j) = \omega_j \frac{\sigma^2}{\omega_j} = \sigma^2,$$

и минимальным значением функционала будет величина

$$S = \sum_{j=1}^n (\epsilon_j^*)^2 = \sum_{j=1}^n \omega_j \epsilon_j^2.$$

Б. Общий случай. Результаты наблюдений подчиняются модели $y = X\theta + \epsilon$. Ошибки ϵ имеют математическое ожидание $M(\epsilon) = 0$ и конечные вторые моменты

$$M(\epsilon_j^2) = \sigma_j^2 = D(y_j),$$

$$M(\epsilon_i, \epsilon_j) = \rho_{ij} \sigma_i \sigma_j = \text{cov}(y_i, y_j), \quad i, j = 1, 2, \dots, n,$$

т. е. ковариационную матрицу

$$D(y) = \begin{pmatrix} \sigma_1^2 & \sigma_1 \sigma_2 \rho_{12} & \dots & \sigma_1 \sigma_n \rho_{1n} \\ \sigma_2 \sigma_1 \rho_{21} & \sigma_2^2 & \dots & \sigma_2 \sigma_n \rho_{2n} \\ \sigma_n \sigma_1 \rho_{n1} & \sigma_n \sigma_2 \rho_{n2} & \dots & \sigma_n^2 \end{pmatrix}.$$

Рассмотрим квадратичную форму

$$\begin{aligned} Q(\theta) &= (y - X\theta)^T D^{-1}(y) (y - X\theta) = \\ &= y^T D^{-1}(y) y - 2y^T D^{-1}(y) X\theta + (X\theta)^T D^{-1}(y) X\theta = \\ &= y^T D^{-1}(y) y - 2\theta^T X^T D^{-1}(y) y + \theta^T X^T D^{-1}(y) X\theta. \end{aligned}$$

Дифференцируя $Q(\theta)$ по θ , получим систему нормальных уравнений

$$2X^T D^{-1}(y) y = 2X^T D^{-1}(y) X\theta.$$

Отсюда имеем оценки

$$\hat{\theta} = (X^T D^{-1}(y) X)^{-1} X^T D^{-1}(y) y.$$

Оценки $\hat{\theta}$ являются также несмещенными и эффективными. Дисперсия оценок имеет вид

$$D(\hat{\theta}) = [X^T D^{-1}(y) X]^{-1}.$$

Предположим, что $D(y)$ можно представить в виде произведения σ^2 на некоторую матрицу N : $D(y) = \sigma^2 N$. Тогда

$$D^{-1}(y) = \frac{1}{\sigma^2} N^{-1} = \frac{W}{\sigma^2}, \quad D(\hat{\theta}) = \sigma^2 (X^T W X)^{-1},$$

где оценка σ^2 имеет вид

$$\hat{\sigma}^2 = \frac{\varepsilon^T W \varepsilon}{n - p}.$$

Пример. Найти оценки вектора параметров θ в модели $\eta = \theta_1 x_1 + \theta_2 x_2$, если заданы матрица плана

$$X = \begin{pmatrix} 3 & 2 & 3 \\ 1 & 1 & 2 \end{pmatrix}^T,$$

вектор наблюдений $y = (4, 4, 4)^T$ и ковариационная матрица для значений y :

$$D(y) = \sigma^2 N, \quad N = \begin{pmatrix} 4 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix}.$$

Решение. Найдем матрицу $D^{-1} = W/\sigma^2$, предварительно обратив матрицу N :

$$W = N^{-1} = \begin{pmatrix} 0,333 & -0,167 & 0,000 \\ -0,167 & 0,333 & 0,000 \\ 0,000 & 0,000 & 0,250 \end{pmatrix}.$$

Поскольку параметр σ^2 постоянен для всех наблюдений, то на точечную оценку $\hat{\theta}$ значение σ^2 влияния не оказывает. В результате получим

$$\hat{\theta} = (1,00; 0,75)^T, \quad \varepsilon = y - X\theta = (0,25; 1,25; -0,50)^T.$$

Тогда имеем

$$\hat{\sigma}^2 = \frac{\varepsilon^T W \varepsilon}{n - p} = \frac{0,500}{3 - 2} = 0,500,$$

$$D(\hat{\theta}) = \hat{\sigma}^2 (X^T W X)^{-1} = \begin{pmatrix} 1,000 & -1,750 \\ -1,750 & 3,437 \end{pmatrix}.$$

Отсюда получаем $\sigma(\hat{\theta}_1) = 1,000$, $\sigma(\hat{\theta}_2) = 1,854$.

Из вида полученной корреляционной матрицы оценок параметров θ

$$\rho_{ij} = \begin{pmatrix} 1,00 & -0,94 \\ -0,94 & 1,00 \end{pmatrix}$$

следует, что бессмысленно в данном примере приводить только средние квадратические отклонения оценок, так как велики недиагональные элементы (рис. 7.6). Большая положительная ошибка в оценке $\hat{\theta}_1$ приведет к большой отрицательной ошибке в оценке $\hat{\theta}_2$. На рис. 7.6 указаны маргинальные средние квадратические отклонения $\sigma(\hat{\theta}_1)$ и $\sigma(\hat{\theta}_2)$ и интервальные оценки $\hat{\theta}_1$ и $\hat{\theta}_2$, они определяются проекциями эллипса на соответствующие оси координат.

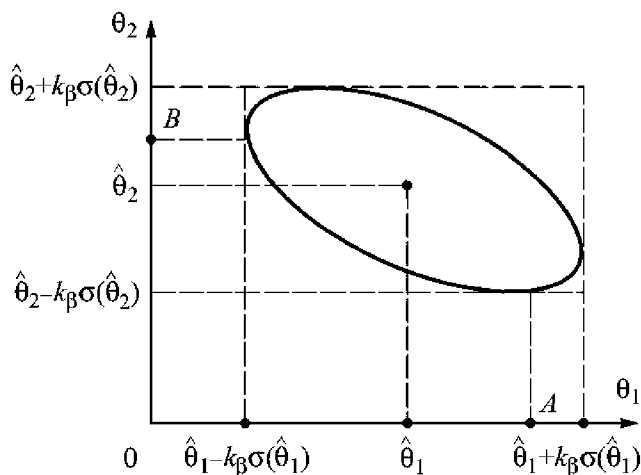


Рис. 7.6. Эллипс рассеяния оценок $\hat{\theta}$:
координата A равна $\hat{\theta}_1 + k_\beta \sigma(\hat{\theta}_1) \sqrt{1 - p^2}$,
координата B равна $\hat{\theta}_2 + k_\beta \sigma(\hat{\theta}_2) \sqrt{1 - p^2}$

В общем случае из условия нормального распределения вектора оценок $\hat{\theta}$ следует, что квадратичная форма

$$Q(\hat{\theta}, \theta) = (\hat{\theta} - \theta)^T D(\hat{\theta})(\hat{\theta} - \theta)$$

распределена по закону $\chi^2(N)$, где N — размерность вектора θ . Поэтому можно записать вероятностное утверждение:

$$P\{Q(\hat{\theta}, \theta) \leq k_{\beta}^2\} = \beta,$$

где β — доверительная вероятность, k_{β}^2 — квантиль для $\chi^2(N)$ -распределения.

Область в пространстве параметров θ задается уравнением $Q(\hat{\theta}, \theta) = k_{\beta}^2$ и имеет вид гиперэллипсоида. Отсюда можно получить доверительную область для вектора оценок $\hat{\theta}$.

В рассматриваемом примере для эллиптической доверительной области при $\beta = 0,68$ имеем $k_{\beta}^2 = 3,8$. Если $k_{\beta} = 1$, то получим $\beta = 0,39$. При коэффициенте корреляции $\rho(\hat{\theta}_1, \hat{\theta}_2) = 0,95$ и квантили $k_{\beta} = 1$ прямоугольная область, включающая в себя эллипс, имеет доверительную вероятность $\beta = 0,498$ [68].

Из рис. 7.7–7.9 можно определить эллиптические и прямоугольные доверительные области.

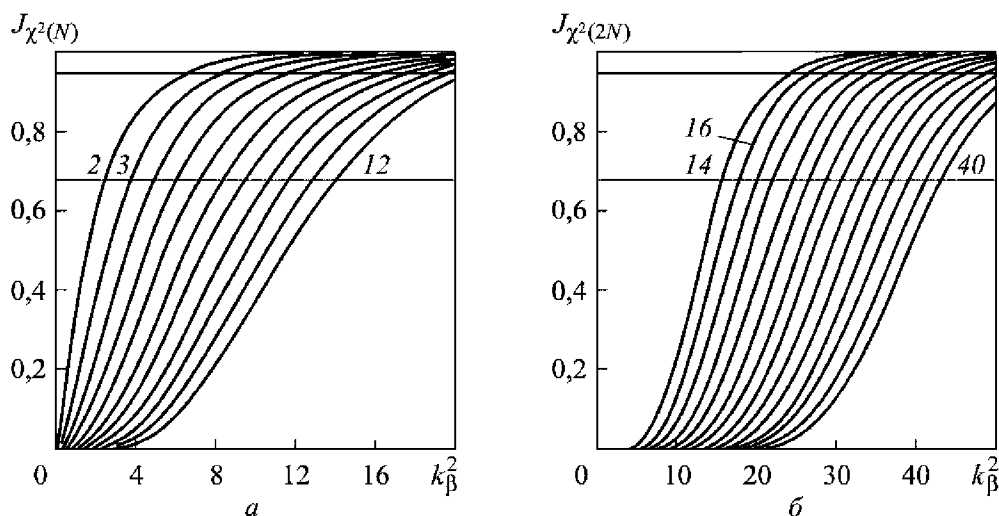


Рис. 7.7. Интеграл функций χ^2 -распределений:

a — для числа параметров $N = 2, 3, \dots, 12$; $б$ — для числа параметров $2N = 14, 16, \dots, 40$

В качестве примера определим вероятность $P(k)$ попадания случайной точки для нормального закона распределения в эллипс рассеяния, полуоси которого равны k средним квадратическим отклонениям по соответствующим осям. Для этого вычислим двойной интеграл

$$P(k) = \iint_G \frac{1}{2\pi\sigma(x)\sigma(y)} \exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma^2(x)} + \frac{y^2}{\sigma^2(y)}\right)\right\} dx dy$$

от двумерной плотности распределения вероятностей с нормальным законом распределения в границах эллипса рассеяния G :

$$\frac{x^2}{\sigma^2(x)} + \frac{y^2}{\sigma^2(y)} = k^2,$$

где $\sigma(x)$ и $\sigma(y)$ — средние квадратические отклонения по осям x и y соответственно. Получим

$$P(k) = 1 - e^{-k^2/2},$$

т. е. если каждая полуось эллипса равна одному среднему квадратическому отклонению $\sigma(x)$ и $\sigma(y)$, то вероятность попадания точки в заданный эллипс рассеяния равна

$$P(k=1) = 1 - e^{-1/2} \approx 0,393.$$

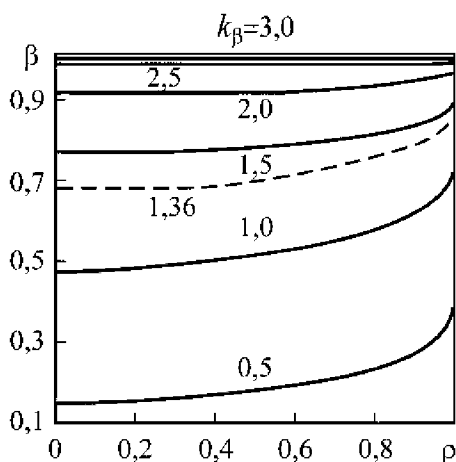


Рис. 7.8. Связь между квантилью k_β описанного прямоугольника с вероятностным содержанием β вписанного эллипса, имеющего коэффициент корреляции ρ

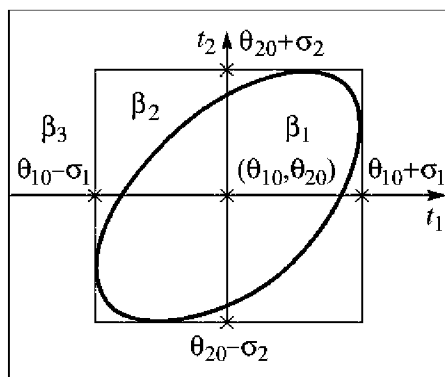


Рис. 7.9. Доверительные области для нормальных оценок $\hat{\theta}$ параметра θ при $k_\beta = 1$ и $\rho = 0,5$; β_1 , β_2 , β_3 — соответственно вероятностное содержание для эллиптической области, описанного прямоугольника и горизонтальной полосы

§ 7.6. Оценка параметров моделей с помощью функции правдоподобия

Одним из наиболее часто применяемых методов оценки свободных параметров функций известного вида (в данном случае функций распределения и плотностей распределения вероятностей) является метод максимума правдоподобия.

Пусть случайная величина x имеет плотность распределения вероятностей $f(x, \theta)$, где θ — неизвестный вектор параметров этого распределения. Проведем независимые выборки и получим реализации x_1, x_2, \dots, x_n . Дифференциал совместной плотности распределения вероятностей результатов наблюдений имеет вид

$$L(x_1, x_2, \dots, x_n; \theta) dx_1 dx_2 \dots dx_n = f(x_1, \theta) f(x_2, \theta) \dots f(x_n, \theta) dx_1 dx_2 \dots dx_n.$$

Совместная плотность вероятности $L(x, \theta) = \prod_{i=1}^n f(x_i, \theta)$ называется функцией *правдоподобия* для выборки x_1, x_2, \dots, x_n .

Если дискретная случайная величина x принимает значение x_1, x_2, \dots, x_n , то в формуле для функции правдоподобия следует плотности заменить вероятностями событий $x = x_i$:

$$f(x_i, \theta) = P\{x = x_i\}.$$

Можно найти такую оценку $\hat{\theta}$, которая доставляет функции правдоподобия $L(\theta)$ максимум: $L(\hat{\theta}) = \max_{\theta} L(\theta)$ и является точечной оценкой параметра θ , т. е. для вычисления оценки $\hat{\theta}$ необходимо решить систему уравнений

$$\frac{\partial L(x, \theta)}{\partial \theta_j} = 0, \quad j = 1, 2, \dots, m.$$

Предположим, что оценка $\hat{\theta}$ одного параметра θ удовлетворяет следующим условиям (условиям регулярности):

1) $\hat{\theta} = \theta(x_1, x_2, \dots, x_n)$ — однозначная функция от x_1, x_2, \dots, x_n , непрерывная и имеющая непрерывные частные производные $\partial \theta / \partial x_i$ во всех точках $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, при этом операции вычисления частных производных и интегрирования коммутируют;

2) область значений x не зависит от вектора параметров θ .

Предположим также, что математическое ожидание оценки имеет вид $M(\hat{\theta}) = \theta + b(\theta)$, где $b(\theta)$ — смещение оценки $\hat{\theta}$.

Теорема. В каждом случае регулярной оценки непрерывного типа квадрат среднего квадратического отклонения оценки $\hat{\theta}$ от истинного значения θ удовлетворяет неравенству

$$D(\hat{\theta}) = M[(\hat{\theta} - \theta)^2] \geq \frac{\left(1 + \frac{db}{d\theta}\right)^2}{\int_{-\infty}^{+\infty} \left(\frac{\partial \ln L}{\partial \theta}\right)^2 L(x, \theta) dx}.$$

Доказательство. Имеем

$$\int_{-\infty}^{+\infty} L(\hat{\theta} | \theta) d\hat{\theta} = 1.$$

Дифференцируя это выражение по θ , получим следующие соотношения:

$$\int_{-\infty}^{+\infty} \frac{\partial L}{\partial \theta} d\hat{\theta} = 0 = \int_{-\infty}^{+\infty} \frac{\partial \ln L}{\partial \theta} L(\hat{\theta} | \theta) d\hat{\theta},$$

так как $\frac{\partial \ln L}{\partial \theta} = \frac{1}{L} \frac{\partial L}{\partial \theta}$.

Имеем

$$M(\hat{\theta}) = \int_{-\infty}^{+\infty} \hat{\theta} L(\hat{\theta} | \theta) d\hat{\theta} = \theta + b(\theta).$$

Дифференцируя это равенство по θ , получим

$$\frac{\partial}{\partial \theta} \int_{-\infty}^{+\infty} \hat{\theta} L(\hat{\theta} | \theta) d\hat{\theta} = 1 + \frac{db(\theta)}{d\theta}. \quad (7.13)$$

Поскольку выражение $\theta + b(\theta)$ не зависит от $\hat{\theta}$, то выполняется равенство

$$\int_{-\infty}^{+\infty} (\theta + b(\theta)) \frac{\partial \ln L}{\partial \theta} L(\hat{\theta} | \theta) d\hat{\theta} = 0. \quad (7.14)$$

Вычитая выражение (7.14) из соотношения (7.13), получим

$$\int_{-\infty}^{+\infty} (\hat{\theta} - \theta - b(\theta)) \frac{\partial \ln L}{\partial \theta} L(\hat{\theta} | \theta) d\hat{\theta} = 1 + \frac{db(\theta)}{d\theta}. \quad (7.15)$$

Используя неравенство Шварца

$$\left(\int_{-\infty}^{+\infty} f(x)g(x)dx \right)^2 \leq \int_{-\infty}^{+\infty} f^2(x)dx \int_{-\infty}^{+\infty} g^2(x)dx,$$

преобразуем интеграл (7.15) к следующему виду:

$$\int_{-\infty}^{+\infty} (\hat{\theta} - \theta - b(\theta))^2 L(\hat{\theta} | \theta) d\hat{\theta} \int \left(\frac{\partial \ln L}{\partial \theta} \right)^2 L(\hat{\theta} | \theta) d\hat{\theta} \geq \left(1 + \frac{db(\theta)}{d\theta} \right)^2.$$

Здесь первый интеграл — выражение для дисперсии $D(\hat{\theta})$, т. е.

$$D(\hat{\theta}) \geq \frac{\left(1 + \frac{db(\theta)}{d\theta} \right)^2}{M \left[\left(\frac{\partial \ln L(\hat{\theta} | \theta)}{\partial \theta} \right)^2 \right]}.$$

Это неравенство называется *неравенством Крамера — Рао*. Теорема доказана.

Поскольку для нахождения $D(\hat{\theta})$ используется $\ln L(\hat{\theta} | \theta)$ и поскольку точки максимумов функций $L(\hat{\theta} | \theta)$ и $\ln L(\hat{\theta} | \theta)$ совпадают в силу монотонности логарифма, то для определения точечной оценки $\hat{\theta}$ решается *уравнение правдоподобия*

$$\frac{\partial \ln L(\hat{\theta} | \theta)}{\partial \theta} = 0.$$

Дифференцирование функции $\ln L(\hat{\theta} | \theta)$ вместо функции $L(\hat{\theta} | \theta)$ часто упрощает расчеты.

Метод определения оценки $\hat{\theta}$ с помощью функции правдоподобия называется *методом максимума правдоподобия* (ММП).

В асимптотике (для состоятельных оценок) имеем

$$M \left[\left(\frac{\partial \ln L}{\partial \theta} \right)^2 \right] \Big|_{\theta=\hat{\theta}} = -M \left[\left(\frac{\partial^2 \ln L}{\partial \theta^2} \right) \right] \Big|_{\theta=\hat{\theta}}.$$

Из неравенства Крамера — Рао следует, что наименьшая величина дисперсии оценки (для эффективной оценки) определяется по формуле

$$D(\hat{\theta}) = \frac{1}{\int_{-\infty}^{+\infty} \left(\frac{\partial \ln L}{\partial \theta} \right)^2 L(x, \theta) dx} = \frac{1}{M \left[\left(\frac{\partial \ln L}{\partial \theta} \right)^2 \right]}.$$

Значит, ковариационная матрица вектора оценок $\hat{\theta}$ найдется из условия

$$D(\hat{\theta}) = N^{-1}, \quad N = \left\| -\frac{\partial^2 \ln L}{\partial \theta_i \partial \theta_j} \right\|, \quad i, j = 1, 2, \dots, k.$$

Пример. Найти точечные оценки математического ожидания m и дисперсии σ^2 функции плотности нормально распределенных случайных величин.

Пусть произвели выборку $\{x_i\}$, $i = 1, 2, \dots, n$, и получили n независимых значений, имеющих нормальное распределение. Функция правдоподобия в этом случае имеет вид

$$L(m, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma}} \exp \left\{ -\frac{(x_i - m)^2}{2\sigma^2} \right\}.$$

Тогда

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

Используя метод максимума правдоподобия, получим уравнения

$$\frac{\partial \ln L}{\partial m} = \frac{1}{\sigma^2} \sum_{i=1}^n (x_i - m) = 0,$$

$$\frac{\partial \ln L}{\partial \sigma^2} = \frac{1}{2\sigma^4} \sum_{i=1}^n (x_i - m)^2 - \frac{n}{2\sigma^2} = 0.$$

Отсюда получим следующие оценки:

$$\hat{m} = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}, \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2.$$

Математическое ожидание оценки \hat{m} есть $M(\hat{m}) = M(\bar{x}) = M(x_i) = \mu$, т. е. \hat{m} является несмещенной оценкой.

Найдем математическое ожидание оценки $\hat{\sigma}^2$. Математические ожидания каждого слагаемого в формуле для $\hat{\sigma}^2$ равны. Тогда

$$\begin{aligned}
 M(\hat{\sigma}^2) &= \frac{1}{n} M[(x_i - \bar{x})^2] = M[(x_i - \mu)^2 + (\bar{x} - \mu)^2 - 2(x_i - \mu)(\bar{x} - \mu)] = \\
 &= \sigma^2 + \frac{\sigma^2}{n} - 2 \frac{\sigma^2}{n} = \sigma^2 \frac{n-1}{n}.
 \end{aligned}$$

Получили, что оценка $\hat{\sigma}^2$ является смещенной.

Несмещенная оценка (исправленная) дисперсии имеет вид

$$s^2 = \frac{n}{n-1} \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2.$$

При выполнении условий регулярности (наиболее важное из них — независимость границы области определения функции $f(x, \theta)$ от θ) оценки, полученные методом максимального правдоподобия, состоятельны и, кроме того, асимптотически нормальны и эффективны.

Если функция правдоподобия имеет вторые производные по θ в интервале, включающем истинное значение θ_0 , и если, кроме того,

$$M \left[\frac{\partial \ln L(x, \theta)}{\partial \theta_i} \right] = 0$$

и существует матрица

$$N(\theta) = -M \left[\frac{\partial^2 \ln L(x, \theta)}{\partial \theta_i \partial \theta_j} \right],$$

элементы которой отличны от нуля в указанном интервале, то оценка $\hat{\theta}$ асимптотически нормально распределена со средним значением θ_0 и дисперсией $[N(\theta)]^{-1}$.

Важным свойством оценки ММП является ее инвариантность: оценка $\hat{\tau}$ функции $\tau(\theta)$ является функцией оценки $\hat{\tau} = \tau(\hat{\theta})$. Например, оценка функции $\tau(\theta) = \theta^2$ равна квадрату оценки $\hat{\theta}$, но оценка $\hat{\tau}$ не обязательно несмещенная.

Интервальная оценка параметра θ может быть получена по графику функции правдоподобия (рис. 7.10).

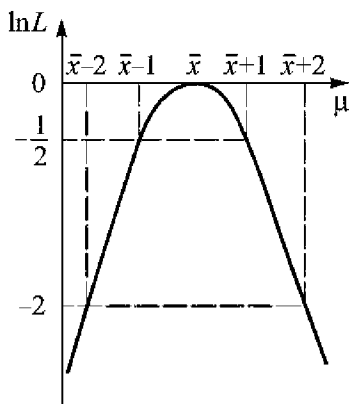


Рис. 7.10. Интервальная оценка, построенная по функции правдоподобия

Одномерный случай. Пусть результаты наблюдений распределены по нормальному закону $N(\mu, \sigma^2)$. Логарифм функции правдоподобия имеет вид параболы:

$$\ln L(x, \mu) = \ln C - \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2 = \ln C - \frac{1}{2} \sum_{i=1}^n (x_i - \bar{x})^2 - \frac{n}{2} (\mu - \bar{x})^2$$

с точкой максимума при $\mu = \bar{x}$. Сдвигаем параболу так, чтобы $\ln L(\bar{x}) = 0$.

Пусть $\ln L = -1/2$ и $n=1$. Тогда

$$\frac{(\mu - \bar{x})^2}{2} = \frac{1}{2}.$$

Отсюда получим доверительный интервал $\bar{x} - 1 \leq \mu \leq \bar{x} + 1$. Однако из свойств нормального распределения следует, что

$$P\{(\bar{x} - \mu)^2 \leq 1\} = 0,683,$$

т. е. $P\{\bar{x} - 1 \leq \mu \leq \bar{x} + 1\} = 0,683$ (см. рис. 7.10).

Пусть $\ln L = -2$; тогда $(\mu - \bar{x})^2 / 2 = 2$. Отсюда получаем доверительный интервал $|\mu - \bar{x}| \leq 2$, или $\bar{x} - 2 \leq \mu \leq \bar{x} + 2$.

Из условия нормального распределения имеем

$$P\{(\bar{x} - \mu)^2 \leq 2\} = 0,955,$$

или $P\{\bar{x} - 2 \leq \mu \leq \bar{x} + 2\} = 0,955$.

Общий случай. Если $L(x, \theta)$ является непрерывной функцией и имеет один максимум, можно найти оператор $g(\theta, x)$, который преобразует кривую $\ln L_\theta(x, \theta)$ в параболу вблизи функции $G(x)$ наблюдений:

$$\ln L_g(x, g(\theta)) = \frac{1}{2}(g - G(x))^2.$$

В асимптотическом случае доказывается, что g можно выбрать независимо от x . Тогда G является оценкой максимального правдоподобия для g .

Используя свойство инвариантности ММП, можно сделать такие же выводы о параметре g , какие делались о параметре нормального распределения θ . Преобразуем доверительный интервал для значений g в соответствующий интервал для значений θ . Поскольку

$$\ln L_\theta(x, \theta) = \ln L_g(x, g(\theta)),$$

то значения θ , соответствующие интервалу для g , при $\ln L_g = -2$ являются точками пересечения прямой $\ln L = -2$ с графиком функции $\ln L_\theta(x, \theta)$. Таким образом, можно сделать выводы об интервальной оценке параметра непараболической функции правдоподобия, не проводя преобразования к параболической функции правдоподобия.

Заметим, что интервальные оценки параметров, полученные по графику функции $\ln L$, будут отличаться от тех же оценок, полученных с помощью матрицы вторых производных, так как в последнем случае проводится квадратичная аппроксимация функции $\ln L$ в окрестности точки максимума. Однако графическая процедура имеет свои особенности:

1) точность способа преобразования имеет порядок $1/n$, так как преобразуем экспериментальную функцию правдоподобия, тогда как нужно было бы получить (преобразовать) нормальное распределение вместо полученного распределения;

2) полученный интервал может быть «широким» и несимметричным для первоначальной переменной θ ;

3) если функция правдоподобия имеет несколько максимумов, то эта процедура выявляет несколько интервалов, однако доверительным интервалом для θ может быть только один непрерывный интервал с тем же самым вероятностным содержанием (не следует забывать, что интервальная оценка указывает интервал, который с вероятностью β накрывает истинное значение θ);

4) если предложенным методом получаем неопределенные (или бесконечные) доверительные интервалы, то для решаемой задачи требуется более сложная интерпретация, чем интерпретация ее как задачи нахождения единственной оценки ММП и доверительного интервала.

§ 7.7. Байесовский подход к оцениванию параметров моделей

В байесовском подходе неизвестный параметр θ является не детерминированной (постоянной) величиной, а случайной величиной, для которой априори известна плотность распределения $\pi(\theta)$ возможных значений θ .

После наблюдения x получим плотность апостериорного распределения вероятностей

$$\pi(\theta | x) = \frac{\left[\prod_{i=1}^n f(x_i | \theta) \right] \pi(\theta)}{\left[\prod_{i=1}^n f(x_i | \theta) \right] \pi(\theta) d\theta}.$$

Байесовский доверительный интервал (θ_n, θ_b) с вероятностным содержанием β определяется следующим образом:

$$\int_{\theta_n}^{\theta_b} \pi(\theta | x) d\theta = \beta.$$

Согласно этому определению доверительный интервал соответствует доле β полной веры наблюдателя в значение оценки параметра θ , т. е. составляет $\beta / (1 - \beta)$. Полное апостериорное распределение $\pi(\theta | x)$ как бы заменяется дискретным двухточечным распределением:

$$P\{\theta_n \leq \theta \leq \theta_b\} = \beta, \quad P\{\theta \notin [\theta_n, \theta_b]\} = 1 - \beta.$$

Как и в классическом варианте нахождения интервальной оценки, доверительный интервал определяется неоднозначно. В обоих случаях не указывается, какое значение внутри интервала является наиболее вероятным. Поэтому желательно дополнительно провести точечную оценку по максимуму функции распределения вероятностей $\pi(\theta | x)$.

Замечание. Для нелинейных функций оценку свободных параметров часто проводят после линеаризации заданной функции, например: $\eta = Ae^{\lambda x} \Rightarrow \ln \eta = \ln A + \lambda x$ или $\eta_1 = \theta_1 + \theta_2 x$, где $\eta_1 = \ln \eta$, $\theta_1 = \ln A$, $\theta_2 = \lambda$. Однако здесь возникают две проблемы: единственности оценок и изменения законов распределения исходных данных, — которые повлияют на величину интервальных оценок.

§ 7.8. Интервальные оценки линии регрессии и прогнозируемых значений функции

Пусть для функции $\eta = f(x, \theta)$ получены оценки $\hat{\theta}$ и их дисперсии $D(\hat{\theta})$. Это означает, что по результатам наблюдений можно построить семейство линий. Для модели $\eta = f(x, \theta)$ оценкой линии регрессии η будет функция $\hat{y} = f(x, \hat{\theta})$. Необходимо опреде-

лечь доверительные интервалы для значений y , которые с заданной доверительной вероятностью β накроют истинные значения η . Кроме того, необходимо найти доверительные интервалы для прогнозов, которые накроют новое наблюдение y , соответствующее заданному значению x .

Рассмотрим эти оценки в случае, когда функция $\eta = \theta_0 + \theta_1(x - \bar{x})$ является линейной, где \bar{x} — среднее значение переменной x . Тогда дисперсия значений \hat{y} определяется по формуле

$$D(\hat{y}) = D(\hat{\theta}_0) + D[\hat{\theta}_1(x - \bar{x})] = D(\hat{\theta}_0) + (x - \bar{x})^2 D(\hat{\theta}_1).$$

Величина $\frac{|\hat{y} - \eta|}{\sqrt{D(\hat{y})}}$ подчиняется распределению Стьюдента с $n - p$ степенями свободы; при заданном β квантиль распределения Стьюдента равна t_β и интервальная оценка имеет вид

$$P\{\hat{y} - t_\beta \sqrt{D(\hat{y})} < \eta < \hat{y} + t_\beta \sqrt{D(\hat{y})}\} = \beta.$$

В общем случае для линии регрессии, описываемой уравнением $\eta(x, \theta) = f(x, \theta)$, дисперсия вычисленных значений \hat{y} определяется выражением

$$D(\hat{y}) = \sum_{j=1}^N \left(\frac{\partial f}{\partial \theta_j} \right)^2 D(\hat{\theta}_j) + 2 \sum_{i=1}^N \sum_{\substack{j=2 \\ j < i}}^N \frac{\partial f}{\partial \theta_j} \frac{\partial f}{\partial \theta_i} D(\hat{\theta}_j, \hat{\theta}_i),$$

где N — размерность вектора θ , $D(\hat{\theta}_j, \hat{\theta}_i)$ — корреляционный момент случайных величин $\hat{\theta}_j$ и $\hat{\theta}_i$.

Для линейных функций $\eta(x, \theta) = \sum_{j=1}^N \theta_j f_j(x)$ дисперсия вычисляемых значений \hat{y} в точке x согласно той же формуле может быть записана в виде

$$D(\hat{y}) = f^T(x) D(\hat{\theta}) f(x);$$

здесь $f(x)$ — вектор, $D(\hat{\theta})$ — матрица оценок $\hat{\theta}$.

Например, для прямой линии $\eta(x, \theta) = \theta_0 + \theta_1(x - \bar{x})$, если $f(x) = (1, x - \bar{x})^T$, имеем

$$D(\hat{y}) = (1, x - \bar{x}) \begin{pmatrix} D(\hat{\theta}_0) & 0 \\ 0 & D(\hat{\theta}_1) \end{pmatrix} \begin{pmatrix} 1 \\ x - \bar{x} \end{pmatrix} = D(\hat{\theta}_0) + (x - \bar{x})^2 D(\hat{\theta}_1).$$

Задав серию значений x и вычислив соответствующие им значения дисперсии $\sigma^2(\hat{y}) = D(\hat{y})$, получим интервальную оценку линии. Для прямой линии интервальную оценку можно получить аналитически [40]. Интервальные границы для $M(\eta | \xi_0)$ в случае прямой линии будут следующие:

$$\hat{\eta} - t_{1-\alpha/2}\sigma(\eta) \leq M(\eta | \xi_0) < \hat{\eta} + t_{1-\alpha/2}\sigma(\eta), \quad (7.16)$$

где $\sigma^2(\hat{\eta}) = (1, \xi_0)D(\hat{\theta})(1, \xi_0)^T$, α — уровень значимости.

Огибающая семейства всех возможных прямых будет кривой второго порядка, уравнение которой получим из формулы (7.16):

$$\sum_{i=1}^n (y_i - \theta_1 - \theta_2 x_i)^2 \leq C_\alpha (1 + \theta_2^2),$$

где C_α — константа, величина которой определяется уровнем значимости α .

Принимая за начало отсчета наблюдаемые средние значения \bar{x} и \bar{y} , перепишем это неравенство в виде [40]

$$\theta_2^2 (s_x^2 - C_\alpha) - 2\theta_2 s_{xy} + \theta_1^2 \leq C_\alpha - s_y^2,$$

где s_x^2 , s_y^2 и s_{xy} — соответственно выборочные дисперсии и выборочная ковариация величин x и y . Это условие можно считать ограничением, которому удовлетворяет «истинная» прямая.

Найдем огибающую семейства всех возможных прямых. Уравнение огибающей [40] имеет вид

$$\frac{(y - \theta_2 x)^2}{C_\alpha - b_1} - \frac{(\hat{\theta}_2 y + x)^2}{b_2 - C_\alpha} = 1 + \hat{\theta}_2^2,$$

где

$$b_1 = s_x^2 - \frac{s_{xy}^2}{\hat{\theta}_2}, \quad b_2 = s_x^2 + \hat{\theta}_2 s_{xy}, \quad \hat{\theta}_2 > 0, \quad b_2 > b_1.$$

Новое наблюдаемое (прогнозируемое) значение y отсчитывается от точек на линии оценки функции регрессии \hat{y} , и величина

$\frac{|y - \hat{y}|}{\sqrt{D(y - \hat{y})}}$ имеет распределение Стьюдента. Предположим, что новое наблюдение имеет дисперсию σ^2 ; тогда

$$D(y - \hat{y}) = \sigma^2 + D(\hat{\theta}_0) + (x - \bar{x})^2 D(\hat{\theta}_1).$$

Интервальные оценки для точек прогноза имеют вид

$$P\{\hat{y} - t_\beta \sqrt{D(y - \hat{y})} < y < \hat{y} + t_\beta \sqrt{D(y - \hat{y})}\} = \beta. \quad (7.17)$$

Для k измерений, проводимых в точке прогноза, вместо значения дисперсии σ^2 в формуле для $D(y - \hat{y})$ надо взять величину σ^2/k . Для параметров прямой линии нетрудно получить оценки

$$\hat{\theta}_0 = \frac{1}{n} \sum_{i=1}^n y_i = \bar{y}, \quad \hat{\theta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x}) y_i}{\sum_{i=1}^n (x_i - \bar{x})^2}.$$

Пусть $D(y_i) = \sigma^2$, $i = 1, 2, \dots, n$. Тогда

$$D(\hat{\theta}_0) = D(\bar{y}) = \frac{\sigma^2}{n},$$

$$D(\hat{\theta}_1) = D \left[\frac{\sum_i (x_i - \bar{x}) y_i}{\sum_i (x_i - \bar{x})^2} \right] = \frac{\sum_i (x_i - \bar{x})^2 D(y_i)}{\left(\sum_i (x_i - \bar{x})^2 \right)^2} = \frac{\sigma^2}{\sum_i (x_i - \bar{x})^2}.$$

Из формул для $D(\hat{\theta}_0)$ и $D(\hat{\theta}_1)$ следует, что дисперсия $D(\hat{y})$ минимальна при $x = \bar{x}$. Задавая различные значения x , можно найти

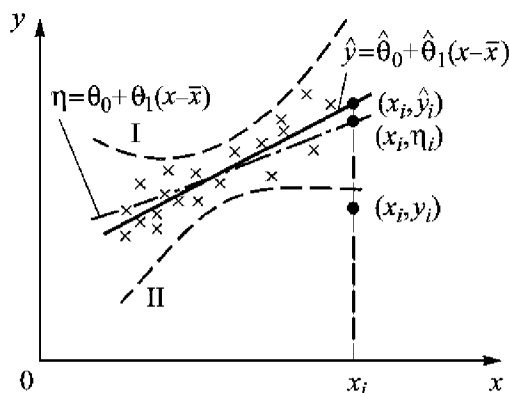


Рис. 7.11. Интервальная оценка линии регрессии

интервальные оценки линии регрессии (линии I и II на рис. 7.11) при заданном β и интервальную оценку нового наблюдения y_i для прогнозируемой точки x_i .

На рис. 7.11 точка (x_i, y_i) показывает нижнее значение интервальной оценки прогнозируемого значения y_i в точке x_i .

§ 7.9. Активный и пассивный эксперименты. Оценивание параметров функции известного вида в пассивном эксперименте

В настоящем параграфе будут рассмотрены алгоритмы, позволяющие находить оценки свободных параметров элементарных функций при наличии погрешностей и в значениях аргумента, и в значениях функции. Эти алгоритмы объединены названием «конфлюэнтный анализ» (англ. *confluence* — слияние, *confluent* — сливающийся). Когда и значения функции, и значения аргумента — случайные величины, процедуры оценки свободных параметров функции зависят от того, как проводился эксперимент: пассивно считывались значения аргумента и функции или значения аргумента устанавливались заранее.

Пассивный эксперимент наиболее часто встречается в процессах получения оценок параметров моделей систем во многих отраслях науки и техники. Здесь требуется найти интервальную оценку параметра θ функции $\eta = f(\xi, \theta)$, когда точные значения η и ξ мы наблюдать не можем, однако можем наблюдать значения случайных величин y и x :

$$x_i = \xi_i + \delta_i, \quad y_i = \eta_i + \varepsilon_i, \quad i = 1, 2, \dots, n,$$

где δ_i и ε_i — соответственно ошибки значений аргументов и функции (случайные величины).

В активном эксперименте мы можем задавать x любые значения (x называется *контролируемой переменной*), на которые налагается помеха δ : $\xi = x + \delta$. Значения y являются результатом влияния случайной ошибки на истинное значение η : $y = \eta + \varepsilon$. Структурное соотношение между наблюдаемыми значениями x_i и y_i примет вид $y = f(x + \delta, \theta) + \varepsilon$ и станет обычным уравнением регрессии, поскольку теперь x не случайная величина.

Рассмотрим пассивную схему эксперимента. Пусть заданы статистический ряд экспериментальных значений $\{x_i\} \subset X$ и соответствующий им ряд значений функции $\{y_i\} \subset Y$, $i = 1, 2, \dots, n$, $n \geq m$, где m — число оцениваемых параметров θ . Предположим, что переменные x_i и y_i не детерминированы, но являются выборками из генеральных совокупностей X и Y с известными функциями плотности распределения вероятностей. Переменные $x_i = \xi_i + \delta_i$ и $y_i = \eta_i + \epsilon_i$ могут быть статистически как зависимы, так и не зависимы; могут быть как коррелированы, так и не коррелированы.

В основном мы будем рассматривать выборки независимых наблюдений, имеющих один закон распределения.

Пусть $f_1(x_i | \theta)$ и $f_2(y_i | \theta)$ — соответственно функции плотности вероятностей случайных величин x_i и y_i , если x_i и y_i непрерывны, и соответственно вероятности значений x_i и y_i , если распределения величин x_i и y_i дискретны; функции $f_1(x_i | \theta)$ и $f_2(y_i | \theta)$ могут быть как одномерными, так и многомерными.

Найдем выражение для совместной плотности вероятности экспериментальных данных при условии, что значения ξ_i и η_i связаны функциональной зависимостью, но их погрешности δ_i и ϵ_i являются независимыми при переходе от одной точки к другой. Тогда совместная плотность вероятности получения одновременно значений x_i и y_i имеет вид

$$p_i = f_1(x_i | \theta) f_2(y_i | \theta).$$

Совместная плотность вероятности получения n статистически независимых точек (x_i, y_i) является функцией

$$L(x, y | \theta) = \prod_{i=1}^n f_1(x_i | \theta) f_2(y_i | \theta).$$

Аналогично можно вывести формулы совместной плотности вероятности для зависимых или коррелированных экспериментальных точек [40, 49, 89].

Для нас важно то, что в выражения для совместной плотности вероятности входят математические ожидания экспериментальных данных, экспериментальные значения и оцениваемые параметры, так как в $f_1(x_i | \theta)$ войдут функция математического ожидания ξ_i , экспериментальные значения x_i и вектор параметров θ , в $f_2(y_i | \theta)$ — функция математического ожидания η_i , эксперимен-

тальные значения y_i , x_i и вектор параметров θ . Кроме того, нам известно функциональное соотношение $\eta_i = f(\xi_i, \theta)$, которое порождает структурное соотношение между наблюдаемыми случайными величинами x_i и y_i :

$$y_i = \psi(x_i, \theta, \delta_i, \varepsilon_i)$$

или

$$y_i = f(x_i - \delta_i, \theta) + \varepsilon_i$$

при аддитивных помехах δ_i , ε_i .

Таким образом, в поставленной задаче следует отметить две проблемы:

- 1) как ввести в рассмотрение погрешность аргумента;
- 2) как минимизировать функционалы при отыскании точечных оценок параметров после введения в них оценок аргументов или выражений для них, так как эти функционалы имеют сложную форму и соответствующие системы уравнений для определения этих оценок нелинейны (известно, что каждая нелинейная система требует особого рассмотрения каждого решения).

Наиболее часто выборка случайных величин имеет распределение Гаусса. Найдем (в качестве примера) для этого распределения вид функционалов, из которых затем могут быть получены оценки искомых параметров (для других функций распределений экспериментальных данных процедура получения минимизируемого функционала будет аналогичной).

Пусть экспериментальные значения x_i и y_i — случайные величины, каждая из которых имеет функцию плотности распределения вероятностей, описываемую функцией Гаусса с математическими ожиданиями ξ_i и η_i , дисперсиями $\sigma^2(x_i)$ и $\sigma^2(y_i)$ и коэффициентом корреляции $\rho_i = \rho(x_i, y_i)$. Тогда плотность вероятности получения точки с координатами (x_i, y_i) имеет вид

$$p_i = \frac{1}{2\pi\sigma(x_i)\sigma(y_i)\sqrt{1-\rho_i^2}} \exp \left\{ -\frac{u_{1i}^2 - 2\rho_i u_{1i} u_{2i} + u_{2i}^2}{2(1-\rho_i^2)} \right\},$$

где

$$u_{1i} = \frac{x_i - \xi_i}{\sigma(x_i)}, \quad u_{2i} = \frac{y_i - \eta_i}{\sigma(y_i)}.$$

Совместная плотность вероятности получения n независимых таких точек является функцией

$$L(x, y) = \prod_{i=1}^n p_i,$$

или

$$\ln L(x, y) = -\frac{1}{2} \sum_{i=1}^n \left[\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} - 2\rho_i \frac{(x_i - \xi_i)(y_i - \eta_i)}{\sigma(x_i)\sigma(y_i)} + \frac{(y_i - \eta_i)^2}{\sigma^2(y_i)} \right] \times \\ \times \frac{1}{1 - \rho_i^2} + \text{const.}$$

Оценки искомых параметров θ находятся из условия минимума функционала

$$F = \frac{1}{2} \sum_{i=1}^n \left[\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} - 2\rho_i \frac{(x_i - \xi_i)(y_i - \eta_i)}{\sigma(x_i)\sigma(y_i)} + \frac{(y_i - \eta_i)^2}{\sigma^2(y_i)} \right]. \quad (7.18)$$

В частном случае, когда погрешности не некоррелированы, выражение (7.18) принимает вид

$$F = \frac{1}{2} \sum_{i=1}^n \left[\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} + \frac{(y_i - \eta_i)^2}{\sigma^2(y_i)} \right]. \quad (7.19)$$

Аналогично находят вид минимизируемого функционала при других законах распределения исходных данных. Чтобы перейти в выражениях (7.18), (7.19) к случаю, когда переменные являются детерминированными, следует положить $\xi_i = x_i$ или $\eta_i = y_i$.

Теперь рассмотрим задачу отыскания минимума функционала типа (7.18) по параметрам θ при условии, что задан вид функции $\eta = f(\xi, \theta)$.

Нам не известны истинные значения ξ (абсцисс) экспериментальных точек, а известны только их доверительные области, тогда как в регрессионном анализе абсциссы известны, причем случайная величина x коррелирована с обобщенной ошибкой функции η (например, для линейной функции $\eta = \theta_1 + \theta_2 \xi$ обобщенная ошибка равна $\varepsilon - \theta_2 \delta$).

Перед тем как определять точки минимума функционалов (7.18) и (7.19) по θ , необходимо каким-то образом определить ξ_i , а только затем, подставив выражения для ξ_i и η_i в функционал, найти минимум получившейся функции нескольких переменных.

Выход любой системы в виде набора значений y содержит информацию не только о параметрах θ модели, но и о действительных значениях входного воздействия ξ . Искомые оценки $\hat{\theta}$ определяются из равенства

$$\left. \frac{\partial F}{\partial \eta} \frac{\partial \eta}{\partial \theta_j} \right|_{\theta_j = \hat{\theta}_j} = 0, \quad j = 1, 2, \dots, m. \quad (7.20)$$

Однако в уравнениях (7.20) истинные значения ξ неизвестны. Чтобы найти их оценки $\hat{\xi}$, доопределим задачу следующим образом:

$$\left. \frac{\partial F}{\partial \xi_i} \right|_{\xi_i = \hat{\xi}_i} = 0, \quad i = 1, 2, \dots, n. \quad (7.21)$$

Очевидно, получаемые оценки значений ξ_i должны принадлежать области неопределенности D_i измеренных величин x_i , $i = 1, 2, \dots, n$, т. е. $\hat{\xi}_i \in D_i$. Когда известен закон распределения погрешности измерения величин x_i , это условие может быть выражено в более конкретной форме: при нормальном законе распределения случайной величины x_i можно считать, что $|x_i - \xi_i| \leq k\sigma(x_i)$, где значение коэффициента k определяется выбранным значением доверительной вероятности.

Таким образом, решение задачи минимизации функционала (7.19) при условии (7.21) эквивалентно решению системы уравнений

$$\sum_{i=1}^n \frac{y_i - \eta_i}{\sigma^2(y_i)} \frac{\partial f}{\partial \theta_j} = 0, \quad j = 1, 2, \dots, m, \quad n \geq m, \quad (7.22)$$

при

$$\frac{x_i - \xi_i}{\sigma^2(x_i)} + \frac{y_i - \eta_i}{\sigma^2(y_i)} \frac{\partial f}{\partial \xi_i} = 0, \quad i = 1, 2, \dots, n. \quad (7.23)$$

Для функций, линейных по параметрам θ , система уравнений (7.22) является системой линейных алгебраических уравнений. Система (7.23) для линейных по ξ функций представляет n не связанных между собой систем из m линейных уравнений.

Замена нормального закона распределения другим, например законом Пуассона или Лапласа, приводит к системе нелинейных уравнений (7.22).

При учете корреляции погрешностей величин x_i и y_i формулы усложняются, а при учете корреляции погрешностей значений x_{i1} и x_{i2} нельзя разбить систему из m уравнений на n независимых систем, в этом случае придется решать систему линейных уравнений в n раз большей размерности, чем при независимых переменных x_{i1} и x_{i2} . Учет корреляции погрешностей всегда ухудшает обусловленность системы уравнений (7.22).

Условие (7.23) для нелинейных функций $\eta = f(\xi, \theta)$ имеет вид n независимых систем из m нелинейных уравнений, которые можно решать методом линеаризации. Сходимость итерационного процесса при этом обеспечивается малостью допустимых интервалов для ξ .

Таким образом, сначала решают регрессионную задачу нахождения вектора оценок $\hat{\theta}$ при значениях $\xi_i = x_i$, $i = 1, 2, \dots, n$. Получают первое приближение для оценок $\hat{\theta}$. Затем определяют точные значения ξ_i , при этом проверяют принадлежность новых значений ξ_i области возможных значений D_i переменной x_i . Эти действия повторяют до тех пор, пока не выполнится одно из условий:

а) на очередном шаге значение функционала (7.19) меньше заданного числа γ ;

б) на соседних итерациях значение функционала F и значения оценок параметра θ мало отличаются, т. е.

$$\left| \frac{F_v - F_{v+1}}{F_v} \right| \leq \gamma_1, \quad \max \left| \frac{\theta_j^v - \theta_j^{v+1}}{\theta_j^v} \right| \leq \gamma_2, \quad j = 1, 2, \dots, m,$$

где γ_1 и γ_2 — заданные числа;

в) исчерпан лимит итераций.

Дисперсии оценок параметров находят с помощью матрицы вторых производных по θ от функционала (7.18) (см. § 7.6).

В работах [20–24] рассмотрено применение конфлюэнтного анализа для основных элементарных функций: прямой линии, линейных многомерных функций, полиномов, систем линейных алгебраических уравнений, сигнмов, кубических сплайнов.

Например, для линейной функции $\eta = a\xi + b$ с учетом выражений (7.19), (7.23) получим следующий минимизируемый функционал:

$$F = \sum_{i=1}^n \frac{(y_i - ax_i - b)^2}{\sigma^2(y_i) + a^2\sigma^2(x_i)},$$

если исходным был функционал (7.19), а значения $\hat{\xi}_i$ определим по формуле (согласно (7.23))

$$\hat{\xi}_i = \frac{\sigma^2(y_i)x_i + a\sigma^2(x_i)(y_i - b)}{\sigma^2(y_i) + a^2\sigma^2(x_i)}$$

и подставим затем в функционал (7.19).

Выражение в знаменателе $\sigma^2(y_i) + a^2\sigma^2(x_i)$ можно рассматривать как сумму дисперсий левой и правой частей i -го уравнения. Такой подход используют при оценке параметров функции многих переменных.

В случае, когда функция η является аналитическим описанием плотности нормального распределения, возникают трудности в нахождении оценок параметров, так как функционал (7.19) в данном случае имеет сложный вид. Поэтому оценки параметров и истинные значения ξ_i находят итерационными методами, предварительно определив «хорошее» нулевое приближение.

В приложении П1 приводится описание программы (инструкция пользователю) для построения линий регрессии X на Y , Y на X (регрессионный анализ) и ортогональной регрессии (конфлюэнтный анализ).

§ 7.10. Анализ других методов оценки параметров функции известного вида с учетом ошибок в значениях функций и аргументов

Предположим, что ошибки измерений ε_i и δ_i — нормально распределенные случайные величины с нулевыми средними значениями, дисперсиями $\sigma^2(y_i)$ и $\sigma^2(x_i)$ соответственно и коэффициентом корреляции $\rho_i = 0$.

Рассмотрим несколько подходов к решению задачи оценки параметров функции известного вида.

I. В одном из подходов используют вместо истинных значений ξ_i значения наблюдаемых x_i при оценивании параметров МНК, т. е. ошибка измерения δ_i игнорируется, в результате получают следующую модель:

$$y = f(x, \theta) + \varepsilon.$$

В основном, как показано в [20, 30, 40], этот подход приводит к несостоятельной оценке с большим асимптотическим смещением.

II. В работе [96] рассмотрена задача, когда ξ является случайной величиной, не зависящей от ε_i и δ_i , с характеристиками $M(\xi) = \mu$, $\text{cov}(\xi) = D_\xi$. Значения ξ_i аппроксимируются результатами измерений x_i , и при нормальном законе распределения случайной величины ξ_i полагают

$$D(\xi | x) = x\Lambda + \mu(I - \Lambda),$$

где $\Lambda = (D_\xi + \sigma^2(x)I)^{-1}D_\xi$, I — единичная матрица.

Получаем функцию

$$y = f(x\Lambda + \mu(I - \Lambda), \theta) + \varepsilon.$$

Если μ и Λ не известны, то, используя исходную выборку x_i , $i = 1, 2, \dots, n$, определяем оценки $\hat{\mu}$, $\hat{\Lambda}$ и получаем модель

$$y = f(x\hat{\Lambda} + \hat{\mu}(I - \hat{\Lambda}), \hat{\theta}) + \varepsilon = f(\hat{\xi}, \hat{\theta}) + \varepsilon. \quad (7.24)$$

Оценки параметров модели (7.24) могут быть определены МНК. Например, для прямой линии $\eta = a\xi + b$ имеем

$$\hat{\xi} = x \frac{\sigma^2(x)}{\sigma^2(x) + \sigma_0^2} + \mu \left(1 - \frac{\sigma^2(x)}{\sigma^2(x) + \sigma_0^2} \right),$$

где σ_0^2 — дисперсия ошибки δ_i , $\sigma^2(x)$ — дисперсия вариационного ряда x . Минимизируемый функционал примет вид

$$F = \frac{\sum_{i=1}^n [y_i - \eta(\xi_i, \theta)]^2}{\sigma^2(y)} = \frac{\sum_{i=1}^n [(y_i - b - a\mu)(\sigma^2(x) + \sigma_0^2) - (x_i - \mu)a\sigma^2(x)]^2}{(\sigma^2(x) + \sigma_0^2)^2 \sigma^2(y)},$$

где $\sigma^2(y)$ — дисперсия вариационного ряда y .

В случае, когда функция $f(\xi, \theta)$ нелинейна по параметрам, используют разложение ее в ряд Тейлора в окрестности точки $\hat{\xi}$ [30,

96]. Например, при нормальном законе распределения с параметрами θ_1 и θ_2 , используя формулу $\eta = f(\xi) + f'(\xi)(x - \xi)$, получим

$$\begin{aligned} \eta &\approx \frac{1}{\sqrt{2\pi} \cdot \theta_2} \exp \left\{ -\frac{(x - \theta_1)^2}{2\theta_2^2} \right\} \Big|_{x=\hat{\xi}} + \\ &+ \frac{1}{\sqrt{2\pi} \cdot \theta_2} \exp \left\{ -\frac{(x - \theta_1)^2}{2\theta_2^2} \right\} \Big|_{x=\hat{\xi}} \left[-\frac{(x - \theta_1)}{\theta_2^2} \right] \Big|_{x=\hat{\xi}} (x - \hat{\xi}) = \\ &= \frac{1}{\sqrt{2\pi} \cdot \theta_2} \exp \left\{ -\frac{(\hat{\xi} - \theta_1)^2}{2\theta_2^2} \right\} \left[1 + \frac{\theta_1 - \hat{\xi}}{\theta_2^2} (x - \hat{\xi}) \right]. \end{aligned}$$

Оценки параметров θ_1 и θ_2 найдем из условия минимума функционала

$$F = \frac{\sum_{i=1}^n [y_i - \eta(\xi_i, \theta)]^2}{\sigma^2(y)}.$$

III. Поставленная задача может быть решена итеративным методом наименьших квадратов с уточняемыми весами [30, 75, 76, 102].

Предполагается, что x_i , $i = 1, 2, \dots, n$, является выборкой из некоторой генеральной совокупности с функцией плотности распределения $f(x_i, \theta)$. При этом первые моменты функции $f(x_i, \theta)$ известны и конечны. В данном случае для построения минимизируемого функционала F необходимо знать вид функции плотности распределения $f(x_i, \theta)$ или ее моменты.

Для определения моментов в [102] предлагается использовать разложение функции $f(x_i, \theta)$ в ряд Тейлора в окрестности точки x_{0i} . Тогда первый и второй моменты функции плотности распределения $f(x_{0i}, \theta)$ находят по формулам [102]:

$$M(y_i | x_{0i}) \approx f(x_i, \theta),$$

$$\sigma^2(y_i | x_{0i}) \approx \sigma^2(y_i) + \left[\frac{\partial f(x_i, \theta)}{\partial x_i} \Big|_{x_i=x_{0i}} \right]^2 \sigma^2(x_i).$$

Итерационный процесс нахождения оценок вектора параметров θ строится следующим образом.

1. Составляется функционал

$$F_0 = \sum_{i=1}^n [y_i - M(y_i | x_{0i})]^2 w_i^0,$$

где $w_i^0 = \sigma^{-2}(y_i)$, и определяется оценка вектора параметров θ^0 , при которой достигается минимум функционала F_0 , т. е. решается регрессионная задача.

2. Определяются величины $w_i^1 = \sigma^2(y_i | x_{0i})$.

3. Составляется функционал

$$F_1 = \sum_{i=1}^n [y_i - M(y_i | x_{0i})]^2 w_i^1 \quad (7.25)$$

и определяется оценка вектора параметров θ^1 .

Операции 2 и 3 повторяют до тех пор, пока относительные изменения параметров на соседних итерациях не будут меньше некоторой малой величины γ :

$$\max \left| \frac{\theta_j^s - \theta_j^{s-1}}{\theta_j^s} \right| \leq \gamma, \quad j = 1, 2, \dots, m.$$

Рассмотрим применение данного метода для линейной функции $\eta_i = ax_i + b$:

$$M(y_i | x_{0i}) \approx ax_{0i} + b, \quad \sigma^2(y_i | x_{0i}) \approx \sigma^2(y_i) + a^2 \sigma^2(x_i).$$

Функционал (7.25) примет вид

$$F_1 = \sum_{i=1}^n \frac{(y_i - ax_i - b)^2}{\sigma^2(y_i) + a^2 \sigma^2(x_i)}.$$

Для нелинейного случая, например для функции плотности, имеющей нормальное распределение, получим

$$M(y_i | x_{0i}) \approx \frac{1}{\sqrt{2\pi} \cdot \theta_2} \exp \left\{ -\frac{(x_{0i} - \theta_1)^2}{2\theta_2^2} \right\},$$

$$\sigma^2(y_i | x_{0i}) \approx \sigma^2(y_i) + \left[\frac{1}{\sqrt{2\pi} \cdot \theta_2} \exp \left\{ -\frac{(x_{0i} - \theta_1)^2}{2\theta_2^2} \right\} \frac{(\theta_1 - x_{0i})}{\theta_2^2} \right]^2 \sigma^2(x_i).$$

Подставив эти выражения в функционал (7.25), определим оценки параметров θ_1 и θ_2 .

В статье [102] приведены асимптотические свойства получаемых оценок. В [30, 75, 76, 102] исследуются также статистические

свойства оценок и вопрос о сходимости итерационного алгоритма получения оценок.

IV. В работе [95] предложен метод, позволяющий учитывать ошибки переменных x_i для линейных моделей.

Допустим, что задана модель с известными дисперсиями ошибок σ_ε^2 и σ_δ^2 :

$$y_i = a\xi_i + b + \varepsilon_i, \quad x_i = \xi_i + \delta_i, \quad i = 1, 2, \dots, n.$$

Традиционными методами, например МНК, определяют оценки параметров a и b модели, считая, что x_i — детерминированные величины. Следовательно, теперь мы имеем модель с известными параметрами:

$$y_i - \hat{b} = \hat{a}\xi_i + \varepsilon_i, \quad x_i = \xi_i + \delta_i, \quad i = 1, 2, \dots, n,$$

или, в матричном виде,

$$\begin{pmatrix} y_i - \hat{b} \\ x_i \end{pmatrix} = \begin{pmatrix} \hat{a} \\ 1 \end{pmatrix} \xi_i + \begin{pmatrix} \varepsilon_i \\ \delta_i \end{pmatrix}.$$

Получили регрессионную модель, где ξ_i — неизвестный параметр, подлежащий оцениванию. Оценки $\hat{\xi}_i$ определяют из выражения

$$\hat{\xi}_i = ((\hat{a}, 1)\Sigma^{-1}(\hat{a}, 1)^T)^{-1}(\hat{a}, 1)\Sigma^{-1}(y_i - \hat{b}, x_i)^T,$$

где $\Sigma = \text{diag}(\sigma_\varepsilon^2, \sigma_\delta^2)$.

Последнее выражение можно преобразовать следующим образом:

$$\hat{\xi}_i = (\hat{a}^2\sigma_\delta^2 + \sigma_\varepsilon^2)^{-1}(\sigma_\delta^2(y_i - \hat{b})\hat{a} + \sigma_\varepsilon^2 x_i).$$

Дисперсии оценок определяют по формулам

$$D(\hat{\xi}_i) = ((\hat{a}, 1)\Sigma^{-1}(\hat{a}, 1)^T)^{-1},$$

или

$$D(\hat{\xi}_i) = (\hat{a}^2\sigma_\delta^2 + \sigma_\varepsilon^2)^{-1}.$$

Далее уточняют оценки параметров a и b при новых значениях аргументов ξ_i . Затем снова определяют оценки $\hat{\xi}_i$. Итерационный процесс продолжается до тех пор, пока относительные изменения параметров на соседних итерациях не будут меньше некоторой малой величины γ :

$$\max \left| \frac{\theta_j^s - \theta_j^{s-1}}{\theta_j^s} \right| \leq \gamma, \quad j = 1, 2.$$

Для линейных моделей мы получили функционал и формулу для расчета значений ξ_i , аналогичную приведенной в § 7.9. Таким образом, в качестве единого подхода к задаче нахождения оценок параметров моделей с учетом погрешностей в значениях функции и аргументов могут быть использованы методы конфлюэнтного анализа, описанные в [20–24, 40].

§ 7.11. О единственности оценок параметров. Состоятельность оценок и алгоритм их получения

Функционал (7.19) является самым простым, и даже он нелинеен по θ в случае оценки параметров θ уравнения прямой при одновременном учете погрешностей и в значениях функции, и в значениях аргумента. Поэтому возникает вопрос о единственности оценок искомых параметров, который тесно связан с выбором нулевого приближения оценок. Естественным образом рассмотреть в качестве нулевого приближения решение регрессионной задачи (например, по методу наименьших квадратов).

Из условия (7.12) следует, что для одной и той же задачи минимальное значение функционала в методе наименьших квадратов больше минимального значения функционала (7.10), поскольку условие (7.12) является условием ортогональности вектора касательной к функции $\eta = f(\xi, \theta)$ в точке $(\hat{\xi}_{i1}, \dots, \hat{\xi}_{im}, \hat{\eta}_i)$, принимаемой за «истинную», и вектора, проведенного через исходную точку $(x_{i1}, \dots, x_{im}, y_i)$ и точку, принимаемую за «истинную». Таким образом, минимизация функционала (7.10) при условии (7.12) приводит к минимизации суммы квадратов «наикратчайших расстояний» от точки $(x_{i1}, \dots, x_{im}, y_i)$ до кривой, тогда как в методе наименьших квадратов минимизируется сумма квадратов отклонений при фиксированных значениях x_{ij} .

Для конкретных видов функции $\eta = f(\xi, \theta)$ в работах [21, 23, 40] получены условия существования единственного решения. Покажем существование единственного решения для линейных функций

$$\eta = \sum_{j=1}^m \theta_j x_j + \theta_0$$

в предположении, что:

- 1) погрешности ε и δ статистически независимы и подчиняются нормальному закону распределения;
- 2) за нулевое приближение взято решение регрессионной задачи [21, 23].

Эти ограничения обусловлены тем, что с помощью преобразований координат функция Гаусса, описывающая нормальный закон распределения, сводится к уравнению прямой линии и оценки параметров этой прямой будут однозначно связаны с оценками параметров функции Гаусса.

Теорема 1. Если погрешности ε и δ измеренных случайных величин x и y статистически независимы и распределены по нормальному закону с известными числовыми характеристиками функции плотности распределения вероятностей и если в качестве нулевого приближения принято решение соответствующей регрессионной задачи, то достаточным условием единственности оценок параметров линейной функции

$$\eta = \theta_0 + \sum_{j=1}^m \theta_j \xi_j$$

является выполнение неравенства $\bar{x}^2 > (1 - m/n)\hat{\sigma}^2(y)$. Здесь $\hat{\sigma}^2(y)$ — оценка дисперсии значений y_i при равнооточных измерениях.

Доказательство. Для простоты формул величины $\sigma(x_{ij})$ и $\sigma(y_i)$ приняты равными единице. Функционал (7.10) с учетом условия (7.12) для линейных функций имеет вид

$$F = \frac{1}{2} \frac{\sum_{i=1}^n \left(y_i - \theta_0 - \sum_{j=1}^m \theta_j x_{ij} \right)^2}{1 + \sum_{j=1}^m \theta_j^2}. \quad (7.26)$$

При фиксированных θ_j , $j = 1, 2, \dots, m$, функционал (7.26) относительно переменной θ_0 есть парабола и любое начальное значение θ_0^0 приведет к единственной оценке. Для переменных θ_j , $j = 1, 2, \dots, m$, функционал (7.26) монотонно возрастает с увеличением разности $\theta_j - \hat{\theta}_j$ от точки минимума вправо до горизонталь-

ной асимптоты, равной $\sum_{j=1}^m \sum_{i=1}^n x_{ij}^2$, и влево — до точки максимума, а

затем монотонно убывает до той же горизонтальной асимптоты. Для доказательства теоремы достаточно указать, когда соответствующий регрессионный функционал Φ_p станет меньше

$\sum_{j=1}^m \sum_{i=1}^n x_{ij}^2$, в этом случае начальное приближение попадает в об-

ласть выпуклости функционала (7.26). В точке минимума имеем

$\Phi_p = (n - m)\hat{\sigma}^2(y)$. Для единственности оценок параметров θ_j должно выполняться условие

$$\Phi_p = (n - m)\hat{\sigma}^2(y) < \sum_{j=1}^m \sum_{i=1}^n x_{ij}^2 = \bar{x}^2 n$$

или

$$\bar{x}^2 > \left(1 - \frac{m}{n}\right) \hat{\sigma}^2(y).$$

Теорема доказана.

В рассматриваемом случае дисперсии значений y_i известны и могут быть использованы для проверки достаточного условия.

Можно сформулировать и более строгие условия, когда значение Φ_p в точке минимума будет меньше $\sum_{j=1}^m \sum_{i=1}^n x_{ij}^2$ и в то же время

точка минимума функционала Φ_p будет принадлежать строго выпуклой области функционала (7.26). Для этого следует проверить любое достаточное условие строгой выпуклости функционала (7.26) в точке минимума регрессионного функционала.

Условия строгой выпуклости функционала F , а тем самым и единственность его точки минимума [42, 46] устанавливаем согласно теоремам [46], по которым достаточным условием строгой выпуклости функционала F является положительная определен-

ность матрицы $M = \left\| \frac{\partial^2 F}{\partial \theta_l \partial \theta_r} \right\|$, а для того чтобы матрица M была

положительно определена, необходимо и достаточно, чтобы каждый из определителей

$$\det \left\| \frac{\partial^2 F}{\partial \theta_0^2} \right\|, \quad \det \left\| \frac{\partial^2 F}{\partial \theta_l \partial \theta_r} \right\|, \quad l, r = 1, 2, \dots, n,$$

был положителен.

Сформулируем достаточные условия того, что регрессионное решение принадлежит области строгой выпуклости функционала (7.26) в случае оценки свободных параметров прямой линии.

Теорема 2. *Достаточным условием того, что регрессионное решение при оценке параметров уравнения прямой линии $\eta = \theta_0 + \theta_1 \xi$ принадлежит строго выпуклой области функционала (7.26), является выполнение неравенства*

$$\sum_i x_i^2 - \frac{1}{n} \left(\sum_i x_i \right)^2 > (n-2) \hat{\sigma}^2(y).$$

Доказательство. Действительно, определитель $\det \left\| \frac{\partial^2 F}{\partial \theta_0^2} \right\|$ положителен всегда. В точке минимума регрессионного функционала, когда $\sum_{i=1}^n (y_i - \theta_0 - \theta_1 x_i) = 0$ и $\sum_{i=1}^n (y_i - \theta_0 - \theta_1 x_i) x_i = 0$, имеем

$$\begin{aligned} \det \left\| \frac{\partial^2 F}{\partial \theta_0 \partial \theta_1} \right\| &= (1 + \theta_1^2) \left[n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 \right] + \\ &+ n(3\theta_1^2 - 1) \sum_{i=1}^n (y_i - \theta_0 - \theta_1 x_i)^2. \end{aligned}$$

Это выражение положительно для любого θ_0 и $|\theta_1| > 1/\sqrt{3}$. Для $|\theta_1| \leq 1/\sqrt{3}$ функционал будет выпуклым, если

$$(1 + \theta_1^2) \left[n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 \right] > \left| n(3\theta_1^2 - 1) \sum_{i=1}^n (y_i - \theta_0 - \theta_1 x_i)^2 \right|.$$

Заменим величину $\sum_{i=1}^n (y_i - \theta_0 - \theta_1 x_i)^2$ на $(n-2) \hat{\sigma}^2(y)$. Тогда получим

$$n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 > \frac{n|(n-2)(3\theta_1^2 - 1)| \hat{\sigma}^2(y)}{1 + \theta_1^2}.$$

Правая часть последнего условия имеет максимум при $\theta_1 = 0$, т. е. для любого θ_1 условие того, что регрессионное решение принадлежит области выпуклости функционала (7.26), имеет вид

$$\sum_{i=1}^n x_i^2 - \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2 > (n-2) \hat{\sigma}^2(y).$$

Теорема доказана.

Условия, обеспечивающие попадание регрессионного решения в выпуклую область функционала (7.10), для других функций не сложно проверить на компьютере, но достаточно сложно выразить в явном виде. Методом статистических испытаний было показано, что при оценке параметров нормального закона распределения (от одномерного до пятимерного) без применения каких-либо преобразований координат итерационный процесс сходиллся, если нулевое приближение отличалось от истинного значения не более чем на 40% при относительных погрешностях наблюдаемых значений признаков, составляющих приблизительно 10 %. Такую точность получения нулевых приближений обеспечивает регрессионный анализ.

Для функций $\eta = f(\xi, \theta)$, линейных по вектору параметров θ , единственность оценки и сходимость нулевого приближения к оцениваемым параметрам при нормально распределенных исходных данных определяются только методом нахождения оценки $\hat{\xi}$, так как при фиксированном векторе оценок $\hat{\xi}$ система уравнений (7.11) является системой линейных алгебраических уравнений с детерминантом, не равным нулю. Для этих функций из условия (7.12), в принципе, могут быть найдены несколько наборов значений $\{\hat{\xi}\}$, каждый из которых принадлежит области возможных значений при заданных значениях $\{x_i\}$. В такой ситуации нет оснований отдавать предпочтение какому-либо одному набору значений $\{\hat{\xi}\}$ и, строго говоря, решение будет не единственно.

Рассмотрим вопрос о состоятельности вектора оценок параметров θ в конфлюэнтном анализе, если оценка $\hat{\xi}$ находится в силу условия (7.12).

Теорема 3. Если оценки параметров прямой $\eta = \theta_0 + \theta_1 \xi$ находятся как координаты точки минимума функционала (7.26), то они определяются по формулам

$$\hat{\theta}_0 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{\theta}_1 x_i),$$

$$\hat{\theta}_1 = \max \{ \hat{\theta}_{11}; \hat{\theta}_{12} \},$$

где

$$\begin{aligned}\hat{\theta}_{11,12} &= \frac{1}{2c} \left((w-d) \pm \sqrt{(d-w)^2 + 4c^2} \right), \\ c &= \sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n y_i \sum_{i=1}^n x_i, \quad w = \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2, \\ d &= \sum_{i=1}^n x_i^2 - \sum_{i=1}^n y_i^2 + \frac{1}{n} \left(\sum_{i=1}^n y_i \right)^2.\end{aligned}$$

Доказательство. Решим систему алгебраических уравнений, обеспечивающую необходимое условие экстремума функционала (7.26):

$$\frac{\partial F}{\partial \theta_0} = 0, \quad \frac{\partial F}{\partial \theta_1} = 0.$$

Из первого уравнения следует, что выполнено равенство

$$\hat{\theta}_0 = \frac{1}{n} \sum_{i=1}^n (y_i - \theta_1 x_i).$$

Исключив из приведенной системы уравнений величину θ_0 , получим для оценки $\hat{\theta}_1$ квадратное уравнение

$$c \hat{\theta}_1^2 + (d - w) \hat{\theta}_1 - c = 0,$$

корни которого и есть $\hat{\theta}_{11}$ и $\hat{\theta}_{12}$. В данном случае нет необходимости применять достаточные признаки экстремума, чтобы определить, какой корень отвечает точке минимума функционала (7.26). Из сечений функционала F видно, что меньший корень отвечает точке максимума, а больший — точке минимума. Теорема доказана.

Теорема 4. Если оценки параметров прямой $\eta = \theta_0 + \theta_1 \xi$ находятся как координаты точки минимума функционала (7.26), то они состоятельны.

Доказательство. Докажем состоятельность оценки $\hat{\theta}_1$. Обозначим s_x^2 , s_y^2 и s_{xy} соответственно выборочные дисперсии наборов $\{x_i\}$, $\{y_i\}$ и выборочную ковариацию величин x_i и y_i . Тогда

$$w - d = \frac{1}{n} \left(\sum_{i=1}^n x_i \right)^2 - \sum_{i=1}^n x_i^2 + \sum_{i=1}^n y_i^2 - \frac{1}{n} \left(\sum_{i=1}^n y_i \right)^2 = s_y^2 - s_x^2,$$

$$c = \sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n y_i \sum_{i=1}^n x_i = s_{xy}.$$

В соответствии с результатами работы [40, гл. 10] выборочные дисперсии s_x^2 , s_y^2 и выборочная ковариация s_{xy} сходятся по вероятности к своим математическим ожиданиям:

$$s_x^2 \rightarrow s_\xi^2 + \sigma^2(x) = s_\xi^2 + 1, \quad n \rightarrow \infty,$$

$$s_y^2 \rightarrow \theta_1^2 s_\xi^2 + \sigma^2(y) = \theta_1^2 s_\xi^2 + 1, \quad n \rightarrow \infty,$$

$$s_{xy}^2 \rightarrow \theta_1 s_\xi^2, \quad n \rightarrow \infty,$$

где s_ξ^2 — дисперсия ненаблюдаемой величины ξ . Теперь из квадратного уравнения $c\hat{\theta}_1^2 + (d - w)\hat{\theta}_1 - c = 0$ имеем

$$\hat{\theta}_{11,12} = \frac{s_\xi^2(\theta_1^2 - 1) \pm \sqrt{s_\xi^4(\theta_1^2 - 1)^2 + 4s_\xi^4\theta_1^2}}{2\theta_1 s_\xi^2} = \frac{(\theta_1^2 - 1) \pm (\theta_1^2 + 1)}{2\theta_1},$$

$$\hat{\theta}_{11} = \theta_1, \quad \hat{\theta}_{12} = -\frac{1}{\theta_1}.$$

Отсюда следует, что оценка $\hat{\theta}_{11}$, соответствующая точке минимума функционала (7.26), состоятельная. Докажем теперь состоятельность оценки $\hat{\theta}_0$:

$$\hat{\theta}_0 = \frac{1}{n} \sum_{i=1}^n y_i - \hat{\theta}_1 \frac{1}{n} \sum_{i=1}^n x_i = \bar{y} - \hat{\theta}_1 \bar{x} \rightarrow \bar{y} - \theta_1 \bar{x}.$$

Согласно [40] имеем $\bar{y} \rightarrow \theta_0 + \theta_1 \xi$, $x \rightarrow \xi$, т. е. $\hat{\theta}_0 = \theta_0 + \theta_1 \xi - \theta_1 \xi = \theta_0$ — оценка состоятельная.

Если в качестве $\hat{\xi}_i$ берут значения x_i , то оценка $\hat{\theta}_1$ не будет состоятельной. Рассмотрим случай, когда $\theta_0 = 0$. Тогда

$$\hat{\theta}_1 = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} = \frac{s_{xy}}{s_x^2};$$

значит, при $n \rightarrow \infty$ находим, что оценка

$$\frac{s_{xy}}{s_x^2} = \frac{\theta_1 s_\xi^2}{s_\xi^2 + 1} = \theta_1 \left(1 - \frac{1}{s_\xi^2 + 1} \right)$$

несостоятельна.

Следствие 1. Если оценки параметров θ из уравнения $\eta = \theta_0 + \sum_{j=1}^m \theta_j \xi_j$ находятся как координаты точки минимума функционала (7.10) и оценки $\hat{\xi}$ находятся из условия (7.12), то полученные оценки $\hat{\theta}$ будут состоятельными.

Следствие 2. В условиях теоремы 4 нулевое приближение может отличаться до 100 % от истинного значения оценок искомым параметров.

Действительно, согласно теореме 4 разность оценок параметра θ_1 составляет

$$\hat{\theta}_{11} - \hat{\theta}_{12} = \theta_1 + \frac{1}{\theta_1} > \theta_1.$$

Учитывая свойства функционала F и получающихся оценок, задачу минимизации функционала F при условии (7.12) решают по следующей схеме.

Шаг 1. Задают начальные значения переменных ξ (результаты наблюдений x) и определяют минимум по θ функционала F при фиксированных ξ . В результате получают первое приближение для оценок искомым параметров θ .

Шаг 2. Точные значения переменных пересчитывают с учетом ограничения (7.12) при полученных приближениях для оценок параметров θ . Для уменьшения числа итераций полезно проверить условие принадлежности нового значения ξ_{ij} области возможных значений при заданном x_{ij} .

Шаг 3. Минимизируют функционал F по θ при новых точных значениях переменных ξ .

Шаги 2 и 3 выполняют поочередно до тех пор, пока не будет выполнено хотя бы одно из трех условий:

1) на очередном шаге значение функционала (7.10) меньше заданного числа ε_1 ;

2) на соседних итерациях значения функционала F и оценок параметров θ отличаются несущественно, т. е.

$$\left| \frac{F_v - F_{v+1}}{F_v} \right| \leq \delta_1, \quad \max \left| \frac{\theta_l^v - \theta_l^{v+1}}{\theta_l^v} \right| \leq \delta_2, \quad l = 1, 2, \dots, k,$$

где δ_1 и δ_2 — заданные числа;

3) исчерпан заданный лимит итераций.

Шаг 4. После нахождения точечных оценок $\hat{\theta}$ определяют дисперсии оценок.

Из приведенного алгоритма следует, что на первой итерации получается регрессионное решение (иными словами, решение, полученное любым из применявшихся ранее традиционных методов). На последующих шагах это решение уточняется. Такой переход от традиционных методов к предлагаемому важен в практических приложениях.

§ 7.12. Оценка параметров многомерной линейной модели

Линейные модели широко применяются в статистических задачах принятия решений, например при нахождении разделяющих поверхностей в задачах распознавания образов.

Рассмотрим применение описанного метода к решению задачи определения оценок параметров линейной модели

$$\eta = f(\xi, \theta) = \sum_{i=1}^n \theta_i \xi_i$$

и к решению систем линейных алгебраических уравнений с погрешностями в матрице системы и в правой части уравнений, если ошибки измерений — независимые нормально распределенные случайные величины с нулевыми средними значениями и известными дисперсиями $\sigma^2(x_{ij})$ и $\sigma^2(y_j)$.

В этом случае функционал (7.10) будет иметь вид

$$F = \frac{1}{2} \sum_{j=1}^m \left(\sum_{i=1}^n \frac{(x_{ij} - \xi_{ij})^2}{\sigma^2(x_{ij})} + \frac{\left(y_j - \sum_{i=1}^n \theta_i \xi_{ij} \right)^2}{\sigma^2(y_j)} \right), \quad (7.27)$$

а ограничение $\hat{\xi}_i \in D_i$ можно записать следующим образом:

$$|x_{ij} - \hat{\xi}_{ij}| \leq 3\sigma(x_{ij}). \quad (7.28)$$

Здесь для упрощения вида функционала (7.27) ошибки измерений считаются статистически независимыми.

Задача минимизации по θ функционала (7.27) при фиксированных значениях ξ является обычной задачей регрессионного анализа. Исследования показали, что для обеспечения минимального времени счета и большей точности лучше всего на первом шаге решать систему линейных уравнений при $\xi_{ij} = x_{ij}$ методом Гаусса с выбором максимального элемента, а затем непосредственно минимизировать по θ функционал (7.27) методом сопряженных градиентов. Учитывая вид функционала (7.27), задачу минимизации по θ легко свести к задаче поиска минимума квадратичной формы

$$G(\theta) = \frac{1}{2} \theta^T A \theta + a^T \theta,$$

которая и решалась методом сопряженных градиентов. Здесь θ — вектор искомых оценок параметров; A — матрица квадратичной формы, элементы которой вычисляются по формуле

$$A_{rp} = \sum_{j=1}^m \frac{1}{\sigma^2(y_j)} x_{rj} x_{pj}, \quad r=1, 2, \dots, n, \quad p=1, 2, \dots, n,$$

x_{rj} — значение r -й переменной, полученное в j -м измерении;

a^T — вектор, компоненты которого имеют вид

$$a_r^T = - \sum_{j=1}^m \frac{y_j}{\sigma^2(y_j)} x_{rj}, \quad r=1, 2, \dots, n.$$

Пересчет точных значений ξ на основании условий (7.21) сводится к решению m несвязанных между собой систем из n линейных уравнений вида

$$\sum_{r=1}^n \frac{\theta_r \theta_p}{\sigma^2(y_j)} \xi_{rj} + \frac{\xi_{pj}}{\sigma^2(x_{pj})} = \frac{x_{pj}}{\sigma^2(x_{pj})} + \frac{\theta_p y_j}{\sigma^2(y_j)},$$

$$p=1, 2, \dots, n, \quad j=1, 2, \dots, m.$$

Полученные новые значения $\hat{\xi}$ должны удовлетворять условию (7.28). В противном случае те ξ_{ij} , которые выходят за указанные границы, заменяют на значение ближайшей граничной точки. В связи с этим иногда можно ожидать увеличение функционала (7.27) при новых точных значениях вектора переменных ξ по сравнению с предыдущим шагом итерационного процесса. Это

приводит к снижению скорости сходимости процесса и даже к возникновению колебаний. Для устранения таких нежелательных последствий после пересчета ξ те наборы $\{\xi_j\}$, на которых произошло увеличение соответствующих слагаемых функционала

$$F_j = \sum_{i=1}^n \frac{(x_{ij} - \xi_{ij})^2}{\sigma^2(x_{ij})} + \frac{\left(y_j - \sum_{i=1}^n \theta_i \xi_{ij}\right)^2}{\sigma^2(y_j)}$$

по сравнению с предыдущей итерацией, заменяются значениями ξ , полученными на предыдущем шаге.

Применение описанного метода позволило в большинстве случаев значительно увеличить скорость сходимости процесса (больше чем в 3 раза). Когда же первоначальный процесс не сходился (возникали колебания), решение достигалось за 12–17 итераций. Элементы дисперсионной матрицы ошибок (матрицы рассеяния) для оценок искомых параметров подсчитываются как элементы матрицы, обратной матрице N с элементами

$$N_{ij} = - \frac{\partial^2 F}{\partial \theta_i \partial \theta_j} \bigg|_{\theta = \hat{\theta}},$$

где $\hat{\theta}$ — полученная оценка параметров θ .

Описанную выше задачу можно интерпретировать как задачу решения переопределенной системы $\eta = \xi\theta$ из m линейных уравнений с n переменными, $m > n$, когда вместо значений η и ξ имеем соответственно столбец $y = (y_1, \dots, y_m)^T$ значений функции, $(n \times m)$ -матрицу X , элементы которой x_{ij} суть значения i -й переменной в j -м измерении, столбец $\theta = (\theta_1, \dots, \theta_n)^T$ неизвестных (искомых) параметров.

§ 7.13. Оценка параметров полиномиальной зависимости

Полиномом можно приближенно описать любую функцию.

Рассмотрим задачу оценки параметров полинома. Требуется определить оценку $\hat{\theta}$ коэффициентов полинома степени m

$$\eta = \sum_{i=0}^m \theta_i \xi^i$$

по известным значениям аргумента и функции, содержащим погрешности измерений.

Пусть исходные данные имеют вид

$$y_i = \eta_i + \varepsilon_i, \quad x_i = \xi_i + \delta_i, \quad i = 1, 2, \dots, n.$$

Предположим, что ошибки измерений ε и δ — нормально распределенные случайные величины с нулевыми средними значениями, известными дисперсиями (соответственно $\sigma^2(x_i)$ и $\sigma^2(y_i)$) и известными коэффициентами корреляции ρ_i между измеренными значениями x_i и y_i . Тогда искомые оценки коэффициентов $\hat{\theta}$ получают путем минимизации функционала

$$F = \frac{1}{2} \sum_{i=1}^n \frac{1}{1 - \rho_i^2} \times \left[\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} + \frac{\left(y_i - \sum_{l=0}^m \theta_l \xi_i^l \right)^2}{\sigma^2(y_i)} - 2\rho_i \frac{(x_i - \xi_i)(y_i - \eta_i)}{\sigma(x_i)\sigma(y_i)} \right] \quad (7.29)$$

при условиях

$$\left. \frac{\partial F}{\partial \xi_i} \right|_{\xi_i = \hat{\xi}_i} = 0, \quad |x_i - \hat{\xi}_i| \leq 3\sigma(x_i). \quad (7.30)$$

Пересчет значений ξ_i согласно условию (7.30) сводится к решению системы из n нелинейных уравнений вида

$$\begin{aligned} \frac{x_i - \xi_i}{\sigma^2(x_i)} - \rho_i \frac{y_i - \sum_{l=0}^m \theta_l \xi_i^l + (x_i - \xi_i) \sum_{l=0}^m l \theta_l \xi_i^{l-1}}{\sigma(x_i)\sigma(y_i)} + \\ + \frac{y_i - \sum_{l=0}^m \theta_l \xi_i^l}{\sigma^2(y_i)} \sum_{l=1}^m l \theta_l \xi_i^{l-1} = 0, \quad i = 1, 2, \dots, n, \end{aligned} \quad (7.31)$$

на отрезке $|x_i - \xi_i| \leq 3\sigma(x_i)$.

В процессе решений системы уравнений (7.31) может оказаться, что:

1) корень одного из уравнений принадлежит отрезку $|x_i - \xi_i| \leq 3\sigma(x_i)$ (значение корня находится методом хорд);

2) корень одного из уравнений не принадлежит отрезку $|x_i - \xi_i| \leq 3\sigma(x_i)$; тогда с помощью алгоритма поиска глобального минимума функции находится наименьшее значение функции

$$\begin{aligned} \varphi(\xi_i) = & \frac{x_i - \xi_i}{\sigma^2(x_i)} - \rho_i \frac{y_i - \sum_{l=0}^m \theta_l \xi_i^l + (x_i - \xi_i) \sum_{l=1}^m l \theta_l \xi_i^{l-1}}{\sigma(x_i) \sigma(y_i)} + \\ & + \frac{y_i - \sum_{l=0}^m \theta_l \xi_i^l}{\sigma^2(y_i)} \sum_{l=1}^m l \theta_l \xi_i^{l-1}, \end{aligned}$$

принадлежащее отрезку $|x_i - \xi_i| \leq 3\sigma(x_i)$.

Поставленная задача минимизации функционала (7.29) решается по итерационной схеме, изложенной в § 7.11. Дисперсии полученных оценок находятся с помощью матрицы вторых производных функционала (7.29). Процедура нахождения оценок и их дисперсий аналогична ранее описанной в § 7.2

§ 7.14. Оценка значений параметров в сигноме

Рассмотрим задачу оценки вектора параметров θ сигнома (самый общий вид многомерного полинома)

$$\eta = f(\xi, \theta) = \sum_{i=1}^k \theta_i \prod_{j=1}^m \xi_j^{\alpha_{ij}},$$

где α_{ij} — показатель степени аргумента сигнома, при условии, что все ошибки измерений — независимые нормально распределенные случайные величины с нулевыми средними и известными дисперсиями $\sigma^2(x_{ij})$ и $\sigma^2(y_i)$. Функционал (7.10) будет иметь вид

$$F = \frac{1}{2} \sum_{i=1}^n \left(\sum_{j=1}^m \frac{(x_{ij} - \xi_{ij})^2}{\sigma^2(x_{ij})} + \frac{(y_i - f(\xi_i, \theta))^2}{\sigma^2(y_i)} \right). \quad (7.32)$$

Ограничение $\xi_i \in D_i$ можно записать следующим образом:

$$|x_{ij} - \xi_{ij}| \leq 3\sigma(x_{ij}).$$

Чтобы упростить вид функционала (7.32), задачу минимизации по θ сводим к задаче поиска минимума квадратичной формы

$$G(\theta) = \frac{1}{2} \theta^T A \theta + a^T \theta,$$

где элементы матрицы квадратичной формы A имеют вид

$$A_{rp} = \sum_{i=1}^n \frac{1}{\sigma^2(y_i)} \prod_{j=1}^n x_{ij}^{\alpha_{ir} + \alpha_{jp}}, \quad r = 1, 2, \dots, k, \quad p = 1, 2, \dots, k,$$

x_{ij} — значение i -й переменной, полученной в j -м измерении;

a^T — вектор с компонентами

$$a_r^T = - \sum_{i=1}^n \frac{y_i}{\sigma^2(y_i)} \prod_{j=1}^n x_{ij}^{\alpha_{jr}}, \quad r = 1, 2, \dots, k.$$

Точка минимума формы $G(\theta)$ определяется методом сопряженных градиентов.

Пересчет точных значений аргументов ξ_{ij} с учетом условий (7.12) сводится к решению n несвязанных между собой систем из m нелинейных уравнений вида

$$\frac{x_{ij} - \xi_{ij}}{\sigma^2(x_{ij})} + \frac{y_i - f(\xi_i, \theta)}{\sigma^2(y_i)} \frac{\partial f}{\partial \xi_{ij}} = 0, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m. \quad (7.33)$$

Здесь

$$\frac{\partial f}{\partial \xi_{ij}} = \sum_{l=1}^k \theta_l \alpha_{ji} x_{il}^{\alpha_{1l}} \dots x_{ij}^{\alpha_{jl}} \dots x_{im}^{\alpha_{ml}},$$

$\xi_i = (\xi_{i1}, \dots, \xi_{im})$ — искомые значения переменных в i -м измерении.

Система уравнений (7.33) решалась методом итераций с использованием линеаризации исходных уравнений на каждом шаге [25, 42]. Такое упрощение вычислений возможно потому, что в силу условия (7.12) все расчеты ведутся в небольшой области пространства переменных. Таким образом, на каждой итерации в процессе решения системы (7.33) решаются системы линейных уравнений

$$\sum_{r=1}^m \frac{\partial \varphi_j(\xi_i)}{\partial \xi_{ir}} \bigg|_{\xi_i = \xi_i^0} (\xi_{ir} - \xi_{ir}^0) = -\varphi_j(\xi_i^0),$$

$$i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m,$$

где $\varphi_j(\xi_i^0)$ — значение левой части j -го уравнения системы (7.33) в точке ξ_i^0 (точке разложения);

$$\frac{\partial \varphi_j(\xi_i)}{\partial \xi_{ir}} = \frac{y_i - f(\xi_i, \theta)}{\sigma^2(y_i)} f''_{\xi_{ij}\xi_{ir}} - \frac{f'_{\xi_{ij}} f'_{\xi_{ir}}}{\sigma^2(y_i)}, \quad r \neq j,$$

$$\frac{\partial \varphi_j(\xi_i)}{\partial \xi_{ir}} = \frac{y_i - f(\xi_i, \theta)}{\sigma^2(y_i)} f''_{\xi_{ij}^2} - \frac{(f'_{\xi_{ij}})^2}{\sigma^2(y_i)} - \frac{1}{\sigma^2(x_{ij})}, \quad r = j.$$

Здесь

$$f''_{\xi_{ij}\xi_{ir}} = \sum_{l=1}^k \theta_l \alpha_{jl} \alpha_{rl} x_{il}^{\alpha_{1l}} \dots x_{ij}^{\alpha_{jl,l-1}} \dots x_{ir}^{\alpha_{rl,l-1}} \dots x_{im}^{\alpha_{ml}},$$

$$f''_{\xi_{ij}^2} = \sum_{l=1}^k \theta_l \alpha_{jl} (\alpha_{jl} - 1) x_{il}^{\alpha_{1l}} \dots x_{ij}^{\alpha_{jl,l-2}} \dots x_{im}^{\alpha_{ml}}.$$

В связи с тем, что уравнения в системе (7.33) замещались приближенными выражениями, в некоторых случаях можно также ожидать увеличения значений функционала (7.32) при новых точных значениях переменных ξ_{ij} по сравнению с предыдущим шагом итерационного процесса, что приведет к снижению скорости сходимости процесса и даже к возникновению колебаний. Для устранения этих нежелательных последствий после пересчета ξ_{ij} те значения ξ_{ij} , на которых произошло увеличение соответствующих слагаемых функционала

$$F_i = \sum_{j=1}^m \frac{(x_{ij} - \xi_{ij})^2}{\sigma^2(x_{ij})} + \frac{(y_i - f(\xi_i, \theta))^2}{\sigma^2(y_i)}$$

по сравнению с предыдущей итерацией, необходимо заменить значениями с предыдущего шага.

Элементы ковариационной матрицы ошибок (матрицы рассеивания) для оценок вектора искомых параметров θ подсчитываются как элементы матрицы, обратной матрице N с элементами

$$N_{rp} = - \left. \frac{\partial^2 F}{\partial \theta_r \partial \theta_p} \right|_{\theta=\hat{\theta}}, \quad r = 1, 2, \dots, k, \quad p = 1, 2, \dots, k,$$

где $\hat{\theta}$ — полученная оценка вектора параметров θ .

Многочисленные вычислительные эксперименты позволяют сделать вывод о состоятельности рассматриваемых оценок.

Пример. Найти оценки вектора параметров θ в модели $\eta = \theta_1 \xi_1 \xi_2^3 + \theta_2 \xi_1^2 \xi_2^2 + \theta_3 \xi_1^3$.

Предварительно были взяты значения: $\theta_1 = 1$, $\theta_2 = -1$, $\theta_3 = 2$, $\xi_1 = (1, 1, 2, 2, 3, 3)^T$, $\xi_2 = (1, 2, 1, 2, 1, 2)^T$, и вычислены значения $\eta = (2, 6, 14, 16, 48, 42)^T$.

С помощью датчика случайных чисел на точные значения η и ξ налагались помехи с математическим ожиданием, равным нулю, и дисперсиями, равными 0,020 для ξ_1 , 0,015 для ξ_2 , 0,1 для η . Получены следующие реализации:

$$y = (2,2057; 6,1090; 14,0602; 15,7987; 47,8996; 42,1214)^T,$$

$$x_1 = (1,0070; 0,9604; 2,0123; 2,0427; 3,0027; 2,9781)^T,$$

$$x_2 = (1,0011; 2,0159; 1,0142; 1,9916; 1,0233; 1,9861)^T.$$

Результат решения регрессионной задачи следующий:

$$\hat{\theta}_{1p} = 0,895, \quad \hat{\theta}_{2p} = -0,893, \quad \hat{\theta}_{3p} = 1,975.$$

Сумма квадратов невязок $\sum_{i=1}^6 (y_i - f(x_i, \theta_p))^2$ при этом равна 0,0232.

Результатом решения конфлюэнтной задачи являются оценки

$$\hat{\theta}_1 = 0,984, \quad \hat{\theta}_2 = -0,979, \quad \hat{\theta}_3 = 1,999.$$

Сумма квадратов невязок равна 0,0007. Матрица рассеяния оценок $\hat{\theta}$ имеет вид

$$\begin{pmatrix} 0,00025 & -0,00022 & -0,00002 \\ -0,00022 & 0,00023 & -0,00004 \\ -0,00002 & -0,00004 & 0,00004 \end{pmatrix}.$$

§ 7.15. Анализ систем в активном эксперименте

Пусть проводится эксперимент для определения оценок свободных параметров θ функции $\eta = f(\xi, \theta)$, описывающей некоторое распределение. Будем задавать x любые значения и измерять соответствующие им значения y , которые можно рассматривать как результат влияния случайной ошибки ϵ на истинное значение η :

$$y = \eta + \epsilon.$$

Переменная x называется *контролируемой*, ее значение фиксировано заранее. Однако и на значения x может налагаться случайная ошибка δ : $\xi = x + \delta$, т. е. неизвестное истинное значение ξ есть случайная величина. Пусть ошибки δ имеют нулевое среднее значение. Тогда среднее значение ξ будет равно x , коэффициент корреляции между ξ и δ равен +1 и структурное соотношение имеет вид

$$y = f(x + \delta, \theta) + \varepsilon. \quad (7.34)$$

Поскольку x не является случайной величиной, то ни ε , ни δ не коррелированы с переменной x . Таким образом, выражение (7.34) представляет собой обычное уравнение регрессии, к которому без всяких изменений можно применять методы регрессионного анализа, например метод наименьших квадратов. Закон распределения случайной величины $f(x + \delta, \theta)$ в выражении (7.34) может быть рассчитан по формулам переноса ошибок. Затем, в соответствии с полученным законом распределения случайной величины $f(x + \delta, \theta)$, методом максимума правдоподобия может быть составлен функционал, координаты точки минимума которого будут определять оценку вектора θ искомых параметров.

Если распределение случайной величины $f(x + \delta, \theta)$ подчиняется нормальному закону, то для оценки вектора параметров θ получим функционал для метода наименьших квадратов, но дисперсия случайной величины $f(x + \delta, \theta)$ будет зависеть от значений координат вектора θ , т. е. соответствующие уравнения для определения оценок $\hat{\theta}$ будут нелинейными.

Если контролируемая переменная x является случайной величиной (т. е. определяется с помощью некоторого процесса случайного выбора), полученные выше выводы останутся в силе, когда δ и ε не коррелированы с x [40]. Предположение о некоррелированности для ошибки ε обычно выполняется, но для ошибки δ ситуация сложнее. Например, требование некоррелированности δ и x в данном случае означает, что большие значения x не приводят к увеличению или уменьшению ошибок в определении истинного значения x . Выполнение этого условия может быть проверено только эмпирическим путем.

Заметим, что при неизвестной дисперсии $\sigma^2(\delta)$ или ее оценки в активной схеме эксперимента не идентифицируемы даже свободные параметры кубической параболы [40].

Рассмотрим в качестве примера задачу оценки свободных параметров θ в линейном уравнении

$$\eta = \theta_1 + \theta_2 \xi.$$

Выражая переменные η и ξ через x и y , получим структурное соотношение

$$y_i = \theta_1 + \theta_2 x_i + \theta_2 \delta_i + \varepsilon_i.$$

Пусть независимые случайные величины δ и ε подчиняются нормальному закону распределения с нулевыми математическими ожиданиями и известными дисперсиями $\sigma^2(x_i)$ и $\sigma^2(y_i)$. Функции плотности вероятности случайных величин $\theta_2 \delta_i + \varepsilon_i$ будут иметь следующие числовые характеристики:

$$M[\theta_2 \delta_i + \varepsilon_i] = 0,$$

$$\sigma^2[\theta_2 \delta_i + \varepsilon_i] = \theta_2^2 \sigma^2(x_i) + \sigma^2(y_i), \quad i = 1, 2, \dots, n.$$

В данном случае функционал метода наименьших квадратов, точка минимума которого определяет оценку вектора параметров θ , имеет вид

$$F = \sum_{i=1}^n \frac{(y_i - \theta_1 - \theta_2 x_i)^2}{\sigma^2(y_i) + \theta_2^2 \sigma^2(x_i)}. \quad (7.35)$$

Рассуждая аналогично, получим функционал метода наименьших квадратов для оценки вектора параметров θ линейных функций

$$\eta = \theta_0 + \sum_{j=1}^m \theta_j \xi_j;$$

структурное соотношение в данном случае имеет вид

$$y_i = \theta_0 + \sum_{j=1}^m \theta_j x_{ji} + \sum_{j=1}^m \theta_j \delta_{ji} + \varepsilon_i.$$

Числовыми характеристиками случайной величины $\sum_{j=1}^m \theta_j \delta_{ji} + \varepsilon_i$ являются

$$M \left[\sum_{j=1}^m \theta_j \delta_{ji} + \varepsilon_i \right] = 0,$$

$$\sigma^2 \left[\sum_{j=1}^m \theta_j \delta_{ji} + \varepsilon_i \right] = \sigma^2(y_i) + \sum_{j=1}^m \theta_j^2 \sigma^2(x_{ji}),$$

функционал метода наименьших квадратов в этом случае записывается в виде

$$F = \sum_{i=1}^n \frac{\left(y_i - \theta_0 - \sum_{j=1}^m \theta_j x_{ji} \right)^2}{\sigma^2(y_i) + \sum_{j=1}^m \theta_j^2 \sigma^2(x_{ji})}. \quad (7.36)$$

Функционалы (7.35) и (7.36) — те же, что и в пассивной схеме эксперимента.

Рассмотрим функцию

$$\eta = \theta_0 + \sum_{j=1}^m \theta_j \varphi_j(\xi),$$

где $\varphi_j(\xi)$ — функции произвольного вида. Тогда структурное соотношение имеет вид

$$y_i = \theta_0 + \sum_{j=1}^m \theta_j \varphi_j(x_i + \delta_i) + \varepsilon_i.$$

Здесь не всегда можно выделить случайную составляющую, присутствующую в $\varphi_j(x_i + \delta_i)$. Метод максимума правдоподобия (ММП) и в этом случае позволяет получить функционал, точка минимума которого дает оценки вектора θ искомых параметров. Вид функционала ММП и способ получения вектора оценок $\hat{\theta}$ определяются конкретным видом функций $\varphi_j(x)$.

В работе [30], чтобы упростить задачу получения оценок $\hat{\theta}$, функции $\varphi_j(x)$ разлагают в ряд в окрестности точки x_i :

$$M[y_i] = \theta^T \psi(x_i) + O(\sigma^3(x_i)),$$

$$\psi_j(x_i) = \varphi_j(x_i) + \frac{1}{2} \sigma^2(x_i) \operatorname{tr} \frac{\partial^2 \varphi_j}{\partial x \partial x^T},$$

где $\operatorname{tr} A$ — след матрицы A , $M[y_i]$ — математическое ожидание значений y_i . Дисперсия значений $\varphi_j(\xi)$ записывается в виде

$$M[(y_i - M[y_i])^2] = \sigma^2(y_i) + \sigma^2(x_i) \theta^T \frac{\partial \varphi}{\partial x^T} \frac{\partial \varphi}{\partial x} \theta + O(\sigma^3(x_i)).$$

Задача свелась к классической регрессионной задаче, оценки параметров θ находятся итерациями по методу наименьших квадратов с учетом того, что

$$y_i = \theta^T \psi(x_i) + \mu_i,$$

$$\psi_j(x) = \varphi_j(x) + \frac{1}{2} \sigma^2(x) \sum_{i=1}^n \frac{\partial^2 \varphi_j(x)}{\partial x_i^2},$$

$$M[\mu_i] = 0, \quad M[\mu_i^2] = \sigma^2(y_i) + \sigma^2(x_i) \theta^T \frac{\partial \varphi}{\partial x^T} \frac{\partial \varphi}{\partial x} \theta + O(\sigma^3(x_i)).$$

Для линейных функций по этим формулам получим тот же функционал (7.36), что и в пассивной схеме эксперимента. Для нелинейных функций оценки, полученные при обработке одних и тех же данных различными методами, будут существенно отличаться (очевидно, что экспериментальные данные должны обрабатываться тем методом, который следует из условий проведения эксперимента и статистики результатов наблюдений).

ПРИНЯТИЕ РЕШЕНИЙ ПО ВЫБОРКЕ ФИКСИРОВАННОГО ОБЪЕМА С УЧЕТОМ ПОГРЕШНОСТИ ПРИЗНАКОВ

§ 8.1. Статистические свойства параметров функции Гаусса, определенных непосредственно и с помощью операций линеаризации

Случайные величины, распределенные по нормальному закону, наиболее часто встречаются в практических приложениях, поскольку сумма даже трех соизмеримых равномерно распределенных случайных величин имеет в результате распределение, близкое к нормальному. Плотность распределения вероятностей в данном случае имеет вид функции Гаусса. В приложениях часто функцию Гаусса

$$y = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp \left\{ -\frac{(x-a)^2}{2\sigma^2} \right\}$$

путем преобразования координат сводят к уравнению прямой

$$u_p = \frac{1}{\sigma}x - \frac{a}{\sigma},$$

где u_p — p -квантиль случайной величины x с функцией распределения вероятностей $F(x)$, т. е. такое значение аргумента функции $F(x)$, для которого вероятность события $x < u_p$ равна заданному значению вероятности p .

Представим (рис. 8.1) на плоскости x, u_p прямую, полученную в процессе линеаризации функции Гаусса. Эта прямая пересекает ось абсцисс в точке $L(x_L, 0)$, а горизонтальную прямую $x_p = -1$ в точке $N(x_N, -1)$. Координата x_L определяет оценку параметра a , т. е. $\hat{a} = x_L$, а разность абсцисс точки $L(x_L, 0)$ и точки $N(x_N, -1)$ определяет оценку параметра σ , т. е. $\hat{\sigma} = x_L - x_N$.

Таким образом, рассматривая уравнение прямой в виде $\eta = \theta_1 + \theta_2 \xi$, получаем, что оценка $\hat{\theta}_1$ будет определять величину $-a/\sigma$, а оценка $\hat{\theta}_2$ — величину $1/\sigma$.

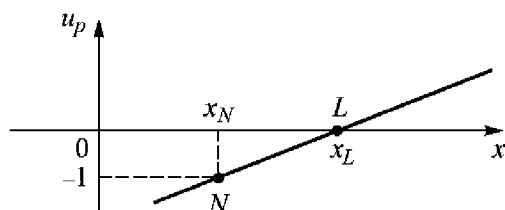


Рис. 8.1. Линеаризация функции Гаусса

Исследуем одновременно статистику параметров a и σ^2 и параметров θ_1 и θ_2 , чтобы понять особенности процедуры линеаризации. Начнем рассмотрение с операции линеаризации. При определении свободных па-

раметров θ_1 и θ_2 уравнение прямой часто записывают в виде

$$\eta = \theta'_1 + \theta_2(\xi - \bar{\xi}),$$

где $\bar{\xi}$ — среднее значение величины ξ . Оценки $\hat{\theta}'_1$ и $\hat{\theta}_2$ статистически независимы, однако это утверждение не выполняется для оценок $\hat{\theta}_1$ и $\hat{\theta}_2$. Оценки $\hat{\theta}'_1$ и $\hat{\theta}_2$ параметров можно получить, не решая совместных систем связанных уравнений, как это приходится делать при другой форме записи модели.

После операции линеаризации для нормально распределенных статистически независимых погрешностей экспериментальных точек (x_i, y_i) минимизируемый функционал имеет следующий вид:

$$F = \sum_{i=1}^n \left(\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} + \frac{(y_i - \eta_i)^2}{\sigma^2(y_i)} \right).$$

Из условий (7.11) и (7.12) получим

$$\xi_i = \frac{\sigma^2(y_i)x_i + \theta_2\sigma^2(x_i)(y_i - \theta_1)}{\sigma^2(y_i) + \theta_2^2\sigma^2(x_i)}.$$

Тогда

$$F = \sum_{i=1}^n \frac{(y_i - \theta_1 - \theta_2 x_i)^2}{\sigma^2(y_i) + \theta_2^2 \sigma^2(x_i)} = \sum_{i=1}^n w(y_i - \theta_1 - \theta_2 x_i)^2,$$

где $w^{-1} = \sigma^2(y_i) + \theta_2^2 \sigma^2(x_i)$.

Ковариационную матрицу оценок $D(\hat{\theta})$ найдем как матрицу N^{-1} , обратную матрице N с элементами

$$-\frac{\partial F}{\partial \theta_1^2}, \quad -\frac{\partial F}{\partial \theta_1 \partial \theta_2} = -\frac{\partial F}{\partial \theta_2 \partial \theta_1}, \quad -\frac{\partial F}{\partial \theta_2^2},$$

вычисленными при найденных значениях оценок $\hat{\theta}_2$ и $\hat{\theta}'_1$.

После определения точечных оценок $\hat{\theta}_1$, $\hat{\theta}_2$ или $\hat{\theta}'_1$, $\hat{\theta}_2$ и их дисперсий $D(\hat{\theta}_1)$, $D(\hat{\theta}_2)$ или $D(\hat{\theta}'_1)$, $D(\hat{\theta}_2)$ находят доверительные интервалы для координат вектора параметров θ . Если число экспериментальных точек достаточно велико и можно считать, что оценки параметров распределены относительно их математических ожиданий по нормальному закону, то для получения доверительных интервалов применяют безразмерную t -статистику Стьюдента, которая подчиняется t -распределению с $\nu = n - 1$ степенями свободы.

Для получения оценок свободных параметров функции Гаусса без линеаризации находится точка минимума функционала

$$F = \sum_{i=1}^n \left(\frac{(x_i - \xi_i)^2}{\sigma^2(x_i)} + \frac{(y_i - \eta_i)^2}{\sigma^2(y_i)} \right)$$

при дополнительном условии

$$\frac{\partial F}{\partial \xi_i} = \frac{x_i - \xi_i}{\sigma^2(x_i)} + \frac{y_i - \eta_i}{\sigma^2(y_i)} \frac{\partial \eta_i}{\partial \xi_i} = 0,$$

$$\text{где } \eta_i = \frac{1}{\sqrt{2\pi} \cdot \theta_2} \exp \left\{ -\frac{(\xi_i - \theta_1)^2}{2\theta_2^2} \right\}, \quad \theta_1 \equiv a, \quad \theta_2 \equiv \sigma.$$

Знание статистики получаемых оценок необходимо для дальнейшего анализа экспериментальных данных и правильной трактовки результатов их обработки, а также для выявления последствий сделанных допущений. Приведем результаты числовых экспериментов для определения закона распределения оценок параметров θ_1 и θ_2 функции Гаусса для различных погрешностей элементов выборок x и y при исследовании зависимости средних квадратических отклонений оценок параметров функции Гаусса, соответствующей прямой линии $\eta = 2 + \xi$, от числа заданных точек на отрезке $[0, 2; 9, 8]$ и закона их расположения на этом отрезке. Как и следовало ожидать из общетеоретических соображений, концентрация точек к концам прямой приводит к уменьшению дисперсии оценок. Затем для различного числа экспериментальных точек (от 6 до 50) и различных комбинаций относительных погрешностей по обеим осям координат ($\delta(x_i) = 1 \dots 10 \%$, $\delta(y_i) = 1 \dots 10 \%$) с помощью датчика нормально распределенных случайных чисел исследовался вид функции распределения оценок.

В каждом варианте вычислялись также дисперсии оценок и определялись доверительные интервалы дисперсий. Приведем результаты, полученные для 17 экспериментальных точек. Как показал анализ оценок параметров θ_1 и θ_2 по критерию χ^2 -распределения с вероятностью, приблизительно равной 0,80, закон распределения оценок можно считать нормальным со средними квадратическими отклонениями соответственно $\sigma(\hat{\theta}_1) = 0,060$ и $\sigma(\hat{\theta}_2) = 0,017$.

Процедура линеаризации имеет и свои недостатки. При обработке нелинейных функций таким методом следует иметь в виду, что для того, чтобы оценки $\hat{\theta}$ соответствующих параметров θ , полученные из преобразованного уравнения, обладали оптимальными свойствами (несмещенностью, минимальной дисперсией и т. д.), необходимо, чтобы предположение аддитивности ненаблюдаемой случайной ошибки было справедливым для преобразованной, а не для первоначальной модели.

Влияние аддитивной ошибки на величину оценки параметров в преобразованной и исходной моделях можно установить, только исследуя каждую конкретную модель.

Из полученных результатов следует, что операция линеаризации занижает реальную интервальную оценку параметра σ .

В табл. 8.1 приведены относительные изменения дисперсии оценки математического ожидания для нормального закона распределения при различных комбинациях относительных погрешностей признака $\delta(x_i)$ и частоты $\delta(n_{0i})$, где n_{0i} — частота попадания значений оценки $\hat{\theta}_1$ в i -ю ячейку гистограммы.

Таблица 8.1

Относительное изменение дисперсии оценки параметра θ_1 при различных комбинациях относительных погрешностей признака $\delta(x_i)$ и частоты $\delta(n_{0i})$

$\delta(n_{0i}), \%$	$\delta(x_i), \%$				
	0,5	1,5	2,5	3,0	5,0
0,5	1	1,078	1,083	1,084	1,089
1,5	5,467	9,064	9,557	9,655	9,753
3,0	9,310	29,16	35,02	36,25	38,27
5,0	10,29	42,96	57,14	60,59	66,04

Отсюда можем сделать следующие выводы:

- 1) неучет погрешности аргумента занижает реальную интервальную оценку параметров;
- 2) разные алгоритмы приводят к различным интервальным оценкам.

Наиболее оптимальные оценки должны получаться в тех алгоритмах, в которых операции обработки данных ведутся непосредственно с исходными экспериментальными данными, когда не нарушается принцип аддитивности и не изменяется искусственно закон распределения исходных данных.

Большое практическое значение имеет положение экспериментальных точек на исследуемом интервале возможных значений. Оптимальный закон их распределения (обеспечивающий минимум дисперсии оценок) зависит от конкретных значений экспериментальных величин и их погрешностей, и без упрощающих предположений о последних получить его в явном виде не удастся. Расчеты показали, что при одинаковых относительных ошибках в каждой точке для всех рассмотренных алгоритмов предпочтительнее выбрать равномерный шаг по оси ординат, чем по оси абсцисс. В этом случае дисперсии оценок будут минимальными.

§ 8.2. Оценка параметров функции плотности распределения вероятностей с учетом погрешности вектора признаков

Применим результаты, полученные в гл. 7, для решения статистической задачи распознавания образов. Рассмотрим первый этап задачи классификации, когда требуется по результатам наблюдений найти оценки функций плотности распределения вероятностей. Пусть известен вид функции распределения $P(\xi, \theta | \omega_j)$, $j = 1, 2, \dots, m$, но не известны значения вектора параметров θ . По результатам наблюдений случайных векторов x_1, x_2, \dots, x_n требуется найти оценки вектора свободных параметров θ и дисперсии оценок $D(\hat{\theta})$. Проблема состоит в том, что наблюдаемые значения x_1, x_2, \dots, x_n имеют соответствующие погрешности и аппроксимирующая функция распределения $P(\xi, \theta | \omega_j)$ не строго пройдет по наблюдаемым точкам, т. е. в каждой наблюдаемой точке образуется область неопределенности, форма которой определяется законами распределения координат наблюдаемой точки. Для получе-

ния оценок параметров и их дисперсий в данной ситуации применим изложенные в гл. 7 методы конфлюэнтного анализа [20, 40].

В приведенных ранее традиционных методах классификации [3, 7, 8, 14, 17–19, 28] определялись только точечные оценки параметров в законах распределения, причем погрешности наблюдаемых случайных величин (векторов) x_1, x_2, \dots, x_n при этом не учитывались.

Предположим, что координаты $\{x_{ik}\}$ наблюдений случайных векторов x_1, x_2, \dots, x_n для j -го класса образов — непрерывные случайные величины с известными функциями плотности распределения вероятностей $f(x_{ik})$, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, l$. Также будем считать, что в процессе аппроксимации результатов наблюдений x_1, \dots, x_n с помощью функции распределения известного вида $P(\xi, \theta | \omega_j)$ предварительно вычислены (способом группировки, например) эмпирические значения вероятностей $P(x_{ik} | \omega_j)$, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, l$, с которыми сравниваются соответствующие значения $P(\xi_{ik}, \theta | \omega_j)$, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, l$. Определен также вид функции плотности распределения вероятностей $\phi(p_{ik} | \theta)$ для значений $P(x_{ik} | \omega_j)$. Тогда согласно методу максимума правдоподобия функция правдоподобия будет иметь вид [21, 40]

$$L(x, p | \theta) = \prod_{i=1}^n \prod_{k=1}^l \phi(p_{ik} | \theta) f(x_{ik}).$$

Предположим, что все случайные величины подчиняются многомерному распределению Гаусса с математическими ожиданиями величин $P(\xi_{ik}, \theta | \omega_j)$ и ξ_{ik} , ковариационной матрицей K и дисперсиями $\sigma^2(x_{ik})$. В этом случае для образа ω_j имеем

$$\begin{aligned} \ln L(x, p | \theta) = & -\frac{1}{2} \left\{ [P - P(\xi, \theta | \omega_j)]^T K^{-1} [P - P(\xi, \theta | \omega_j)] + \right. \\ & \left. + \sum_{i=1}^n \sum_{k=1}^l \frac{(x_{ik} - \xi_{ik})^2}{\sigma^2(x_{ik})} \right\} + \text{const}, \end{aligned} \quad (8.1)$$

где P — набор значений $P(x_{ik} | \omega_j)$, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, l$ и $P(\xi, \theta | \omega_j)$ — набор элементов $P(\xi_{ik}, \theta | \omega_j)$, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, l$.

Оценки вектора параметров θ функции $P(\xi, \theta | \omega_j)$ и их дисперсии $D(\hat{\theta})$ найдем из условия максимума (8.1) или как точку минимума функционала

$$F(x, p | \theta) = [P - P(\xi, \theta | \omega_j)]^T K^{-1} [P - P(\xi, \theta | \omega_j)] + \\ + \sum_{i=1}^n \sum_{k=1}^l \frac{(x_{ik} - \xi_{ik})^2}{\sigma^2(x_{ik})}. \quad (8.2)$$

Функционал (8.2) можно было бы сразу записать как функционал ортогональной регрессии [21, 43], позволяющий учесть как погрешность определения P , так и погрешности измерений величин x_1, x_2, \dots, x_n . Как уже отмечалось в гл. 7, в ортогональной регрессии минимизируется сумма квадратов расстояний от наблюдавшихся точек до линии (поверхности) регрессии.

Точечные оценки вектора параметров θ и неизвестных ξ_{ik} можно найти из условий

$$\left. \frac{\partial F}{\partial \theta_v} \right|_{\theta = \hat{\theta}} = 0, \quad v = 1, 2, \dots, N, \quad (8.3)$$

$$\frac{p_{ik} - p(\hat{\xi}_{ik}, \theta | \omega_j)}{\sigma^2(p_{ik})} p'_{\xi_{ik}}(\hat{\xi}_{ik}, \theta | \omega_j) + \frac{x_{ik} - \hat{\xi}_{ik}}{\sigma^2(x_{ik})} = 0, \quad (8.4) \\ i = 1, 2, \dots, n, \quad k = 1, 2, \dots, l,$$

где $\sigma^2(p_{ik})$ — дисперсия значений p_{ik} , $p'_{\xi_{ik}}$ — производная от функции $p(\hat{\xi}_{ik}, \theta | \omega_j)$ по ξ_{ik} . Решая системы уравнений (8.3) и (8.4) совместно, находим точечные оценки $\hat{\theta}$ и $\hat{\xi}$. Условия (8.3) — система алгебраических уравнений относительно ξ_{ik} , из которой можно найти явные выражения для ξ_{ik} , подставить их в (8.2) и получить более простую форму функционала (8.2).

Для прямой $p = a + bx$ функционал (8.2) при $\sigma^2(p_{ik}) \equiv \sigma^2(p)$ и $\sigma^2(x_{ik}) \equiv \sigma^2(x)$, $i = 1, 2, \dots, n$, $k = 1, 2, \dots, l$, примет вид

$$F(x, p | \theta) = \frac{1}{\sigma^2(p) + b^2 \sigma^2(x)} \sum_{i=1}^n (p_i - a - bx_i)^2; \quad (8.5)$$

для гиперплоскости $p = a + \sum_{k=1}^l b_k x_k$ при тех же предположениях

$$F(x, p | \theta) = \frac{1}{\sigma^2(p) + \sum_{k=1}^l b_k^2 \sigma^2(x_k)} \sum_{i=1}^n \left(p_i - a - \sum_{k=1}^l b_k x_k \right)^2. \quad (8.6)$$

Функционалы (8.5) и (8.6) позволяют непосредственно (согласно (8.3)) найти оценки свободных параметров θ и их дисперсий $D(\hat{\theta})$. Дисперсионную матрицу оценок свободных параметров θ находим из следующего условия [21]:

$$D(\hat{\theta}) = M^{-1}, \quad M = \left(-\frac{\partial^2 F}{\partial \theta_{j_1} \partial \theta_{j_2}} \right) \bigg|_{\theta = \hat{\theta}}, \quad j_1, j_2 = 1, 2, \dots, N.$$

Пользуясь полученными формулами, покажем теперь, что линейная разделяющая функция, определенная без учета погрешностей признаков x и y для двух образов, для которых $p(x | \omega_i)$, $i = 1, 2$, — функция плотности двумерного гауссова распределения с математическими ожиданиями (x_i, y_i) , $i = 1, 2$, и одной ковариационной матрицей $K = \begin{pmatrix} \sigma^2(x) & 0 \\ 0 & \sigma^2(y) \end{pmatrix}$, является смещенной оценкой реальной разделяющей функции.

Разделяющая функция в данном случае имеет вид уравнения прямой. Данная прямая ортогональна к середине отрезка, проходящего через точки (x_i, y_i) , $i = 1, 2$, т. е. имеет угловой коэффициент, равный

$$\frac{(x_2 - x_1)\sigma^2(y)}{\sigma^2(x)(y_2 - y_1)}.$$

Найдем уравнение прямой $y = a + bx$, проходящей через точки (x_i, y_i) , $i = 1, 2, \dots, n$, учитывая дисперсии оценок $\sigma^2(x)$ и $\sigma^2(y)$. Оценки параметров a и b искомой прямой определим как координаты точки минимума функционала (8.5) из системы уравнений (8.3), взяв в качестве параметров θ соответственно a и b :

$$\frac{1}{\sigma^2(y) + b^2 \sigma^2(x)} \sum_{i=1}^n (y_i - a - bx_i) = 0,$$

$$\frac{\sum_{i=1}^n (y_i - a - bx_i)x_i}{\sigma^2(y) + b^2 \sigma^2(x)} + \frac{\sum_{i=1}^n (y_i - a - bx_i)^2}{[\sigma^2(y) + b^2 \sigma^2(x)]^2} b \sigma^2(x) = 0.$$

Из первого уравнения получим

$$\hat{a} = \frac{1}{n} \left(\sum_{i=1}^n y_i - b \sum_{i=1}^n x_i \right).$$

Подставим \hat{a} во второе уравнение, которое станет квадратным относительно \hat{b} . Оценку углового коэффициента b определим из квадратного уравнения

$$\hat{b}^2 + \hat{b} \frac{\sigma^2(y) \left[\left(\sum_{i=1}^n x_i \right)^2 - n \sum_{i=1}^n x_i^2 \right] + \sigma^2(x) \left[n \sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2 \right]}{\sigma^2(x) \left(\sum_{i=1}^n y_i \sum_{i=1}^n x_i - n \sum_{i=1}^n x_i y_i \right)} - \frac{\sigma^2(y)}{\sigma^2(x)} = 0.$$

Для двух точек (x_1, y_1) , (x_2, y_2) это уравнение примет вид

$$\hat{b}^2 + \hat{b} \frac{\sigma^2(x)(y_1 - y_2)^2 - \sigma^2(y)(x_1 - x_2)^2}{\sigma^2(x)(y_1 - y_2)(x_1 - x_2)} - \frac{\sigma^2(y)}{\sigma^2(x)} = 0.$$

В общем случае угловой коэффициент $\frac{(x_2 - x_1)\sigma^2(y)}{\sigma^2(x)(y_2 - y_1)}$ не явля-

ется корнем данного квадратного уравнения, за исключением редких случаев, когда не различимы разные линии регрессии, что подтверждает смещенность оценки, полученной без учета $\sigma^2(x)$ и $\sigma^2(y)$. Угловой коэффициент b однозначно связан с угловым коэффициентом разделяющей прямой: их произведение равно -1 .

Далее оценим смещение разделяющих функций при учете погрешностей признаков на примере одномерных гауссовых функций плотности распределения вероятностей $p(x|\omega_1)$ и $p(x|\omega_2)$ с математическими ожиданиями m_1 и m_2 соответственно, а также с одинаковыми дисперсиями σ^2 .

В традиционных методах не различаются значения x и ξ , поэтому в качестве функции плотности распределения берется $p(x|\omega_j)$ вместо $p(\xi|\omega_j)$. Тем не менее в процессе оценки свободных параметров m_1 , m_2 и σ^2 функций, распределенных по

нормальному закону, должны быть получены соответствующие погрешности Δm_1 , Δm_2 и $\Delta \sigma$.

Разделяющая граница в традиционных методах может быть определена согласно выражению

$$\ln \frac{p(x|\omega_1)}{p(x|\omega_2)} = 0, \quad (8.7)$$

где

$$p(x|\omega_i) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp \left\{ -\frac{1}{2\sigma^2} (x - m_i)^2 \right\}, \quad i = 1, 2,$$

и в данном случае при $\Delta m_1 = \Delta m_2 = 0$ разделяющая граница проходит через точку $(m_1 + m_2)/2$.

В действительности из-за наличия погрешностей Δm_1 , Δm_2 и $\Delta \sigma$ вместо выражения (8.7) следует записать

$$\ln \frac{p(x|\omega_1) \pm \Delta p(x|\omega_1)}{p(x|\omega_2) \pm \Delta p(x|\omega_2)} = 0. \quad (8.8)$$

Погрешности $\Delta p(x|\omega_i)$ получим как линейные члены разложения в ряд Тейлора функций $p(x|\omega_i)$, $i = 1, 2$:

$$\Delta p(x|\omega_i) = p'_{m_i}(x|\omega_i) \Delta m_i + p'_\sigma(x|\omega_i) \Delta \sigma, \quad i = 1, 2. \quad (8.9)$$

Здесь $p'_{m_i}(x|\omega_i)$ и $p'_\sigma(x|\omega_i)$ — производные функции $p(x|\omega_i)$ по m_i и σ соответственно.

Найдем максимальное смещение разделяющей границы (в числителе выражения (8.8) прибавляется погрешность $\Delta p(x|\omega_1)$, а в знаменателе вычитается $\Delta p(x|\omega_2)$), учитывая только Δm_1 и Δm_2 . Используя соотношения (8.9), перепишем выражение (8.8) в следующем виде:

$$\ln \frac{p(x|\omega_1)}{p(x|\omega_2)} + \ln \frac{1 + \frac{(x - m_1)}{\sigma^2} \Delta m_1}{1 - \frac{(x - m_2)}{\sigma^2} \Delta m_2} = 0.$$

Разложим функции $\ln \left(1 + \frac{x - m_1}{\sigma^2} \Delta m_1 \right)$ и $\ln \left(1 - \frac{x - m_2}{\sigma^2} \Delta m_2 \right)$ в ряд Маклорена. Поскольку здесь не обязательно выполняется

условие $|x| < 1$ для разложения функции $\ln(1+x)$ в ряд Маклорена, то воспользуемся разложением

$$\ln \frac{1+x}{1-x} = 2\left(x + \frac{x^3}{3} + \dots\right), \quad |x| < 1,$$

и сделаем замену переменной

$$\frac{1+x}{1-x} = \frac{N+k}{k}$$

(см. [46]). Тогда $x = k/(2N+k)$ — правильная положительная дробь и

$$\ln \frac{N+k}{N} = 2 \left[\frac{k}{2N+k} + \frac{1}{3} \left(\frac{k}{2N+k} \right)^3 + \dots \right]. \quad (8.10)$$

Выбираем значение N , по которому найдем k . В рассматриваемом случае

$$N=1, \quad k_i = \frac{x - m_i}{\sigma^2} \Delta m_i, \quad i=1, 2.$$

При условии (8.10), если $\Delta m_1 = \Delta m_2 = \Delta m$, выражение (8.8) принимает вид

$$\begin{aligned} x &= \frac{0,5(m_1^2 - m_2^2) + 2\Delta m(m_1 + m_2)}{m_1 - m_2 + 4\Delta m} \approx \\ &\approx \frac{m_1 + m_2}{2} + \frac{2\Delta m(m_1 + m_2)}{m_1 - m_2 + 4\Delta m}. \end{aligned} \quad (8.11)$$

Второе слагаемое здесь определяет возможное смещение разделяющей границы. При другой комбинации знаков $\Delta p(x|\omega_j)$, $j=1, 2$, в выражении (8.8) смещение будет равно $2\Delta m$.

Найдем теперь максимальное смещение границы, учитывая только погрешность $\Delta\sigma$. Выражение (8.8) с учетом равенства (8.10) примет вид

$$\begin{aligned} &8\Delta\sigma x^2 + x[2\sigma(m_1 - m_2) - 8\Delta\sigma(m_1 + m_2)] - \\ &- \sigma(m_1^2 - m_2^2) - 4\sigma^2\Delta\sigma + 4\Delta\sigma(m_1^2 - m_2^2) = 0. \end{aligned} \quad (8.12)$$

Смещение разделяющей границы в данном случае следует из того, что $x = (m_1 + m_2)/2$ не является корнем уравнения (8.12).

Обратимся к алгоритму классификации, с помощью которого можно учесть погрешности априорной информации $p(\xi|\omega_i)$, $i=1, \dots, m$. В § 6.1 было показано, что разделяющая функция, которую получают в традиционных методах из условия

$$\ln \frac{P(\omega_1)p(x|\omega_1)}{P(\omega_2)p(x|\omega_2)} = 0,$$

является смещенной оценкой реальной разделяющей функции. Несмещенная оценка разделяющей функции может быть получена путем стохастической аппроксимации по граничным точкам, найденным из условия

$$\ln \frac{P(\omega_1)\hat{p}(\xi|\omega_1)}{P(\omega_2)\hat{p}(\xi|\omega_2)} = 0, \quad (8.13)$$

где $\hat{p}(\xi|\omega_j)$ — наблюдаемые значения плотности вероятностей появления в выбранных областях D_i , $i=1, 2, \dots, n$, $x_i \in D_i$, измеренных значений x_1, x_2, \dots, x_n для j -го образа. Значения $\hat{p}(\xi|\omega_j) \equiv \hat{p}$ известны, они использовались для определения вида функциональной зависимости $p(x|\omega_j)$ на первом этапе, когда предполагалось, что $\xi = x$.

Из заданных условий определяют общий вид (класс) разделяющих функций $\Psi(\xi, \theta)$ (линейный, гиперквадратный и т. п.), свободные параметры θ которых оценивают по точкам из условия (8.13), решая совместно системы уравнений (8.3) и (8.4), заменяя в последних $p(\xi|\omega_j)|_{\xi=x}$ на $\Psi(\xi, \theta)$ и p_i на числовые значения

$$\ln \frac{P(\omega_1)\hat{p}(\xi|\omega_1)}{P(\omega_2)\hat{p}(\xi|\omega_2)}.$$

Другая возможность найти разделяющую функцию $\Psi(\xi, \theta)$ — использовать сразу условие (8.13) и получить аналитический вид разделяющей функции $\Psi(\xi, \theta)$.

Наиболее часто в задачах распознавания образов применяют линейные разделяющие функции (гиперплоскости). Оценки свободных параметров гиперплоскости находят как координаты точки минимума функционала (8.6). В данном случае в системе алгебраических уравнений (8.3) уравнение для оценки параметра a будет линейным, а остальные уравнения образуют систему квадратных

уравнений относительно параметров b_k , $k = 1, 2, \dots, l$. Методы решения подобных систем, как и ряда задач аппроксимации результатов наблюдений элементарными функциями при учете погрешностей во всех координатах, рассмотрены в работе [20].

Дисперсия оценок значений функции $\Psi(\xi, \hat{\theta})$ при $\xi = x$ определяется по формуле

$$D[\Psi(x, \hat{\theta})] = \sum_{j=1}^n \left(\frac{\partial \Psi(x, \theta)}{\partial \theta_j} \right)^2 + 2 \sum_{i=1}^n \sum_{j=2, j>i}^n \frac{\partial \Psi(x, \theta)}{\partial \theta_i} \frac{\partial \Psi(x, \theta)}{\partial \theta_j} D(\hat{\theta}_i, \hat{\theta}_j)$$

при $\theta = \hat{\theta}$.

Изложенные результаты получены в предположении, что функция потерь $L(\omega, d)$ имеет вид

$$L(\omega_i, d_i) = \begin{cases} 0 & \text{при } i = j, \\ 1 & \text{при } i \neq j. \end{cases}$$

В общем случае, когда функция потерь равна нулю для правильного решения, равна l_1 , если ошибочно выбирается образ ω_2 вместо ω_1 , равна l_2 , если ошибочно выбирается ω_1 вместо ω_2 , решение с минимальным риском находят по выражению

$$l_1 P(\omega_1) p(\xi | \omega_1), \quad i = 1, 2.$$

Образ ω_1 выбирается, если выполнено условие

$$l_1 P(\omega_1) p(\xi | \omega_1) > l_2 P(\omega_2) p(\xi | \omega_2).$$

Уравнение разделяющей функции определяют, используя выражение

$$\ln \frac{l_1 P(\omega_1) p(\xi | \omega_1)}{l_2 P(\omega_2) p(\xi | \omega_2)} = 0.$$

§ 8.3. Плохая обусловленность и некорректность в задачах оценки параметров функции

Итерационный алгоритм решения конфлюэнтной задачи состоит из двух основных этапов: решения системы алгебраических уравнений для получения оценки $\hat{\theta}$ и решения методом линеаризации систем нелинейных алгебраических уравнений для определения оценок $\hat{\xi}$ истинных значений аргументов ξ . При этом воз-

можно ситуация, когда бесконечно малым приращениям в исходных данных могут отвечать сколь угодно большие изменения в решении [20, 71]. Такие системы называются *плохо обусловленными*, а задачи — *некорректными*. Приращения в исходных данных могут быть вызваны как ошибками в измерениях, так и округлением величин в процессе расчетов на компьютере. В плохо обусловленных системах строгое математическое решение может не соответствовать «физической» постановке задачи. Критической величиной, которая определяет физическую надежность строгого математического решения, является отношение наибольшего собственного значения симметрической матрицы $A^T A$, где A — матрица системы алгебраических уравнений, к наименьшему. Квадратный корень этого отношения показывает увеличение помех в направлении, соответствующем наименьшему собственному значению матрицы $A^T A$ [20, 54].

В качестве другой характеристики обусловленности системы с квадратной матрицей A размерностью n вводят N -обуславливающие числа,

$$N = \frac{1}{n} \|A\| \|A^{-1}\|,$$

где $\|A\|$ и $\|A^{-1}\|$ — нормы соответственно матрицы A и обратной матрицы A^{-1} .

Наилучшими обусловленными матрицами являются *ортогональные*, для которых N -обуславливающие числа равны единице. Ортогональные матрицы удовлетворяют условию $A^{-1} = A^T$.

В процессе обработки результатов наблюдений и аппроксимации исходных данных функциями различных видов с большим числом оцениваемых параметров в решении появляются осцилляции. Они возникают не только из-за погрешностей наблюдений, но и в результате неадекватного представления исследуемого явления выбранными функциями. С одной стороны, желательно описать как можно точнее изучаемое явление большим числом параметров, с другой — увеличение размерности задачи ухудшает обусловленность систем, и задача становится некорректной. Обусловленность систем линейных алгебраических уравнений еще больше ухудшается, когда мы имеем дело с коррелированными исходными случайными величинами.

Задача считается поставленной корректно, если ее решение удовлетворяет условиям Адамара, которые для операторного уравнения $y = A\theta$ формулируются следующим образом [71]:

- 1) решение θ существует для любого $y \in Q_A \subseteq F$;
- 2) решение θ единственно в пространстве U ;
- 3) решение θ непрерывно зависит от y , т. е. если приращение Δy стремится к нулю, то приращение $\Delta\theta$ также стремится к нулю.

Другими словами, задача некорректна, если определитель линейного оператора A равен нулю. Здесь θ является элементом метрического пространства U , а y — элементом метрического пространства F . Областью определения оператора A , действующего из U в F , является $D_A \subseteq U$, областью его значений — $Q_A = A(D_A) \subseteq F$.

Иногда плохо обусловленные и некорректные задачи называют *некорректными*. Этим термином мы будем пользоваться в дальнейшем.

Таким образом, чтобы корректно учитывать реальную экспериментальную информацию, необходимо иметь метод решения некорректных задач. Для этой цели, в частности, применяется метод регуляризации А.Н. Тихонова и большая серия методов, развитых на его основе [21, 54, 71].

Строго говоря, в задачах, приведенных в гл. 7, мы должны были бы применять на каждой итерации метод регуляризации А.Н. Тихонова для решения плохо обусловленных систем линейных алгебраических уравнений. В действительности имеет место более простая ситуация.

Во-первых, если области допустимых значений аргументов не пересекаются, что имеет место в большинстве практических задач, и учитываются ограничения, налагаемые на допустимые значения оценок $\hat{\xi}$ при вычислениях, то процесс нахождения оценок $\hat{\xi}$ является корректным.

Во-вторых, если в практических задачах число определяемых параметров невелико, то одна из возможных причин некорректности задачи — большая размерность системы — не возникает и соответствующая система линейных алгебраических уравнений может быть достаточно хорошо обусловлена. Тем не менее, некорректные задачи в процессе принятия решений приходится решать.

Пусть требуется найти решение θ системы линейных алгебраических уравнений

$$y = A\theta,$$

минимизируя невязку $\rho^2(A\theta, y) = J(\theta)$. При определенных условиях задача минимизации невязки $\rho^2(A\theta, y)$ может быть некорректной. В методе регуляризации А.Н. Тихонова вводится функция $\Omega(\theta)$, определенная на непустом множестве $U_\Omega \subseteq U$ и называемая *стабилизатором*. Функция $\Omega(\theta)$ должна обладать следующими свойствами [71]:

- 1) $\Omega(\theta) \geq 0$ для всех $\theta \in U_\Omega$;
- 2) множество $\Omega_C = \{\theta | \theta \in U_\Omega; \Omega(\theta) \leq C\}$ является ρ -компактным при любом $C = \text{const} \geq 0$, т. е. из любой последовательности $\{\theta_k\} \in \Omega_C$ можно выбрать подпоследовательность $\{\theta_{k_i}\}$, ρ -сходящуюся к некоторой точке $\theta \in \Omega_C$;
- 3) множество $U_\Omega^* = U_\Omega \cap U^*$ непустое (U^* — множество точек минимумов функции $J(\theta) = \rho_F^2(A\theta, y)$).

Далее берется какая-либо положительная последовательность $\{\alpha_k\}$, сходящаяся к нулю, и при каждом $k = 1, 2, \dots$ на множестве U_Ω определяется функция Тихонова

$$T_k(\theta) = J(\theta) + \alpha_k \Omega(\theta), \quad \theta \in U_\Omega.$$

Минимум функции Тихонова для различных значений k определяет минимизирующую последовательность $\{\theta_k\}$, сходящуюся к регуляризованному решению $\hat{\theta}_\rho$.

Существуют алгоритмы, в которых по величине погрешности исходных данных определяется единственное (оптимальное) значение параметра регуляризации α в функции Тихонова и сразу находится регуляризованное решение $\hat{\theta}_\rho$. Нас будет интересовать не только регуляризованное решение $\hat{\theta}_\rho$, но и интервальная оценка этого решения, поскольку мы оперируем исходными случайными величинами.

Рассмотрим функцию Тихонова как функцию Лагранжа следующей задачи: минимизировать функцию $\Omega(\theta)$ на множестве $U_\Omega \subseteq U$ при условии

$$\rho^2(A\theta, y) = \delta^2.$$

Здесь δ — величина погрешности исходных данных, параметр α функции Тихонова является множителем Лагранжа.

Выбор стабилизатора $\Omega(\theta)$ в методе регуляризации неоднозначен, часто для систем линейных алгебраических уравнений его выбирают в виде $\Omega(\theta) = \|\theta\|^2$. Для выделения редких сигналов применяют методы неквадратичной l_p -регуляризации, где $\|\theta\|^p$, $0 < p < 1$ [110].

Если функция $\rho^2(A\theta, y)$ для $y = A\theta + \varepsilon$ может быть записана в виде

$$\rho^2(A\theta, y) = (y - A\theta)D^{-1}(y - A\theta),$$

где $D(y)$ — ковариационная матрица погрешностей исходных данных, и стабилизатор имеет линейный вид

$$\Omega(\theta) = L\theta - R,$$

то получим следующее:

- 1) вектор оценок $\hat{\theta}$ точки минимума функции Тихонова

$$T = (y - A\theta)^T D^{-1}(y - A\theta) + \alpha^T (L\theta - R)$$

имеет вид $\hat{\theta} = FA^T D^{-1}(y) + G^T R$;

- 2) матрица вторых моментов оценок имеет вид

$$D(\hat{\theta}) = C^{-1} - C^{-1}L^T(LC^{-1}L^T)^{-1}LC^{-1}; \quad (8.14)$$

3) дисперсия оценки параметра регуляризации α (множителя Лагранжа) имеет вид

$$D(\hat{\alpha}) = (LC^{-1}L^T)^{-1}. \quad (8.15)$$

Здесь $F = C^{-1} - C^{-1}L^T(LC^{-1}L^T)^{-1}LC^{-1}$, $G = (LC^{-1}L^T)^{-1}LC$, $C = A^T D^{-1}(y)A$. Диагональные элементы второго члена выражения (8.14) служат мерой уменьшения дисперсий оценок в методе регуляризации.

Смешанные вторые моменты вектора оценок параметров θ могут увеличиваться или уменьшаться в зависимости от конкретной задачи. Смешанный второй момент для θ и α равен нулю: оценки $\hat{\theta}$ и $\hat{\alpha}$ не коррелированы.

Из выражений (8.14) и (8.15) следует, что матрица вторых моментов вектора параметров θ является подматрицей, обратной матрице исходных уравнений, получаемой после дифференцирования функции Тихонова $T(\theta)$ по θ и α , а матрица вторых моментов α — той же подматрицей со знаком минус. Таким образом, дисперсии оценок $\hat{\theta}$ и $\hat{\alpha}$ можно определить с помощью матрицы $M(\theta, \alpha)$, элементы которой — вторые производные функции Тихонова по θ и α , взятые со знаком минус и вычисленные при найденных значениях оценок $\hat{\theta}$ и заданных значениях $\hat{\alpha}$. Дисперсия оценки $\hat{\alpha}$ показывает возможный интервал выбора параметра регуляризации. Обратив матрицу $M(\theta, \alpha)$, получим подматрицы, определяющие ковариационные матрицы (8.14) и (8.15) оценок $\hat{\theta}$ и $\hat{\alpha}$ соответственно.

Отметим, что значение константы R не влияет на значения получаемых дисперсий (8.14) и (8.15). Однако в методе регуляризации не применяется линейный стабилизатор. Для нахождения интервальных оценок можно, предварительно определив точечные оценки $\hat{\theta}$ и $\hat{\alpha}$ с помощью других известных методов [21, 71], воспользоваться только что полученными результатами, линеаризовав стабилизатор $\Omega(\theta)$ в окрестности оценки $\hat{\theta}$. Определив интервальные оценки, нетрудно непосредственно проверить, не вносит ли линеаризация стабилизатора искажений, превышающих погрешность исходных данных.

Пример. Найдём минимум функции Тихонова по θ и α :

$$T(\theta_1, \theta_2) = (\theta_1 - 1)^2 + (\theta_2 - 1)^2 + \alpha(\theta_1^2 + \theta_2^2) \rightarrow \min.$$

В этом выражении стабилизатор имеет вид

$$\Omega(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2.$$

Координаты точки минимума функции $T(\theta_1, \theta_2)$ при малых α близки к $(1, 1)$. Разложим функцию $\Omega(\theta_1, \theta_2)$ в ряд Тейлора в окрестности точки $(1, 1)$:

$$\Omega(\theta_1, \theta_2) \approx 2 + 2(\theta_1 - 1) + 2(\theta_2 - 1).$$

После замены переменных $\theta_1 \rightarrow \theta_1 - 1$, $\theta_2 \rightarrow \theta_2 - 1$ функция Тихонова будет иметь вид

$$T(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2 + \alpha(2\theta_1 + 2\theta_2 + 2).$$

Вычислив матрицу, элементы которой — вторые производные функции $T(\theta_1, \theta_2)$ по θ_1 , θ_2 и α , и обратив ее, получим матрицу вторых моментов оценок $\hat{\theta}$ и $\hat{\alpha}$:

$$M^{-1}(\theta, \alpha) = \begin{pmatrix} 0,25 & -0,25 & \\ -0,25 & 0,25 & \\ & & 0,25 \end{pmatrix}.$$

Элементы (1, 3), (2, 3), (3, 1) и (3, 2) матрицы здесь не указаны, так как они не несут информации (учитываются только соответствующие подматрицы). Значения элементов матрицы не зависят от величины константы в разложении стабилизатора $\Omega(\theta_1, \theta_2)$. Поэтому можно записать эквивалентную задачу следующим образом:

$$f(\theta_1, \theta_2) = \theta_1^2 + \theta_2^2 \rightarrow \min$$

при $2(\theta_1 + \theta_2) = \text{const} = C$. Отсюда имеем

$$f(\theta_1) = \theta_1^2 + \left(\frac{C}{2} - \theta_1\right)^2 \rightarrow \min, \quad \frac{d^2 f(\theta_1)}{d\theta_1^2} = 4,$$

дисперсии оценок $\hat{\theta}_1$ и $\hat{\theta}_2$ равны $1/4$, что совпадает с ранее полученными результатами. Без учета условия-ограничения дисперсии оценок равны $1/2$.

Вопрос о некорректности задачи получения оценок в конфлюэнтном анализе можно было бы рассмотреть в самом начале, при определении функционала, из которого находится конфлюэнтное решение задачи. К этому функционалу (см., например, (7.8), (7.9), (7.11)) следует добавить стабилизатор Ω . Если $\Omega = \Omega(\theta)$, то мы придем к изложенным здесь результатам.

Действительно, в общем случае функция Тихонова имеет вид

$$T = \ln L(x, y | \theta) + \alpha \Omega(\theta),$$

где $\Omega(\theta)$ — дифференцируемая функция. Тогда оценка вектора параметров θ при фиксированном значении $\hat{\alpha}$ находится из системы уравнений

$$\frac{\partial}{\partial \theta_j} [\ln L(x, y | \theta) + \hat{\alpha} \Omega(\theta)] = 0, \quad j = 1, 2, \dots, m.$$

Состоятельность и асимптотическая нормальность таких оценок $\hat{\theta}$ в регрессионной модели доказана в работе [71].

Дисперсии оценок $\hat{\theta}$ и $\hat{\alpha}$ определяются по формулам

$$D(\hat{\theta}) = A^{-1} - A^{-1}B(B^T A^{-1}B)^{-1}B^T A^{-1},$$

$$D(\hat{\alpha}) = (B^T A^{-1}B)^{-1},$$

где $A = \left\| \frac{\partial^2 T}{\partial \theta_i \partial \theta_j} \right\|$, $B = \left\| \frac{\partial^2 T}{\partial \theta_i \partial \alpha} \right\|$, $i, j = 1, 2, \dots, m$.

Смешанный второй момент оценок $\hat{\theta}$ и $\hat{\alpha}$ равен нулю: оценки $\hat{\theta}$ и $\hat{\alpha}$ не коррелированы.

С помощью матрицы вторых производных получим ковариационную матрицу оценок

$$D(\hat{\theta}, \hat{\alpha}) = \begin{pmatrix} A^{-1} - A^{-1}B(B^T A^{-1}B)^{-1}B^T A^{-1} & A^{-1}B(B^T A^{-1}B)^{-1} \\ (B^T A^{-1}B)^{-1}B^T A^{-1} & -(B^T A^{-1}B)^{-1} \end{pmatrix}, \quad (8.16)$$

но учитывать в (8.16) надо только диагональные элементы. Например, для задачи

$$T = (\theta_1 - 1)^2 + (\theta_2 - 1)^2 + \alpha(\theta_1^2 + \theta_2^2) \rightarrow \min$$

ковариационная матрица имеет вид

$$D(\hat{\theta}, \hat{\alpha}) = \begin{pmatrix} \frac{\hat{\theta}_2^2}{2(1+\hat{\alpha})(\hat{\theta}_1^2 + \hat{\theta}_2^2)} & \frac{-\hat{\theta}_1 \hat{\theta}_2}{2(1+\hat{\alpha})(\hat{\theta}_1^2 + \hat{\theta}_2^2)} & \hat{\theta}_2^2 \\ \frac{-\hat{\theta}_1 \hat{\theta}_2}{2(1+\hat{\alpha})(\hat{\theta}_1^2 + \hat{\theta}_2^2)} & \frac{\hat{\theta}_1^2}{2(1+\hat{\alpha})(\hat{\theta}_1^2 + \hat{\theta}_2^2)} & \\ & & -\frac{1+\hat{\alpha}}{2(\hat{\theta}_1^2 + \hat{\theta}_2^2)} \end{pmatrix}.$$

При $\hat{\theta}_1 = \hat{\theta}_2 = 1$ и $\hat{\alpha} \ll 1$ получим приведенную ранее матрицу $M^{-1}(\theta, \alpha)$. Из выражения для $D(\hat{\theta}, \hat{\alpha})$ следует, что с увеличением значения $\hat{\alpha}$ (с увеличением вклада ограничений) дисперсии оценок параметров уменьшаются.

В ряде работ показано (см., например, [21]), что с помощью методов безусловной оптимизации функции $J(\theta)$, в частности методов сопряженных градиентов, наискорейшего (градиентного) спуска, можно получить регуляризованное решение, если ограни-

чить число итераций $n \leq n_0$ поиска экстремума таким образом, чтобы при выполнении условия $J(\theta) = \delta$, где δ определяется погрешностью исходных данных, процесс минимизации прекращался. Полученное решение и будет регуляризованным. При увеличении числа итераций $n > n_0$ решение может стать неустойчивым. Параметром регуляризации здесь является число итераций n_0 . Подобный подход привлекает своей простотой. Для определения дисперсий и оценок в этом случае необходимо учесть вид уравнения траектории движения от начального приближения к решению при условии, что в точке регуляризованного решения значения первых производных функционала малы по сравнению со значениями вторых производных.

§ 8.4. Классификация образов по измеренному с ошибкой вектору признаков

Рассмотрим процедуру распознавания образов — второй этап задачи классификации. Пусть нам известны все характеристики наблюдавшихся случайных величин x_1, x_2, \dots, x_n , по которым на первом этапе нашли вид функции распределения $P(\xi, \theta | \omega_j)$, $j = 1, 2, \dots, m$, эмпирические значения плотностей вероятностей $p_i(\xi | \omega_j)$, $i = 1, 2, \dots, n$, точечную и интервальную оценки разделяющей функции $\Psi(\xi, \theta)|_{\xi=x}$. Требуется провести измерение случайной величины (вектора признаков) x , по которой должен быть классифицирован объект $\omega_j \in \Omega$. Считаем, что интервальная оценка для случайной величины x известна.

Прежде всего отметим, что с определением интервальной оценки разделяющей функции появляется нулевая зона — зона неопределенности. Если наблюдавшееся значение x (а точнее, ее «истинное значение» ξ) попадает в эту зону, то требуется, строго говоря, следующее дополнительное наблюдение вектора признаков. Получив наблюдение x , необходимо посмотреть, не пересекается ли допустимый интервал значений случайной величины x с интервальной оценкой разделяющей функции. Если пересечения нет, применяется традиционный метод. В противном случае надо предварительно найти «истинное значение» наблюдавшегося признака ξ . Однако по единственному измерению x нельзя найти со-

ответствующие оценки признака ξ . Поэтому необходимо результаты наблюдений x присовокупить к имеющимся результатам наблюдений, полученным в процессе обучения системы распознавания, т. е. к данным, характеризующим каждый объект $\omega_j \in \Omega$.

Оценку $\hat{\xi}$ получим из уравнения (8.4), допуская, что случайная величина x может принадлежать двум соседним распределениям, описывающим разные классы, т. е. получим $\hat{\xi}_1$ и $\hat{\xi}_2$. В большинстве случаев, например для функций плотности распределения Гаусса, условие (8.4) приводит к трансцендентному уравнению, которое может быть решено только численными методами. Для численных методов полезно использовать дополнительное ограничение вида

$$\hat{\xi}_i \in [x - 3\sigma(x), x + 3\sigma(x)], \quad i = 1, 2,$$

где $\sigma(x)$ — среднее квадратическое отклонение для наблюдавшейся случайной величины x .

Вблизи решающих функций (границ раздела) большинство функций распределения хорошо аппроксимируются гиперплоскостями (в \mathbb{R}^2 — прямыми линиями). Так, для одномерных гауссовых плотностей условие (8.4) примет вид

$$\frac{p_i - p_i(\hat{\xi}_i | \omega_i)}{\sigma^2(\hat{p})} p_{\xi_i}(\hat{\xi}_i | \omega_i) + \frac{x - \hat{\xi}_i}{\sigma^2(x)} = 0, \quad i = 1, 2,$$

где

$$p_i(\hat{\xi}_i | \omega_i) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \exp \left\{ -\frac{1}{2} \frac{(m_i - \hat{\xi}_i)^2}{\sigma^2} \right\},$$

$$p_{\xi_i}(\hat{\xi}_i | \omega_i) = p_i(\hat{\xi}_i | \omega_i) \frac{m_i - \hat{\xi}_i}{\sigma^2}, \quad i = 1, 2,$$

p_i определяют по результатам всех наблюдений x_1, x_2, \dots, x_n (например, путем группировки).

Разложим «теоретическую» функцию плотности $p_i(\hat{\xi}_i | \omega_i)$ в ряд в окрестности наблюдаемой точки x . Получим уравнение прямой

$$p_i(\hat{\xi}_i | \omega_i) \approx p_i(x) \left(1 + \frac{x - m_i}{\sigma^2} x \right) + p_i(x) \frac{x - m_i}{\sigma^2} \hat{\xi}_i.$$

Обозначим

$$p_i(x) \left(1 + \frac{x - m_i}{\sigma^2} x \right) \equiv a_i, \quad p_i(x) \frac{x - m_i}{\sigma^2} \equiv b_i$$

и получим в окрестности точки x уравнение $p_i(\hat{\xi}_i | \omega_i) = a_i + b_i \hat{\xi}_i$. Тогда оценки $\hat{\xi}_i$ найдем из условия

$$\hat{\xi}_i = \frac{(\hat{p}_i - a_i)b_i + x}{\sigma^2(\hat{p}) + b_i^2 \sigma^2(x)}, \quad i = 1, 2.$$

По значениям $\hat{\xi}_1$ и $\hat{\xi}_2$, соответствующим первому и второму распределениям, проводим классификацию (если ξ_1 и ξ_2 не попадают в зону неопределенности, то требуется еще одно дополнительное наблюдение). В случае, когда обстоятельства не позволяют провести дополнительные измерения, принимают решение по наиболее вероятному значению $\hat{\xi}_i$, $i = 1, 2$.

Рассмотренный в § 8.2 и § 8.4 алгоритм решения статистической задачи распознавания образов (статистической задачи решения) позволяет полностью учесть статистическую природу результатов наблюдений как на этапе нахождения условных плотностей распределения вероятностей и разделяющих функций, так и на этапе классификации по результатам наблюдений вектора признаков.

В задачах распознавания образов для принятия решения, когда применяется решающая процедура с фиксированным объемом выборки, используется критерий отношения правдоподобия. Рассмотрим, как изменится этот критерий при учете неопределенности исходной информации.

Согласно традиционному подходу определим отношение правдоподобия между классами ω_i и ω_j следующим образом:

$$\lambda = \frac{p(x | \omega_i)}{p(x | \omega_j)}, \quad \text{или} \quad \ln \lambda = \ln \frac{p(x | \omega_i)}{p(x | \omega_j)}, \quad i \neq j.$$

Тогда, применяя байесовское решающее правило, получим $d^* = d_i$, т. е. $x \sim \omega_i$, если $\lambda \geq \frac{P(\omega_j)}{P(\omega_i)}$, или $\ln \lambda \geq \ln \frac{P(\omega_j)}{P(\omega_i)}$, $i, j = 1, 2, \dots, m$, $i \neq j$.

Этот же результат следует из критерия Неймана—Пирсона, согласно которому при данном числе наблюдений n оптимальное

решение задачи выбора гипотезы H_1 о том, что $x \sim \omega_1$, против гипотезы H_2 о том, что $x \sim \omega_2$, с вероятностью, не меньшей $1 - \alpha$, если верна гипотеза H_1 , и с вероятностью, не меньшей $1 - \beta$, если верна гипотеза H_2 , определяется отношением правдоподобия λ_n , имеющим вид

$$\lambda_n = \prod_{i=1}^n \frac{p(x_i | \omega_1)}{p(x_i | \omega_2)} = \frac{p_n(x | \omega_1)}{p_n(x | \omega_2)}.$$

Здесь α — ошибка первого рода, β — ошибка второго рода.

Этот критерий обеспечивает наименьшую ошибку β (наиболее мощный критерий).

Рассмотрим, как изменятся результаты решения данной задачи при известных оценках плотностей $p(x | \omega_1)$ и $p(x | \omega_2)$:

$$\lambda = \frac{p(x | \omega_i) \pm \Delta p_i}{p(x | \omega_j) \pm \Delta p_j},$$

или

$$\ln \lambda = \ln \frac{p(x | \omega_i) \pm \Delta p_i}{p(x | \omega_j) \pm \Delta p_j}. \quad (8.17)$$

В последнем случае (см. (8.7)) решение $x \sim \omega_i$ имеет место, если

$$\ln \lambda \geq \ln \frac{P(\omega_j)}{P(\omega_i)}, \quad i, j = 1, 2, \dots, m, \quad i \neq j. \quad (8.18)$$

Перепишем (8.17) в следующем виде:

$$\ln \lambda = \ln \frac{p(x | \omega_i) \pm \Delta p_i}{p(x | \omega_j) \pm \Delta p_j} \cong \ln \frac{p(x | \omega_i)}{p(x | \omega_j)} \pm \frac{\Delta p_i}{p(x | \omega_i)} \mp \frac{\Delta p_j}{p(x | \omega_j)}.$$

Тогда вместо условия (8.18) имеем

$$\ln \frac{p(x | \omega_i)}{p(x | \omega_j)} \pm \frac{\Delta p_i}{p(x | \omega_i)} \mp \frac{\Delta p_j}{p(x | \omega_j)} \geq \ln \frac{P(\omega_j)}{P(\omega_i)}. \quad (8.19)$$

Учитывая, что погрешности Δp_i и Δp_j могут быть разных знаков, для надежной идентификации образов вместо условия (8.19) следует использовать условие

$$\ln \frac{p(x | \omega_i)}{p(x | \omega_j)} + \frac{|\Delta p_i|}{p(x | \omega_i)} + \frac{|\Delta p_j|}{p(x | \omega_j)} \geq \ln \frac{P(\omega_j)}{P(\omega_i)}.$$

Два последних слагаемых в левой части неравенства определяют нулевую зону (зону неопределенности в принятии решений) в задачах распознавания образов при фиксированном объеме выборки.

§ 8.5. Классификация летательных аппаратов с учетом погрешностей в измерениях признаков

Рассмотрим задачу классификации объектов по ряду признаков. Объектами распознавания являются различные типы (классы) летательных аппаратов (л. а.): тип 1, тип 2 и т. д. Преждеположим, что в результате активной радиолокации были получены следующие радиолокационные характеристики:

- 1) эффективная площадь рассеяния (ЭПР);
- 2) спектральные и временные характеристики отраженных от объектов сигналов.

Характеристики были измерены радиолокационной станцией (РЛС) с длиной волны 10 м.

При использовании этих данных были выделены признаки, определяющие л. а. Такими признаками являлись: скорость движения объекта v , высота полета H , частота модуляции отраженного сигнала f_m и значения ЭПР, измеренные при различных ракурсах.

Способы вычисления скорости движения и высоты полета л. а. по спектральным и временным характеристикам отраженных от объектов сигналов описаны в [19, 65]. Модуляция отраженного сигнала вызывается вращением турбин или винтов л. а. Частота модуляции f_m зависит от типа двигателя (винтовой или реактивный) и от количества двигателей.

Эффективная площадь рассеяния является оценкой интенсивности вторичного излучения л. а. и зависит от диэлектрической и магнитной проницаемости материала, из которого изготовлен л. а., от ракурса, под которым наблюдается л. а., рабочей частоты РЛС, от формы и размеров л. а., от поляризации приемной и передающей антенн РЛС [56–58].

Априори известно, что признаки статистически независимы, плотности распределения каждого признака каждого класса соответствуют нормальному закону распределения $N(\mu, \sigma^2)$ и априорные вероятности классов равны $P(\omega_j) = 1/5$, $j = 1, \dots, 5$, j — число классов.

В табл. 8.2 приведены значения числовых характеристик плотностей вероятности каждого признака для каждого класса л. а.

Таблица 8.2

**Значения числовых характеристик
условных плотностей вероятности признаков**

Тип летательного аппарата	Скорость движения v , м/с		Высота полета H , м		Частота модуляции f_m , кГц	
	μ	σ	μ	σ	μ	σ
1	800	200	2000	1583	9,5	1,3
2	1500	366	10 000	1816	15	1,6
3	400	100	500	491	1,5	0,25
4	200	100	500	250	15	0,2
5	550	100	10 000	166	—	—

Зависимости ЭПР от ракурса л. а. при длине волны $\lambda = 10$ м и горизонтальной поляризации для различных классов л. а. приведены на рис. 8.2. Величины ЭПР нормированы относительно значения 10^{-5} м^2 . Из рис. 8.2 можно сделать следующий вывод: если учесть погрешности измерений, то различные летательные аппараты будут не различимы.

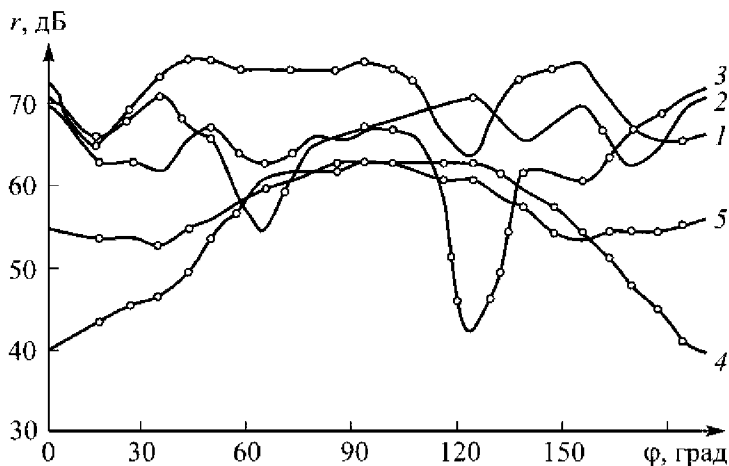


Рис. 8.2. Зависимость ЭПР летательных аппаратов от ракурса при длине волны $\lambda = 10$ м (номер линии на графике соответствует типу летательного аппарата)

Необходимо было, используя данную информацию, осуществить распознавание конечного числа классов по результатам измерений признаков. Критерием распознавания является вероят-

ность ошибки, которая не должна превышать заданной вероятности $P(e)_3 = 0,1$. Вероятность ошибки $P(e)$ определяется по следующей формуле [28]:

$$P(e) = 1 - \sum_{i=1}^s \int_{R_i} p(x | \omega_i) P(\omega_i) dx,$$

где R_i — область принятия решений для i -й гипотезы, s — число объектов.

Радиолокационные характеристики, как и любые измерения, содержат случайные ошибки. Погрешности радиолокационных измерений подразделяются на ошибки, вносимые л. а., и ошибки, вносимые аппаратурой. Основные ошибки, вносимые л. а., происходят вследствие флуктуации амплитуды сигнала, а также зависят от состояния окружающей среды. Из числа ошибок, вносимых аппаратурой, наибольшее значение имеют ошибки, появляющиеся вследствие собственных шумов приемного устройства. Если заданы параметры РЛС, а также характеристики л. а. и окружающей среды, уровень ошибки распознавания для определенной точки пространства обзора РЛС может быть рассчитан как средняя квадратическая сумма всех отдельных составляющих, вычисленных для этой точки. Однако нас интересует определение ошибок во всей зоне обзора РЛС. Для упрощения вычислений в работах [19, 65] предложена методика расчета оптимальной точности, при которой суммарная ошибка не зависит от погрешностей, вносимых л. а., а оценивается по проектным данным или данным испытаний РЛС.

В данной задаче ошибки измерения по каждому признаку определялись ошибками, вносимыми РЛС, и для таких признаков, как скорость движения, высота полета и частота модуляции, средние квадратические погрешности соответствуют следующим значениям: $\sigma(v) = 100$ м/с, $\sigma(H) = 250$ м, $\sigma(f_m) = 0,2$ кГц. Предполагалось, что средняя квадратическая погрешность измерения ЭПР составляет 10 дБ и одинакова при всех ракурсах; точность определения ракурса составляет 10° .

С учетом погрешностей признаков числовые характеристики условных плотностей вероятности изменяются (табл. 8.3). В табл. 8.3 приводятся также средние квадратические отклонения числовых характеристик условных плотностей распределения вероятностей, с помощью которых легко вычислить интервальные оценки функций условных плотностей распределения вероятностей в интересующих нас областях.

Таблица 8.3

Значения числовых характеристик и их средних квадратических отклонений с учетом погрешности наблюдений

Тип летательно-го аппарата	Скорость движения v , м/с				Высота полета H , м				Частота модуляции f_m , кГц			
	μ	$\Delta\mu$	σ	$\Delta\sigma$	μ	$\Delta\mu$	σ	$\Delta\sigma$	μ	$\Delta\mu$	σ	$\Delta\sigma$
1	720	70	230	40	2150	205	1630	250	10,2	0,9	1,25	0,21
2	1610	120	340	65	9100	850	1750	280	13,5	1,2	1,7	0,3
3	420	30	105	20	465	37	410	95	1,6	0,1	0,24	0,04
4	190	20	97	21	515	46	270	52	15,5	1,4	0,18	0,02
5	510	55	112	22	1080	980	156	29	—	—	—	—

Анализируя рис. 8.2, получаем, что наилучшим образом классы л. а. по ЭПР разделяются при ракурсах 30° , 60° , 120° . Возьмем в качестве признаков эти значения ракурсов (табл. 8.4).

Таблица 8.4

Значения числовых характеристик условной функции плотности ЭПР для каждого класса л. а.

Тип летательного аппарата	$\varphi = 30^\circ$		$\varphi = 60^\circ$		$\varphi = 120^\circ$	
	μ	σ	μ	σ	μ	σ
1	73	10	74	10	64	10
2	71	10	55	10	71	10
3	62	10	63	10	43	10
4	47	10	61	10	63	10
5	53	10	60	10	61	10

Для вычисления вероятности ошибки воспользуемся методом, предложенным в [14, 17–19].

При большой размерности пространства признаков непосредственное интегрирование значительно усложняется, поскольку усложняется выбор пределов интегрирования. Так, например, для получения вероятности ошибки и ее интервальных оценок при использовании n -признаков необходимо вычислять n -мерные интегралы

$$P(e) = I - \sum_{j=1}^s \int_{R_j} \dots \int p(x | \omega_j) P(\omega_j) dx_1 \dots dx_n,$$

где R_j — область пространства \mathbb{R}^n , которая характеризует образ ω_j .

Таким образом, непосредственное вычисление вероятности ошибки (в. о.) описанным способом вызывает значительные затруднения.

В работе [14] рассмотрен метод вычисления вероятности ошибки распознавания классов, использующий каждый признак в отдельности. Формула вычисления в. о. для данного метода имеет вид

$$P(e) = \sum_{i=1}^T [P(x_i) - \max P(\omega_j) p(x_i | \omega_j)], \quad (8.20)$$

где $P(x_i) = \sum_{j=1}^s P(\omega_j) p(x_i | \omega_j)$, s — общее число классов; T — количество реализаций, $T = R^N$, R — число значений признаков, N — общее число признаков.

В случае, когда функции плотности соответствуют многомерному нормальному распределению, использование формулы (8.20) дает возможность перейти к серии одномерных нормальных распределений и использовать для них линеаризующее преобразование.

Границы интервальной оценки вероятности ошибки в данном случае будут определяться по следующим формулам:

$$\begin{aligned} P_H(e) &= \sum_{i=1}^T [P_H(x_i) - \max_{\omega} P(\omega_j) p_H(x_i | \omega_j)] = \\ &= \sum_{i=1}^T \left[\sum_{j=1}^s P(\omega_j) p_H(x_i | \omega_j) - \max_{\omega} P(\omega_j) p_H(x_i | \omega_j) \right], \\ P_R(e) &= \sum_{i=1}^T \left[\sum_{j=1}^s P(\omega_j) p_R(x_i | \omega_j) - \max_{\omega} P(\omega_j) p_R(x_i | \omega_j) \right]. \end{aligned}$$

Использование в рассматриваемом случае одного любого признака дает в. о. больше заданной $P_3(e) = 0,1$. Например, в. о. при использовании одного признака — скорости — составляет $P_1(e) = 0,252$, $P_1(e) > P_3(e)$.

Как указывалось ранее, в процедуре идентификации объектов при известных точечных и интервальных оценках функции плот-

ности распределения вероятностей по наблюдаемому вектору признаков x_n следует определить оценки $\hat{\xi}_n$ и величину эллиптической (гиперэллипсоидной) области, которая покрывает истинное значение ξ_n с доверительной вероятностью β . Эта область резко увеличивается с возрастанием числа координат вектора признаков ξ_n . Отклонение от наблюдаемого значения x_n по каждой координате равно $k_\beta \sigma(x)$: для доверительной вероятности $\beta = 0,683$ и для одной координаты x_n имеем $k_\beta = 1$, для двух координат — $k_\beta \cong 2$, для трех координат — $k_\beta \cong 3,4$, для пяти координат — $k_\beta \cong 6$, или, другими словами, колебание в одно среднее квадратическое отклонение для одного параметра имеет доверительную вероятность $\beta = 0,683$, для двух параметров — $\beta = 0,38$, для трех параметров — $\beta = 0,20$ и т. д.

Другой пример распознавания образов при наблюдении двух признаков показан на рис. 8.3. В традиционном подходе решение принимается по значениям координат (x_1, x_2) , что приводит к не-

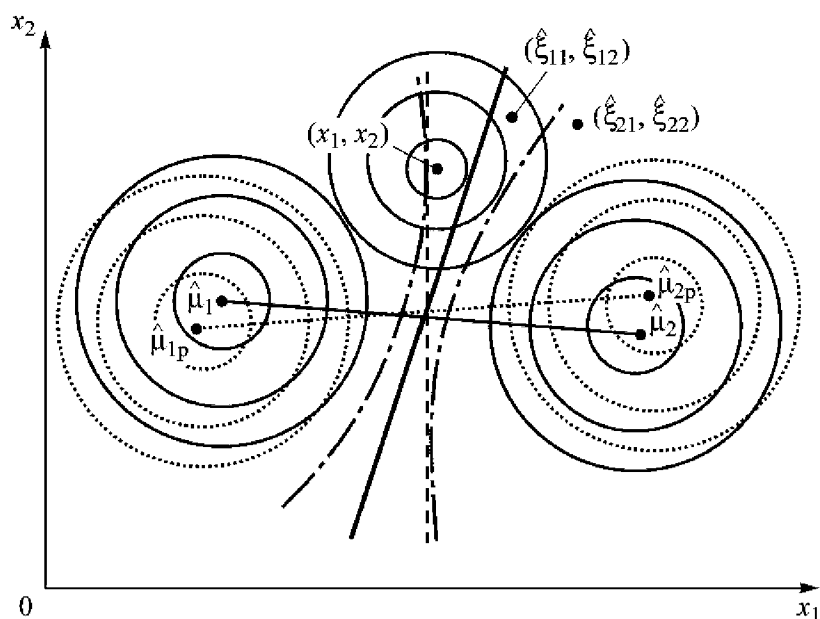


Рис. 8.3. Распознавание образов с учетом погрешностей наблюдений признаков при известном с точностью до параметров виде функций условных плотностей вероятности:

..... — без учета погрешностей признаков; — — — — с учетом погрешностей признаков; — · — · — — интервальная оценка решающей функции

верному решению. Решение следует принимать по оценкам $\hat{\xi}$. Строго говоря, в данном случае решение принято быть не может, поскольку и наблюдаемые значения (x_1, x_2) , и оценки $\hat{\xi}$ попали в область неопределенности принятия решений (нулевую зону).

Проведем идентификацию л. а. при одновременно наблюдаемых признаках $v \cong 1000$ м/с, $H \cong 5500$ м, $f_m = 12$ кГц, ЭПР = 69 дБ. Традиционные методы идентифицируют объект как объект типа 1, тогда как с учетом погрешности измерений признаков это объект типа 2 при вероятности ошибки $p_{\text{ош}} = 0,23$ со средним квадратическим отношением вероятности ошибки $\sigma(p_{\text{ош}}) = 0,03$. Только привлечение дополнительных значений ЭПР позволило уменьшить вероятность ошибки до 0,1.

РАСПОЗНАВАНИЕ ОБРАЗОВ ПРИ НЕИЗВЕСТНОМ ЗАКОНЕ РАСПРЕДЕЛЕНИЯ ЗНАЧЕНИЙ ПРИЗНАКОВ

§ 9.1. Оценка параметров классификаторов по выборке фиксированного объема

Ранее, в гл. 6 и гл. 8, было показано, что в задачах распознавания образов необходимо учитывать:

1) погрешности измеряемых признаков при определении условных плотностей вероятности признаков и при идентификации объектов (в противном случае мы получаем смещенные оценки функции условной плотности распределения вероятностей и принимаем недостоверные решения);

2) не только наиболее вероятные оценки функций условной плотности распределения вероятностей и решающих функций, но и их интервальные оценки (тем самым показывается, что в любой задаче распознавания образов имеется нулевая зона (зона неопределенности), задаваемая, в частности, интервальной оценкой решающей функции).

В гл. 8 были описаны разработанные методы учета статистических характеристик вектора признаков в задачах распознавания образов с фиксированным объемом выборки в предположении, что вид функции условной плотности распределения вероятностей $p(\xi, \theta | \omega)$ вектора признаков ξ для класса ω нам известен, но не известны значения параметров θ и оценки $\hat{\xi}$ (параметрические методы).

Кроме того, в задачах распознавания образов часто имеют место случаи, когда [3, 14, 25, 28, 81, 83–85]:

а) вид функции распределения вероятностей признаков не известен, но статистика наблюдений достаточна для установления этого вида (непараметрические методы);

б) вид функции распределения вероятностей признаков не известен, но полагают известным вид (или виды) разделяющих (ре-

шающих) функций, а выборки используют для оценок значений параметров классификатора.

Последняя ситуация будет рассмотрена в настоящей главе.

Классификатор по минимуму расстояния

Пусть имеются выборки для c классов и для каждого класса определены векторы средних значений μ_i , $i = 1, 2, \dots, c$. Чтобы по наблюдаемому вектору признаков x определить класс, следует измерить расстояния $\|x - \mu_i\|$ от x до каждого из c векторов средних значений и выбрать класс, соответствующий ближайшему среднему значению. Если каждый из векторов средних значений считать идеальным прототипом или эталоном для образов своего класса, то это будет процедурой сравнения с эталоном.

Расстояние $r = \|x - \mu_i\|$ можно определить по-разному (разными формулами), но мы будем использовать евклидово расстояние. Если априорные вероятности не равны, то квадрат расстояния $\|x - \mu_i\|^2$ должен быть нормирован по дисперсии σ^2 и смещен на величину $\ln P(\omega_i)$. (Это следует из нормального закона распределения случайных величин x при равных дисперсиях $\sigma^2 I$.) Когда вектор x равно близок к двум различным векторам средних значений, при принятии решения следует предпочесть класс, априори более вероятный.

Если ковариационные матрицы K для всех классов одинаковы и признаки подчинены нормальному закону распределения, то для классификации вектора признаков x следует определить квадратичное махаланобисово расстояние

$$r^2 = (x - \mu_i)^T K^{-1} (x - \mu_i)$$

от x до каждого из c векторов средних значений и отнести x к классу, соответствующему ближайшему среднему значению μ_i . В случае неравных априорных вероятностей при принятии решения несколько большее предпочтение ($\ln P(\omega_i)$) отдается классу, априори более вероятному.

Это один из наиболее простых критериев, который, однако, не учитывает возможные виды разделяющих функций, не говоря уже о том, что не учитываются погрешности векторов μ_i , $i = 1, 2, \dots, c$, и x и, тем самым, не определена область неопределенности принимаемых решений.

Линейная разделяющая функция для двух и многих классов

Линейные разделяющие функции наиболее просты и удобны с точки зрения аналитического исследования. Часто их используют ради упрощения вычислительного процесса.

Задача определения линейной разделяющей функции формулируется как задача минимизации некоторой функции критерия, в качестве которого используется *выборочный риск* — средние потери при классификации множества конструктивных выборок, хотя могут использоваться и другие критерии.

Линейная разделяющая функция $g(x)$ для двух классов может быть записана в виде

$$g(x) = wx + w_0,$$

где w — весовой вектор, w_0 — величина порога.

Для двух классов ω_1 и ω_2 применяется следующее решающее правило: принять решение ω_1 , если $g(x) > 0$, и принять решение ω_2 , если $g(x) < 0$.

В линейном случае уравнение $g(x) = 0$ представляет гиперплоскость H , w — нормаль этой гиперплоскости, направленная в сторону области решений R_1 для ω_1 . Разделяющая функция $g(x)$ должна представлять собой алгебраическое расстояние от x до гиперплоскости H . Поэтому предпочтительнее сразу пользоваться нормальным (нормированным) уравнением гиперплоскости, для чего общее уравнение гиперплоскости следует разделить на длину вектора w , равную $\|w\|$. Тогда расстояние r от точки x до гиперплоскости H выразится формулой

$$r = \frac{g(x)}{\|w\|}.$$

При $w_0 = 0$ гиперплоскость проходит через начало координат. Если имеется не два, а c классов, то задачу можно свести к $c-1$ задачам для двух классов, где решением i -й задачи служит линейная разделяющая функция, определяющая границу между точками, соответствующими решению ω_i , и точками, не соответствующими решению ω_i .

Для решения многомерной задачи можно также использовать $(c-1)c/2$ линейных разделяющих функций, по одной для каждой пары классов.

В обоих подходах могут образовываться области, в которых классификация не определена. Чтобы избежать этой неопределенности, применяется классификатор, называемый *линейной машиной*. Этим классификатором определяются c линейных разделяющих функций

$$g_i(x) = w_i^T x + w_{i0}, \quad i = 1, 2, \dots, c;$$

по x определяется класс ω_i , если $g_i(x) > g_j(x)$ для всех $j \neq i$.

Если области решений R_i и R_j соприкасаются, то границей между ними будет часть гиперплоскости H_{ij} , определяемой соотношением

$$g_i(x) = g_j(x)$$

или

$$(w_i - w_j)^T x + w_{i0} - w_{j0} = 0.$$

Здесь $(w_i - w_j)$ — вектор нормали, а расстояние от x до плоскости H_{ij} определено формулой

$$r = \frac{g_i - g_j}{\|w_i - w_j\|}.$$

Параметрические и непараметрические методы эффективно используют для решения задач в пространствах меньшей размерности. Поэтому при распознавании образов, которые имеют признаки размерностью d , применяется прием, позволяющий уменьшить размерность до одного: проецируются d -мерные данные на прямую (для двух классов).

Пусть имеется множество $(n \times d)$ -мерных выборок $x_1, x_2, \dots, x_n \in X$, из которых n_1 выборок принадлежат подмножеству X_1 , помеченному ω_1 , и n_2 лежат в подмножестве X_2 , помеченном ω_2 .

Рассмотрим линейную комбинацию компонент вектора x :

$$y = w^T x, \quad \text{или} \quad y = w_1 x_1 + \dots + w_n x_n.$$

Если длина $\|w\| = 1$, то каждая компонента y_i есть проекция соответствующего x_i на прямую, сонаправленную с вектором w . Доказано, что наилучшее направление вектора w совпадает с направлением прямой, проходящей через точки m_1 и m_2 , где

$$m_i = \frac{1}{n_i} \sum_{x \in X_i} x, \quad i = 1, 2.$$

На плоскости имеем

$$x = (x_1, x_2)^T, \quad m_i = (m_{i1}, m_{i2})^T.$$

Спроецируем множества X_1 и X_2 на прямую с направляющим вектором w , получим множество Y , состоящее из n выборок y_1, y_2, \dots, y_n и разделенное на подмножества Y_1, Y_2 ($Y_1 \subset Y, Y_2 \subset Y$).

Вместо идентификации образов по выборке $x \in X$ проводят идентификацию по проекции $y \in Y$ и вводят понятие линейного дискриминанта Фишера. Среднее значение выборки для спроецированных точек обозначим \tilde{m} .

Линейный дискриминант Фишера определяется как такая линейная разделяющая функция $y = w^T x$, для которой функция критерия

$$J(w) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{s_1^2 + s_2^2}$$

максимальна, где

$$s_i^2 = \sum_{y \in Y_i} (y - \tilde{m}_i)^2, \quad \tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y = \frac{1}{n_i} \sum_{x \in X_i} w^T x = w^T m_i, \quad i = 1, 2.$$

В итоге получают

$$w = S_w^{-1}(m_1 - m_2),$$

где S_w — матрица разброса внутри класса, она пропорциональна ковариационной выборочной матрице для совокупности d -мерных данных,

$$S_w = S_1 + S_2, \quad S_i = \sum_{x \in X_i} (x - m_i)(x - m_i)^T, \quad i = 1, 2.$$

Например, известно, что для нормально распределенных случайных величин с равными ковариационными матрицами K и математическими ожиданиями μ_1 и μ_2 линейная разделяющая функция имеет вид

$$w = K^{-1}(\mu_1 - \mu_2).$$

В случае c классов обобщение линейного дискриминанта Фишера требует определения $c-1$ разделяющих функций; здесь осуществляется проекция из d -мерного пространства X на $(c-1)$ -мерное пространство Y (полагаем $d \geq c$). Линейные разделяющие функции имеют направления $m_i - m$, где

$$m_i = \frac{1}{n_i} \sum_{x \in X_i} x, \quad m = \frac{1}{n} \sum_{x \in X_i} x = \frac{1}{n} \sum_{i=1}^c n_i m_i.$$

При малой статистике каждое новое наблюдение, по которому ведется идентификация образов, должно быть объединено с прежними наблюдениями. Чтобы выявить изменение решающей функции (или другого критерия), необходимо сравнить полученную разделяющую функцию с прежней.

§ 9.2. Обобщенные линейные разделяющие функции

Линейная разделяющая функция имеет вид

$$g(x) = w^T x + w_0 \equiv w_0 + \sum_{i=1}^d w_i x_i,$$

где w_i — компоненты весового вектора w , d — размерность признаков x .

Можно ввести квадратичную разделяющую функцию

$$g(x) = w_0 + \sum_{i=1}^d w_i x_i + \sum_{i=1}^d \sum_{j=1}^d w_{ij} x_i x_j,$$

причем можно считать, что $w_{ij} = w_{ji}$, поскольку $x_i x_j = x_j x_i$. Для квадратичной разделяющей функции разделяющая поверхность, определяемая уравнением $g(x) = 0$, является поверхностью второго порядка, или гиперквадрикой. Продолжая вводить в уравнение для $g(x)$ дополнительные члены вида $w_{ijk} x_i x_j x_k$, получим класс полиномиальных разделяющих функций, которыми можно аппроксимировать разделяющую функцию $g(x)$ любого вида. Однако хотелось бы остаться в классе линейных функций, поэтому вводят понятие *обобщенных линейных разделяющих функций*, которые имеют следующий вид:

$$g(x) = \sum_{i=1}^d a_i y_i(x),$$

или

$$g(x) = a^T y(x), \quad y(x) = (y_1(x), y_2(x), \dots, y_d(x))^T,$$

где a_i — компоненты d -мерного весового вектора a , $y_i(x)$ — произвольные функции от x .

Пример. Пусть $g(x) = a_1 + a_2x + a_3x^2$; тогда обобщенная линейная разделяющая функция имеет вид $g(x) = a_1y_1 + a_2y_2 + a_3y_3$, где $y_1 = 1$, $y_2 = x$, $y_3 = x^2$, или $y = (1, x, x^2)^T$.

Однородная разделяющая функция $a^T y = 0$ разделяет точки в данном отображенном пространстве посредством гиперплоскости, проходящей через начало координат (что является положительным фактом). Однако увеличение размерности d при переходе к обобщенной функции затрудняет практическое использование этого приема, хотя переход к обобщенной функции удобен при рассмотрении линейных разделяющих функций. В этом случае

$$y = (1, x_1, \dots, x_d)^T = (1, x)^T, \quad a = (w_0, w_1, \dots, w_d)^T = (w_0, w)^T.$$

Переход от d -мерного пространства X к $(d+1)$ -мерному пространству Y не нарушает соотношений в расстояниях между выборками, но гиперплоскость $a^T y = 0$ проходит через начало координат пространства Y ; расстояние от y до гиперплоскости равно $|a^T y| / \|a\|$.

Для нахождения обобщенной линейной разделяющей функции по результатам наблюдений необходимо определить весовой вектор a . Пусть имеется множество n выборок векторов y_1, \dots, y_n , одна часть которых относится к образу ω_1 , а другая часть — к образу ω_2 . Эти выборки мы хотели бы использовать для определения весового вектора a , который правильно классифицировал бы все выборки. Если такой вектор существует, то выборки называются *линейно разделяемыми*. Выборка y_i классифицируется правильно, если $a^T y_i > 0$ и y_i помечен ω_1 или если $a^T y_i < 0$, но y_i помечен ω_2 . Во втором случае y_i будет также классифицироваться правильно, если $a^T(-y_i) > 0$. Последнее условие дает возможность ввести понятие нормирования для случая двух классов: проводится замена знаков всех выборок, относящихся к классу ω_2 . При введении нормирования можно найти весовой вектор a , для которого выполнено условие: если $a^T y_i > 0$, то y_i помечен ω_1 для всех векторов выборки. Данный вектор a называется *разделяющим вектором*, или *вектором решения*. Каждая выборка y_i налагает ограничение на возможное расположение вектора решения. Уравнение $a^T y_i = 0$ определяет гиперплоскость, проходящую через начало координат в весовом пространстве, для которой y_i является нормальным вектором и должен находиться с положительной стороны гиперплоскости (рис. 9.1).

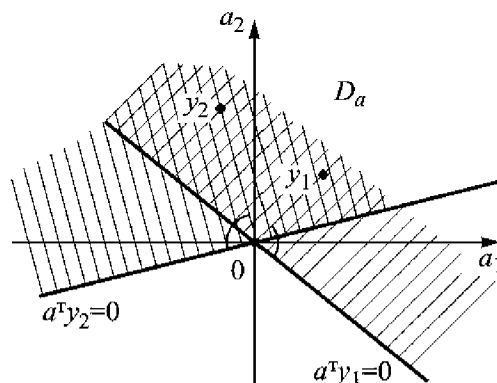


Рис. 9.1. Область решений в задаче распознавания образов

Система неравенств, образуемая выборками y_1, y_2, \dots, y_n , или неравенствами $a^T y_i > 0$, $i = 1, 2, \dots, n$, определяет область решений D_a , но сам вектор решения определяется неоднозначно. Например, вектор решения может быть выбран как вектор, минимизирующий расстояния от выборок до разделяющей плоскости. Другими словами, необходимо определить критерий (целевую функцию $J(a)$), которому должен удовлетворять вектор решения.

В качестве целевых функций используются:

1) персептронная функция критерия

$$J_p(a) = \sum_{y \in Y} (-a^T y),$$

где Y — множество выборок, классифицируемых с ошибкой, если разделяющим вектором является a (эта функция не может быть отрицательной и достигает нулевого значения, когда a является вектором решения);

2) другие функции критерия, например

$$J_q(a) = \sum_{y \in Y} (a^T y)^2, \quad J_r(a) = \frac{1}{2} \sum_{y \in Y} \frac{(a^T y - b)^2}{\|y\|^2},$$

$$J_s(a) = \sum_{i=1}^n (a^T y_i - b_i)^2,$$

где вектор b — некоторый допуск.

Процесс определения вектора решения a часто является итерационным, построенным на градиентном спуске:

$$\begin{cases} a_1 \text{ произвольно,} \\ a_{k+1} = a_k - p_k \nabla J(a_k), \quad k \geq 1. \end{cases}$$

При этом должны выполняться условия: $a_k^T y^k \leq 0$ для всех k или $a_i^T y^k \leq b$ для всех k при заданном допуске b . Здесь $\nabla J(a_k)$ — градиент целевой функции (критерия) в точке a_k , p_k — положительный скалярный коэффициент, определяющий величину шага.

Для линейно разделяемых выборок итерационный процесс сходится. В случае неразделяемых множеств сходимости может не быть. При малом числе наблюдений можно получить достаточно смещенную оценку вектора a .

В процедуре нахождения решения по методу наименьших квадратов (например, для последней целевой функции $J_s(a)$) весовой вектор находится как в случае линейно разделяемых, так и в случае линейно не разделяемых выборок.

Найдем с помощью метода наименьших квадратов вектор a , при котором выполнены равенства

$$a^T y_i = b_i, \quad i = 1, 2, \dots, n,$$

где b_i — произвольно заданные положительные константы (в общем случае заранее не известные).

Запишем систему $a^T y_i = b_i$, $i = 1, 2, \dots, n$, в матричном виде:

$$Ya = b,$$

где Y — матрица размером $n \times d$, i -й строкой которой является y_i^T ; b — вектор-столбец: $b = (b_1, b_2, \dots, b_n)^T$.

Тогда вектор a находится из условия минимума целевой функции по a :

$$J_s(a) = \|Ya - b\|^2 = \sum_{i=1}^n (a^T y_i - b_i)^2,$$

откуда $a = (Y^T Y)^{-1} Y^T b = Y^+ b$. Если матрица $Y^T Y$ размером $d \times d$ невырождена, то решение единственно. Если матрица $Y^T Y$ вырождена, то решение не единственно. Тогда решение находится из общего определения операции псевдообращения $Y^+ = (Y^T Y)^{-1} Y^T$ в виде $Y^+ = \lim_{\epsilon \rightarrow 0} (Y^T Y + \epsilon I)^{-1} Y^T$ при $\epsilon \rightarrow 0$.

Решение с наименьшей квадратичной ошибкой зависит от вектора допуска b , и различные способы выбора b приводят к различным свойствам полученного решения. Если вектор b задан произвольно, то нет гарантии того, что указанный вектор a будет разделяющим в случае разделяемых множеств.

Автоматизируем процедуру выбора вектора b , используя метод градиентного спуска. Градиент функции $J_s(a)$, вычисленный по a , записывается в виде

$$\nabla_a J_s = 2Y^T(Ya - b),$$

а градиент, вычисленный по b , задается в виде

$$\nabla_b J_s = -2(Ya - b).$$

При любом b всегда можно считать, что $a = Y^+b$, получая тем самым равенство $\nabla_a J_s = 0$ и минимизируя $J_s(a)$ по a за один шаг. Необходимо учитывать условие $b > 0$ (не допуская равенства $b = 0$). Можно положить вначале $b > 0$ и не уменьшать значения координат вектора b . Этого можно достичь при сохранении отрицательных координат градиента $\nabla_b J_s$, если вначале все положительные координаты вектора $\nabla_b J_s$ положить равными нулю. Приходим к процедуре спуска (по алгоритму Хо — Капьяпа) — минимизации функции $J_s(a, b)$ [18]:

$$b_1 > 0, \quad b_{k+1} = b_k + 2\rho e_k^T,$$

где e_k — вектор ошибки, $e_k = Ya_k - b_k$, e_k^T — положительная составляющая вектора ошибки:

$$e_k^T = \frac{1}{2}(e_k + |e_k|),$$

$|e_k|$ — вектор, координаты которого суть абсолютные величины соответствующих координат вектора e_k , $a_k = Y^+b_k$, $k = 1, 2, \dots$

Рассматриваются значения коэффициента $\rho \in (0, 1)$.

Алгоритм образует последовательность векторов допуска b и определяет разделяющий вектор для случая разделяемых множеств или явно обнаруживает неразделяемость для случая неразделяемых множеств. Вектор e_k^T должен стать нулевым, чтобы прекратилось дальнейшее изменение a_k , b_k , e_k . Если выборки линейно неразделимы, то из этого больше не следует, что при e_k^T , равном нулю, вектор e_k тоже равен нулю. Для задачи с неразделяемым множеством можно получить нулевой вектор ошибки с положительными координатами. Если это происходит, то выборки неразделимы.

Известны другие варианты этого алгоритма, когда оптимизируется выбор шага и исключаются процедуры обращения (псевдо-обращения) матрицы Y и вычисления Y^T .

Решение по методу наименьшей квадратичной ошибки в пределе при $n \rightarrow \infty$ приближается в смысле минимума средней квадратической ошибки к разделяющей функции Байеса

$$g_0(x) = P(\omega_1 | x) - P(\omega_2 | x).$$

Это следует из того, что если предположить, что выборки взяты независимо в соответствии с вероятностным законом

$$P(x) = p(x | \omega_1)P(\omega_1) + p(x | \omega_2)P(\omega_2),$$

то решение по методу наименьшей квадратичной ошибки с использованием расширенного вектора y дает разложение в ряд функции

$$g(x) = a^T y,$$

где $y = y(x)$ — функция от x .

Если определить среднюю квадратическую ошибку аппроксимации выражением

$$\epsilon^2 = \int [a^T y - g_0(x)]^2 p(x) dx,$$

то можно доказать, что величина ϵ^2 минимизируется, если вектором решения является $a = Y^+ b$.

При аппроксимации $g_0(x)$ разделяющая функция $a^T y$ позволяет получить непосредственную информацию относительно апостериорных вероятностей

$$P(\omega_1 | x) = \frac{1}{2}(1 + g_0(x)), \quad P(\omega_2 | x) = \frac{1}{2}(1 - g_0(x)).$$

Качество аппроксимации зависит от функций $y_i(x)$ и числа членов в разложении $a^T y$. Критерий средней квадратической ошибки в основном распространяется не на точки, близкие к поверхности решения $g_0(x) = 0$, а на точки, для которых $P(x) > 0,5$. Таким образом, разделяющая функция, которая наилучшим образом аппроксимирует разделяющую функции Байеса, не обязательно минимизирует вероятность ошибки.

Для построения разделяющих функций используется *потенциальная функция* $K(x, x_i)$ с текущей точкой x и выборкой x_i . Вид

потенциальной функции определяется видом выборки и существенно влияет на результаты. В общем виде можно указать, что $K(x, x_i)$ обратно пропорциональна $\|x - x_i\|^2$, например

$$K(x, x_i) = \frac{\sigma^2}{\sigma^2 + \|x - x_i\|^2}.$$

Разделяющая функция $g(x)$ получается итерационным методом, когда выборки x_1, x_2, \dots рассматриваются последовательно:

$$g_{k+1}(x) = g_k(x) + r_k(x, x_k)K(x, x_k),$$

где r_k — некоторая функция ошибки.

Использование метода потенциальных функций наиболее оправдано, когда либо невелико число выборок, либо размерность x достаточно мала, чтобы функцию $g(x)$ можно было представить в виде таблицы дискретных значений x .

Например, можно разделяющую функцию $g(x)$ определить по следующему алгоритму:

$$g_{k+1}(x) = \begin{cases} g_k(x) + K(x, x_k), & \text{если } x_k \sim \omega_1 \text{ и } g(x_k) \leq 0, \\ g_k(x) - K(x, x_k), & \text{если } x_k \sim \omega_2 \text{ и } g(x_k) \geq 0, \\ g_k(x) & \text{в остальных случаях.} \end{cases}$$

§ 9.3. Оценка разделяющего вектора с помощью методов математического программирования

Чтобы оценить вектор a для обобщенной линейной разделяющей функции $a^T y = 0$, применяют методы математического программирования. Пусть имеется множество из n выборок векторов y_1, y_2, \dots, y_n и требуется найти вектор a , удовлетворяющий неравенству $a^T y_i \geq b_i > 0$ при всех i .

Введем искусственную переменную $t \geq 0$, чтобы выполнялось условие $a^T y_i + t \geq b_i$.

Получим следующую задачу линейного программирования: минимизировать переменную t при условиях

$$a^T y_i + t \geq b_i, \quad i = 1, 2, \dots, n, \quad t \geq 0.$$

В традиционной задаче линейного программирования полагают, что $a \geq 0$. В нашем случае это условие неприемлемо, так как

вектор решения может иметь как положительные, так и отрицательные координаты.

Представим компоненты вектора a в виде комбинации двух неотрицательных компонент a_j^1 и a_0 (см. § 2.1):

$$a_j = a_j^1 - a_0, \quad j = 1, 2, \dots, d,$$

$$a_j^1 \geq 0, \quad a_0 \geq 0.$$

Получим следующую классическую задачу линейного программирования для переменных $t, a_1^1, a_2^1, \dots, a_d^1, a_0$, результатом решения которой будет решающий вектор a : минимизировать t при условиях

$$\sum_{j=1}^d y_{ij}(a_j^1 - a_0) + t \geq b_i, \quad i = 1, 2, \dots, n,$$

$$t \geq 0, \quad a_0 \geq 0, \quad a_j^1 \geq 0, \quad j = 1, 2, \dots, d.$$

Методы решения подобных задач хорошо изучены и рассмотрены в гл. 7 (см. также [18, 61]).

Можно сформулировать и другую задачу линейного программирования: сведем задачу минимизации персептронной функции критерия к задаче линейного программирования. Напомним, что минимизация этой функции дает разделяющий вектор в случае разделяемых множеств и приводит к решению в том случае, когда выборки неразделимы.

Персептронная функция критерия задается в виде

$$J_p(a) = \sum_{y \in Y} (-a^T y),$$

где Y — множество выборок, классифицируемых с ошибкой посредством вектора a .

Чтобы исключить тривиальное решение $a = 0$, введем положительный вектор допуска b и запишем функцию критерия в виде

$$J'_p(a) = \sum_{y_i \in Y'} (b_i - a^T y_i),$$

где Y' — множество выборок y_i , удовлетворяющих условию $a^T y_i \leq b_i$.

Функция $J'_p(a)$ является кусочно-линейной, поэтому следует ввести n искусственных переменных и соответствующих им огра-

ний и построить некоторую эквивалентную линейную целевую функцию. Рассмотрим задачу нахождения векторов a и t , минимизирующих линейную функцию

$$Z = \sum_{i=1}^n t_i$$

при ограничениях $t_i \geq 0$, $t_i \geq b_i - a^T y_i$, $i = 1, 2, \dots, n$.

С учетом неотрицательности переменных получим следующую задачу линейного программирования для нахождения вектора a :

минимизировать функцию $Z = \sum_{i=1}^n t_i$ при условиях

$$\sum_{j=1}^{\hat{d}} y_{ij}(a_j^1 - a_0) + t_i \geq b_i, \quad t_i \geq 0, \quad a_0 \geq 0, \quad a_j^1 \geq 0, \quad j = 1, 2, \dots, \hat{d}.$$

Выбор $a = 0$, $t_i = b_i$, $i = 1, 2, \dots, n$, обеспечивает получение допустимого базисного решения, с которого начинается традиционный алгоритм симплекс-метода. Сходимость методов линейного программирования гарантирована как в случае линейной разделяемости, так и в случае, когда исходные выборки неразделяемы.

Использование процедуры минимизации квадратичной ошибки позволяло находить разделяющий вектор в случае разделяемых множеств и обнаруживать неразделяемость в случае неразделяемых множеств. Поэтому естественно для отыскания решающего вектора a применить следующую задачу квадратичного программирования: минимизировать функцию

$$J_s = \sum_{i=1}^n (a^T y_i - b_i)^2$$

при условиях

$$\sum_{j=1}^{\hat{d}} y_{ij}(a_j^1 - a_0) \geq b_i, \quad i = 1, 2, \dots, n,$$

$$a_0 \geq 0, \quad a_j^1 \geq 0, \quad j = 1, 2, \dots, \hat{d}.$$

Анализ различных алгоритмов для нахождения линейных разделяющих функций показывает, что выбор подходящего алгоритма определяется такими факторами, как простота программирования, количество и размерность выборок. Если линейная разделяющая функция обеспечивает незначительный процент ошибок,

любая из этих процедур при корректном ее применении позволит получить приемлемое решение.

§ 9.4. Разделяющие функции для случая многих классов

Процедуры получения разделяющих функций для случая двух классов (образов) стремятся распространить на случай многих классов. Универсального метода такого распространения не существует, но при использовании различных приемов решения разных задачах это удастся сделать. Так, в случае линейной машины для c классов вместо множества разделяющих функций $g_i(x) = a_i^T y$, $i = 1, 2, \dots, c$, вводится разделяющая функция вида $\alpha^T \eta_{ij}$, где

$$\begin{aligned}\alpha &= (a_1, a_2, \dots, a_c)^T, \\ \eta_{12} &= (y, -y, 0, \dots, 0)^T, \\ \eta_{13} &= (y, 0, -y, \dots, 0)^T, \\ &\dots\dots\dots, \\ \eta_{1c} &= (y, 0, 0, \dots, -y)^T, \\ &\dots\dots\dots,\end{aligned}$$

т. е. в строке η_{ij} на месте i -го элемента находится y , на месте j -го элемента находится $-y$, остальные элементы равны нулю.

Если $\alpha^T \eta_{ij} > 0$ для всех $i \neq j$, то линейная машина, соответствующая компонентам вектора α , будет правильно классифицировать y .

В этой процедуре размерность исходных данных увеличивается в c раз, число выборок — в $c-1$ раз, что делает трудоемким применение данного алгоритма. Однако такое сведение c -мерной задачи к двумерной полезно для доказательства сходимости.

При распространении метода наименьших квадратов на случай многих классов наиболее часто используют прием, при котором исходная задача сводится к рассмотрению множества c задач для двух классов. Если сразу решать задачу минимизации квадрата ошибки в случае многих классов аналогично случаю двух классов, то вводят матрицу Y размером $n \times \hat{d}$, которую разбивают на блоки следующим образом:

$$Y = (Y_1, Y_2, \dots, Y_c)^T,$$

где выборки, помеченные символом ω_i , включают в себя строки Y_i . Затем вводят матрицу весовых векторов A размером $\hat{d} \times c$:

$$A = (a_1, a_2, \dots, a_c)^T,$$

и матрицу B размером $n \times c$:

$$B = (B_1, B_2, \dots, B_c)^T,$$

где все элементы B_i являются нулевыми, за исключением элементов i -го столбца, равных 1. Тогда след квадратичной матрицы ошибок $(YA - B)^T(YA - B)$ будет минимален, если $A = Y^+ B$, где Y^+ — псевдообращение матрицы Y .

Полученный результат может быть обобщен. Пусть λ_{ij} — потери для случая, когда принимается решение о выборе ω_i вместо истинного ω_j , и предположим, что j -я подматрица матрицы B задается выражением

$$B_j = - \begin{pmatrix} \lambda_{1j} \lambda_{2j} \dots \lambda_{cj} \\ \lambda_{1j} \lambda_{2j} \dots \lambda_{cj} \\ \dots \dots \dots \\ \lambda_{1j} \lambda_{2j} \dots \lambda_{cj} \end{pmatrix}, \quad j = 1, 2, \dots, c,$$

где число строк равно n_j , $j = 1, 2, \dots, c$.

Тогда если число выборок стремится к бесконечности, то решение $A = Y^+ B$ дает разделяющие функции $a_i^T y$, которые обеспечивают оптимальное (в смысле минимума средней квадратической ошибки) приближение байесовской разделяющей функции

$$g_{0i}(x) = - \sum_{j=1}^c \lambda_{ij} P(\omega_j | x).$$

§ 9.5. Учет погрешностей наблюдений при оценке значений параметров классификаторов

В § 9.1–9.4 рассмотрены основные традиционные методы оценки значений параметров классификаторов по выборке. В основном они сводятся к получению разделяющих функций между классами (образами), в частности к линейным разделяющим функ-

циям или к обобщенным линейным разделяющим функциям. Однако все рассмотренные алгоритмы не учитывают погрешности измерений наблюдаемых признаков, на основе которых принимаются решения, что при очень богатой статистике может быть как-то компенсировано. При небольшом числе наблюдений мы получим смещенные оценки разделяющих (решающих) функций, интервальные оценки разделяющих (решающих) функций будут значительными и зоны неопределенности в принятии решения будут большими. Таким образом, для принятия надежного решения необходимо учитывать погрешности измерений наблюдаемых признаков, что позволит правильно определить интервальные оценки разделяющих функций, а тем самым и зоны неопределенности принимаемых решений.

Зона неопределенности классификатора по минимуму расстояния

Рассмотрим квадрат евклидова расстояния между измеренным вектором признаков x и эталонным вектором признаков μ одного из объектов (образов):

$$r^2 = (x - \mu)^T K^{-1} (x - \mu),$$

где K — ковариационная (дисперсионная) матрица наблюдений признаков x и μ .

Например, на плоскости \mathbb{R}^2 в предположении, что

$$D(x - \mu) = D(x) + D(\mu),$$

$$K = \begin{pmatrix} D(x_1) + D(\mu_1) & \dots & 0 \\ 0 & \dots & D(x_2) + D(\mu_2) \end{pmatrix},$$

квадрат расстояния r вычисляется по формуле

$$r^2 = \frac{(x_1 - \mu_1)^2}{D(x_1) + D(\mu_1)} + \frac{(x_2 - \mu_2)^2}{D(x_2) + D(\mu_2)}. \quad (9.1)$$

Все величины, входящие в выражение (9.1), получены в результате эксперимента и имеют погрешности.

Введем общий вектор θ размерности n , координатами которого являются все входящие в формулу (9.1) величины; дисперсию ошибки i -го параметра обозначим $D(\theta_i)$. Тогда за счет погрешностей наблюдаемых величин дисперсия ошибки в определении r^2 может быть вычислена по формуле

$$D(r^2) = \sum_{i=1}^n \left(\frac{\partial r^2}{\partial \theta_i} \right)^2 D(\theta_i) + 2 \sum_{i=1}^n \sum_{\substack{j=2 \\ j>i}}^n \frac{\partial r^2}{\partial \theta_i} \frac{\partial r^2}{\partial \theta_j} D(\theta_i, \theta_j), \quad (9.2)$$

где $D(\theta_i, \theta_j)$ — корреляционный момент оценок координат вектора θ . В предположении некоррелированности оценок θ формула упрощается:

$$D(r^2) = \sum_{i=1}^n \left(\frac{\partial r^2}{\partial \theta_i} \right)^2 D(\theta_i). \quad (9.3)$$

Для приведенного примера (формула (9.1)) имеем

$$\begin{aligned} D(r^2) = & \sum_{i=1}^2 \left(\frac{\partial r^2}{\partial x_i} \right)^2 D(x_i) + \sum_{i=1}^2 \left(\frac{\partial r^2}{\partial \mu_i} \right)^2 D(\mu_i) + \\ & + \sum_{i=1}^2 \left(\frac{\partial r^2}{\partial (D(x_i))} \right)^2 D(D(x_i)) + \sum_{i=1}^2 \left(\frac{\partial r^2}{\partial (D(\mu_i))} \right)^2 D(D(\mu_i)). \end{aligned}$$

Дисперсия $D(r^2)$ определяет величину неопределенности $\sqrt{D(r^2)}$ квадрата расстояния между измеренным вектором признаков и эталонным, учет которой обязателен в процессе принятия решения.

Если вводятся другие определения расстояния между векторами измеренных признаков и эталонов, то величина неопределенности вычисляется аналогичным образом по формулам (9.2) и (9.3).

Зона неопределенности линейных разделяющих функций

Будем считать, что найдено уравнение линейной разделяющей функции, т. е. определен решающий вектор \hat{a} , или, другими словами, оценки свободных параметров $\hat{a}_1, \hat{a}_2, \dots, \hat{a}_m$ этой линейной функции.

В процессе определения оценок \hat{a}_l , $l = 1, 2, \dots, m$, должны быть найдены как минимум дисперсии этих оценок. Они могут быть получены:

- а) аналитическим путем (этот случай будет рассмотрен ниже);
- б) методом статистических испытаний.

В случае б), зная погрешности наблюдаемых значений признаков эталонов, с помощью датчиков случайных чисел проведем необходимую серию статистических испытаний. Каждое наблюдение

даст серию точек, для которых получают оценки параметров \hat{a}_l , $l = 1, 2, \dots, m$. Для l -го параметра будет получен вариационный ряд, по которому можно определить наиболее вероятное значение \hat{a}_l и дисперсию этого значения $D(\hat{a}_l)$. При необходимости может быть проведен и более строгий анализ вариационного ряда.

Пусть \hat{a} — m -мерный вектор, координатами которого будут оценки \hat{a}_l , $l = 1, 2, \dots, m$, $D(\hat{a})$ — ковариационная матрица оценок \hat{a}_l , $l = 1, 2, \dots, m$. Тогда дисперсия $D(\hat{y})$ оценки разделяющей линейной функции $y(a)$ для любого значения y находится по формуле

$$D(\hat{y}) = y^T D(\hat{a}) y,$$

где y — вектор с координатами y_1, y_2, \dots, y_m . Эта формула является частным случаем формулы (9.2). Если матрица $D(\hat{a})$ диагональная, то получим аналог формулы (9.3).

Пример. По результатам наблюдений найти интервальную оценку обобщенной линейной разделяющей функции

$$y(a) = a_1 y_1 + a_2 y_2 + a_3 y_3.$$

Решение. Положим, что в процессе наблюдений определены оценки \hat{a}_l и их дисперсии $D(\hat{a}_l)$, $l = 1, 2, 3$. Считаем, что корреляцией оценок \hat{a}_l , $l = 1, 2, 3$, можно пренебречь, т. е. матрица $D(\hat{a})$ имеет вид

$$D(\hat{a}) = \begin{pmatrix} D(\hat{a}_1) & 0 & 0 \\ 0 & D(\hat{a}_2) & 0 \\ 0 & 0 & D(\hat{a}_3) \end{pmatrix}.$$

Дисперсию $D(\hat{y})$ оценок значений разделяющей функции $y(a)$ находим по формуле

$$\begin{aligned} D(\hat{y}) &= (y_1, y_2, y_3) \begin{pmatrix} D(\hat{a}_1) & 0 & 0 \\ 0 & D(\hat{a}_2) & 0 \\ 0 & 0 & D(\hat{a}_3) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \\ &= y_1^2 D(\hat{a}_1) + y_2^2 D(\hat{a}_2) + y_3^2 D(\hat{a}_3). \end{aligned}$$

Зная значения $y(a)$ и $D(\hat{y})$, для каждого y построим доверительные интервалы значений $\hat{y} = y(\hat{a})$ разделяющей функции. Огибающая линия, проведенная по этим точкам, даст интервальную оценку (зону неопределенности в принятии решений) данной

разделяющей функции. Пусть доверительная вероятность равна β ($\beta = 0,95; 0,997; \dots$). Интервальная оценка для каждой точки y разделяющей функции определяется условием

$$P\{\hat{y} - t_\beta \sqrt{D(\hat{y})} < y(a) < \hat{y} + t_\beta \sqrt{D(\hat{y})}\} = \beta,$$

где t_β — квантиль распределения Стьюдента — определяется выбранным значением β и числом степеней свободы $\nu = n - m$ (n — число наблюдаемых точек, m — число оцениваемых параметров). Здесь $\hat{y} = y(\hat{a})$ — значения $y(a)$ при найденных оценках \hat{a} . Задав значение β , выбираем значение t_β и определяем зону неопределенности принятия решений с помощью предлагаемой разделяющей функции. При малом числе эталонных наблюдений эта зона будет достаточно большой.

В ряде случаев область неопределенности может быть получена аналитическим путем. Разделяющий вектор a может быть определен по методу наименьших квадратов, когда в качестве критерия рассматривается функция

$$J_s(a, b) = \|Ya - b\|^2,$$

где Y — матрица измеренных значений признаков, b — вектор допуска.

Метод наименьших квадратов позволяет получить не только оценки \hat{a} , но и дисперсию этих оценок $D(\hat{a})$ (см. § 7.5). При ковариационной матрице измерений $D(y)$ функцию $J_s(a, b)$ можно записать в виде

$$J_s(a, b) = (Ya - b)^T D^{-1}(y)(Ya - b).$$

Тогда дисперсия оценок \hat{a} будет иметь вид

$$D(\hat{a}) = [Y^T D^{-1}(y)Y]^{-1}.$$

Дальнейшие действия при найденных оценках \hat{a} и $D(\hat{a})$ аналогичны ранее описанным: находятся оценки разделяющей функции $y(\hat{a})$ и дисперсии значений $D(\hat{y})$ этой функции, затем определяются интервальные оценки разделяющей функции, а тем самым и зона неопределенности принимаемых решений.

§ 9.6. Распознавание образов по измеренному вектору признаков

Пусть известны средние квадратические отклонения для погрешности измеренных векторов признаков x как в процессе обучения системы распознавания, так и в процессе идентификации образов. Закон распределения погрешностей наблюдений неизвестен, число точек наблюдения ограничено. Будем искать обобщенную линейную разделяющую функцию.

Рассмотрим обобщенную линейную разделяющую функцию

$$g(x) = a^T y,$$

где a — весовой вектор, y — вектор, координатами которого являются константа 1 и функции от x , определяющие вид разделяющих функций (линейные, полиномиальные и т. д.). Задача сводится к определению *весового вектора (вектора решения)* a . Для однозначного выбора вектора решения следует ввести некую целевую функцию (или критерий). Чтобы включить в рассмотрение погрешности наблюдений x (а они трансформируются по формулам переноса ошибок в погрешности для y), используем метод наименьшей квадратичной ошибки. С учетом погрешностей наблюдений признаков, как подробно показано в гл. 7, целевая функция, минимум которой определяет вектор решений a , имеет вид

$$J(a) = \sum_{i=1}^n [(a^T \xi_i - b_i)^2 + (\xi_i - y_i)^2], \quad (9.4)$$

где ξ_i — истинное значение y_i , т. е. $y_i = \xi_i + \delta_i$; b_i — вектор допуска, который также должен быть определен. Функция критерия $J(a)$ использует все выборки.

Функционал (9.4) получен как функционал ортогональной регрессии в предположении, что средние квадратические отношения признаков равны единице. Необходимо его дополнить условием для нахождения неизвестного вектора ξ :

$$\left. \frac{\partial J(a)}{\partial \xi} \right|_{\xi=\hat{\xi}} = 0. \quad (9.5)$$

Преимуществом выбранного критерия является тот факт, что решение, следующее из этого критерия, приближается в пределе к

разделяющей функции Байеса $g_0(x) = P(\omega_1 | x) - P(\omega_2 | x)$, хотя может не обеспечивать минимальную ошибку решения.

Рассмотрим два подхода:

- 1) вектор допуска b задается;
- 2) вектор допуска b оценивается в алгоритме построения разделяющей функции.

В первом случае все координаты вектора b равны -1 , если наблюдения относятся к образу ω_1 , и равны $+1$, если наблюдения относятся к образу ω_2 . Тогда

$$J(a) = \sum_{y \in Y_1} [(a^T \xi - 1)^2 + (\xi - y)^2] + \sum_{y \in Y_2} [(a^T \xi + 1)^2 + (\xi - y)^2],$$

где Y_1 и Y_2 — множества наблюдений для ω_1 и ω_2 соответственно.

Объединив условия (9.4) и (9.5), для обобщенной линейной разделяющей функции получим функционал (9.4) в более приемлемом для практического применения виде:

$$J(a) = \sum_{j=1}^{n_1} \frac{\left(\sum_{i=1}^k a_i y_{ij} - 1 \right)^2}{1 + \sum_{i=1}^k a_i^2 \sigma^2(y_i)} + \sum_{j=n_1+1}^{n_2} \frac{\left(\sum_{i=1}^k a_i y_{ij} + 1 \right)^2}{1 + \sum_{i=1}^k a_i^2 \sigma^2(y_i)}, \quad (9.6)$$

где $\sigma^2(y_i) \equiv \sigma^2(x_i)$ — средняя квадратическая ошибка i -го вектора наблюдений, одинаковая во всех наблюдениях; $j = 1, 2, \dots, n_1$ — номер наблюдений для ω_1 , $j = n_1 + 1, \dots, n_2$ — номер наблюдений для ω_2 .

Точка минимума функционала (9.6) определяет точечную оценку вектора решения a , а следовательно, и точечную оценку разделяющей функции. Ковариационная матрица оценки вектора решения a определяется из условия

$$D(\hat{a}) = N^{-1}, \quad N = \left\| -\frac{\partial^2 J(a)}{\partial a_l \partial a_m} \right\|_{a=\hat{a}}, \quad l, m = 1, 2, \dots, k,$$

а дисперсия значений разделяющей функции в точке наблюдения y имеет вид

$$D(g(\hat{y})) = y^T D(\hat{a}) y,$$

где $y = (y_1, y_2, \dots, y_k)^T$.

При неизвестных векторах допуска b , как следует из результатов гл. 7 для линейной функции, функционал (9.6) приобретает вид

$$J(a) = \sum_{j=1}^n \frac{\left(\sum_{i=1}^k a_i y_{ij} - b_j \right)^2}{1 + \sum_{i=1}^k a_i^2 \sigma^2(y_i)},$$

где принято, что координаты вектора b — детерминированные величины. Для каждой оценки вектора a с помощью градиентного спуска образуется последовательность векторов допуска b :

$$b_1 > 0, \quad b_{v+1} = b_v + 2pe_v^+,$$

где

$$e_v^+ = \frac{1}{2}(e_v + |e_v|), \quad e_v = a_v^T y - b_v, \quad 0 < p < 2.$$

В процедуре идентификации образов по вектору наблюдений x_n строится вектор u_n , который присоединяется к вектору ранее проведенных наблюдений, и с помощью функционала (9.6) или условия (9.5) для каждой j -й гипотезы определяются оценки $\hat{\xi}_{jn}$, по которым принимается решение.

Результат определения линейной разделяющей функции и идентификации летательных аппаратов по наблюдаемому вектору

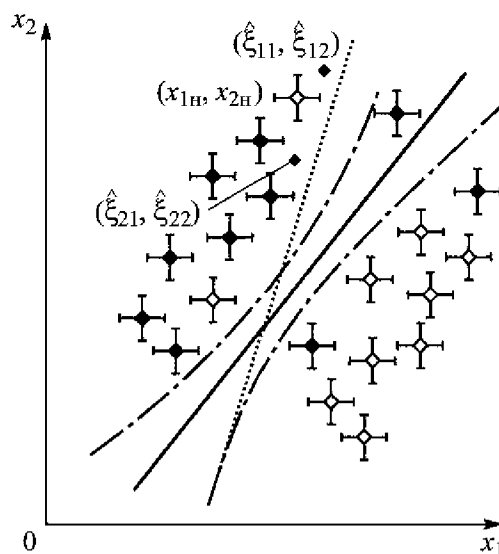


Рис. 9.2. Распознавание образов с учетом погрешности наблюдений признаков с помощью линейной разделяющей функции

$(x_{1н}, x_{2н})$ приведен на рис. 9.2. Различие решающих функций и отличие оценок «истинных значений» наблюдаемого вектора $(\hat{\xi}_{j1}, \hat{\xi}_{j2})$, $j = 1, 2$, от вектора $(x_{1н}, x_{2н})$ очевидны.

На рис. 9.2 указаны значения признаков и погрешности их измерения для двух объектов. Светлые точки относятся к одному объекту, темные — к другому. Представлены линейные разделяющие функции как с учетом погрешности измерения (сплошная линия), так и без учета погрешности (линия из точек). Для первой разделяющей функции определены интервальные оценки (штрихпунктирные линии). По наблюдаемым значениям $(x_{1н}, x_{2н})$ определены оценки «истинных» значений координат точки наблюдения ($\hat{\xi}_1$ и $\hat{\xi}_2$). Точку наблюдения присоединяем последовательно к предшествующим выборкам для каждого объекта.

Получены соответственно значения $(\hat{\xi}_{11}, \hat{\xi}_{12})$ и $(\hat{\xi}_{21}, \hat{\xi}_{22})$, по которым следует принимать решение.

§ 9.7. Алгоритм идентификации объектов с учетом погрешности признаков

Рассмотрим задачу распознавания (идентификации) объектов, когда закон распределения признаков не известен, но известны значения признаков и, возможно, средние квадратичные отклонения измеренных значений признаков.

Алгоритм построения разделяющей функции может опираться на всю совокупность обучающей выборки или использовать только часть точек и признаков. Рассмотрим обобщенные линейные разделяющие функции, весовой вектор в которых оценивается с помощью операции минимизации квадратичной невязки. При этом используем методы конфлюэнтного анализа, в которых учитываются погрешности наблюдений признаков при минимизации функционала квадратичной невязки при фиксированном векторе допуска (пороге).

Для минимизации функционалов применяется одна из разновидностей метода покоординатного спуска с контролем точности результатов и с переменным шагом. Динамическое размещение массивов переменных в памяти компьютера позволяет рассматривать задачи произвольной размерности. Для решения систем линейных алгебраических уравнений и операций с матрицами применяются соответствующие стандартные процедуры.

На рис. 9.3 представлена общая блок-схема компьютерной программы, реализующей данный алгоритм. Программа написана на языке Fortran-90 в среде Microsoft Developer Studio с привлечением некоторых стандартных вычислительных процедур, а также диалоговых и графических возможностей Windows 95, 98.

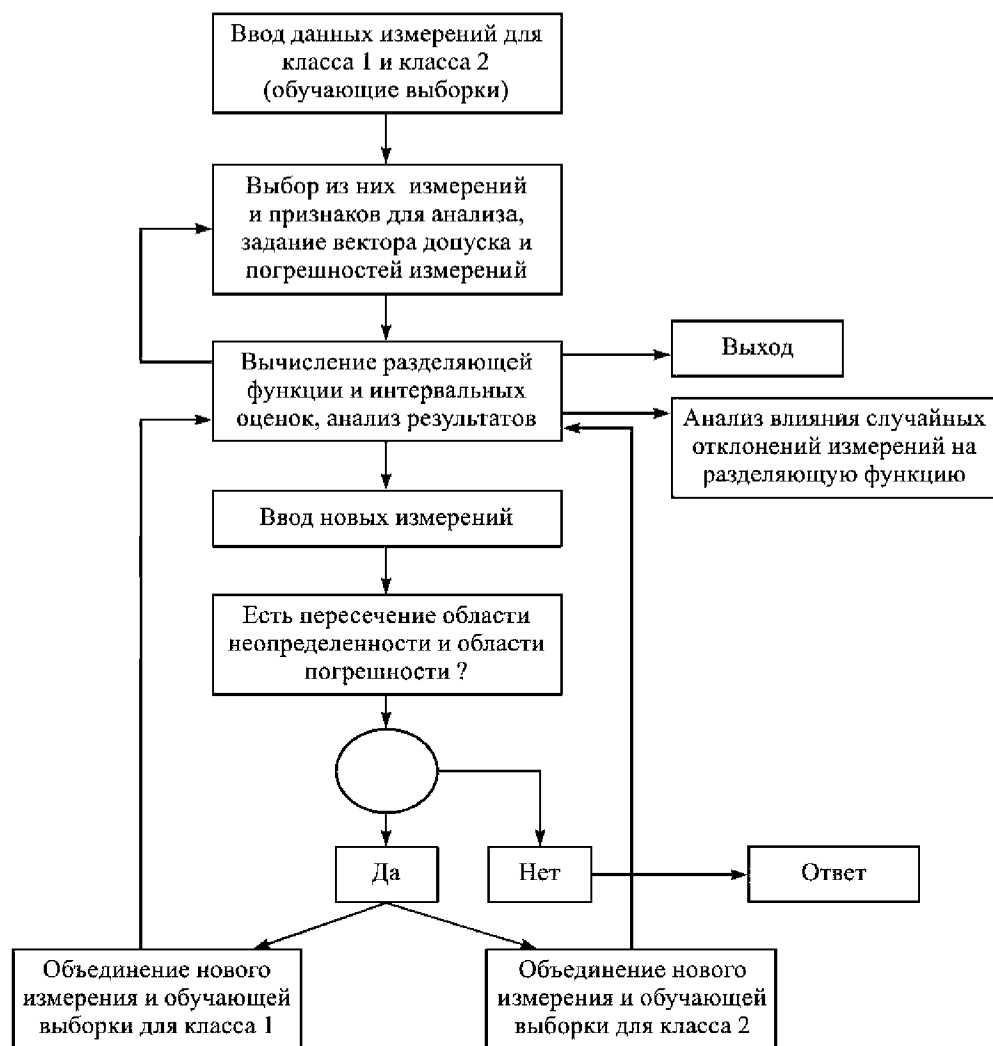


Рис. 9.3. Общая блок-схема программы

Компьютерная программа используется для определения и графического представления поверхности, разделяющей экспериментальные точки двух классов, а также идентификации на этой основе неизвестных точек. Программа применима для произвольного числа точек и признаков. В ней учитываются погрешности измерений. Программа работает в диалоговом режиме в операционной среде Windows 95, 98 и более поздних версий.

При работе программы используются следующие файлы:

- 1) выполняемый Split.exe;
- 2) содержащий числовые данные для точек класса 1 (например, File1);
- 3) содержащий числовые данные для точек класса 2 (например, File2);
- 4) содержащий числовые данные для точек, подлежащих идентификации («новых точек») (например, FileN).

Размещение и имена файлов с данными — произвольные, но структуры данных в этих файлах одинаковы и строго определены. Все данные находятся в текстовом виде и могут редактироваться в произвольном редакторе.

Первая строка — произвольный текст, описывающий содержание файла.

Вторая строка — два целых числа, разделенных пробелами. Первое число — количество признаков, второе — количество точек.

Третья и последующие строки содержат собственно данные. Числа в строках записаны в произвольном формате, но между ними должно быть не менее одного пробела. Номер столбца определяет номер признака, а номер строки — номер точки.

После запуска программы (файл Split.exe) запрашивается имя файла с данными класса 1 (File1) и с данными класса 2 (File2).

Затем появляется основное диалоговое окно, где с помощью стандартных средств Windows необходимо выбрать, какие точки (Class1 и Class2) и какие признаки (Features) будут использованы для вычисления и построения разделяющей поверхности. Здесь же можно установить значения вектора допуска (b_1, b_2) и стандартных отклонений (Dispersion) наблюдаемых точек в зависимости от признака, а также задать тип разделяющей функции: линейная (linear) или квадратичная (quadratic).

Кнопка «ОК» запускает процедуру построения разделяющей поверхности. Результаты отображаются в нескольких окнах на экране.

1. Окно «Graphic: Deviation from discrimination surface» содержит графическое представление отклонений наблюдаемых точек от разделяющей поверхности. Цветные прямоугольные контуры дают соответствующие интервальные оценки разделяющей функции.

2. Окно «Table: Deviation from discrimination surface» содержит в основном те же данные в виде таблицы.

3. Окно «Parameters» содержит некоторые характеристики, описывающие процедуру расчета разделяющей функции.

4. Окно «Discrimination Line-2D» открывается, когда число признаков равно двум. В этом окне представлены распределение выбранных наблюдаемых точек на плоскости (оси эллипсов соответствуют величинам стандартных отклонений), разделяющая линия (черным цветом), ее интервальные оценки (синим цветом). Для просмотра окно следует выделить и расширить.

После анализа результатов необходимо нажать кнопку на экране «Push to continue» (левый верхний угол экрана) для продолжения работы программы.

При этом открывается диалоговое окно «Repeat calculations or Input new points», в котором предлагается сделать выбор:

1) вернуться к предыдущему этапу работы программы (Repeat points selection);

2) перейти к вводу новой точки для ее идентификации (Input new points);

3) перейти к анализу влияния небольших случайных отклонений наблюдаемых точек на положение разделяющей линии (Points random deviation).

Вариант 3) возможен только для плоского случая. При выборе этого продолжения и дальнейшем многократном нажатии кнопки на экране «Push to continue» происходят случайные вариации положения наблюдаемых точек и соответствующий пересчет разделяющей функции. Результаты отображаются в окне «Discrimination Line-2D».

Выбор варианта 2) приводит к этапу ввода и анализа новой точки. При этом открывается диалоговое окно, где необходимо указать имя файла с новыми данными (FileN). После этого в другом диалоговом окне надо отметить одну точку для исследования. Затем в режиме диалога предлагается выбрать одно из продолжений:

а) отобразить положение новой точки и интервальные оценки на графике в окне «Graphic: Deviation from discrimination surface»;

б) добавить новую точку к точкам класса 1 и пересчитать положение разделяющей поверхности; результаты отображаются так же, как описано выше при вычислении разделяющей функции;

в) добавить новую точку к точкам класса 2 и пересчитать положение разделяющей поверхности; результаты отображаются, как в п. б).

После анализа результатов можно нажать кнопку «Push to continue» для продолжения работы с другой новой точкой или закрыть приложение (программу) обычным образом. Содержимое окон можно копировать и выдавать на печать.

Программное обеспечение включает в себя выполняемый файл Split.exe и файлы с данными. Программа выполняется в диалоговом режиме в операционной среде Windows 95, 98.

Выполнение программы прерывается, если:

- 1) файлы данных не имеют предписанной структуры;
- 2) в соответствующих диалогах не выбрано ни одной точки и (или) ни одного признака.

Об этом выдается соответствующее предупреждение. В этих случаях необходимо запустить программу снова. Предупреждения выдаются также при плохой обусловленности используемых в процессе счета матриц.

Для исследования решения задачи построения разделяющей функции проводилась серия тестовых расчетов. Минимум функционала находился по вектору a . Диапазоны изменения числа k параметров в различных задачах разные: от $k = 2$ до нескольких десятков. Характерный вид функционала для $k = 2$ представлен на рис. 9.4–9.6, откуда видно, что для успешного нахождения минимума функционала необходимо удачно выбрать нулевое приближение в итерационной процедуре. Это осуществляется, если воспользоваться традиционным подходом в процедуре идентификации, т. е. считать, что $\sigma^2(y_i) = 0$. В этом случае для определения вектора оценок \hat{a} , являющегося точкой минимума функционала $J(a)$, получается система линейных алгебраических уравнений. Далее одним из итерационных методов поиска экстремума опреде-

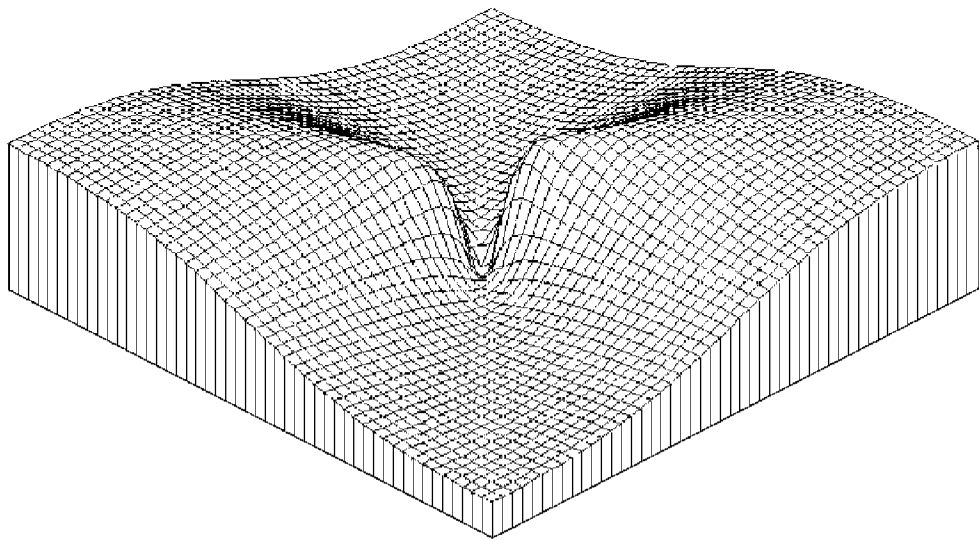


Рис. 9.4. Поверхность минимизируемого функционала

ляются вектор решений a и разделяющая функция. Дисперсионная матрица аналитически определяется из соответствующих формул, а по ней — зона неопределенности. Вектор допуска b выбирается из условия наилучшего разделения данных наблюдений.

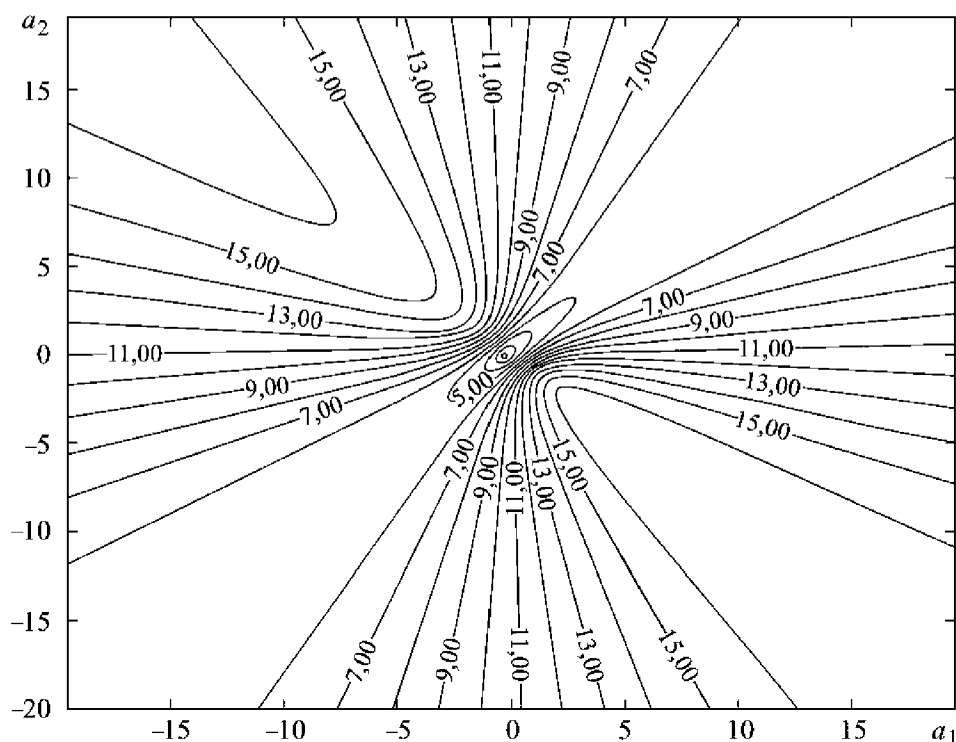


Рис. 9.5. Линии уровня функционала

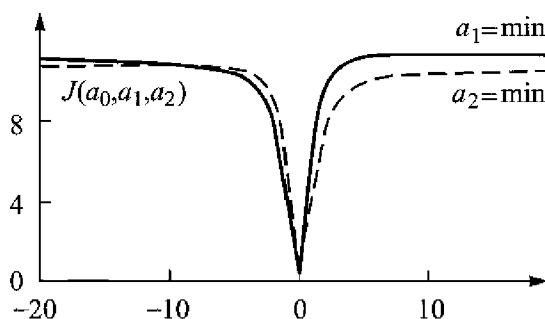


Рис. 9.6. Сечения функционала

На рис. 9.7–9.11 представлены результаты тестовых расчетов при $k = 2$ для различных исходных данных и параметров задачи. На всех этих рисунках используются одинаковые обозначения. Прямоугольники определяют местоположение идентифицируемых точек двух классов, причем нижний всегда соответствует точкам

класса 1. Точки внутри прямоугольников распределены равномерно. Числа внутри прямоугольников дают количество точек соответствующего класса. Тонкая прямая линия определяет разделяющую линию, полученную без учета погрешностей признаков (традиционное решение — нулевое приближение). Жирная прямая определяет разделяющую линию, полученную предлагаемым методом. Пунктирные линии дают соответствующие интервальные оценки.

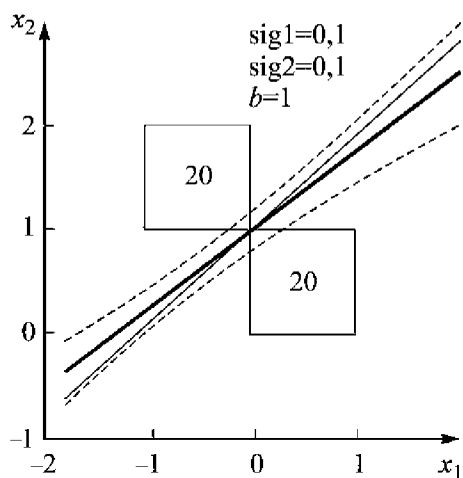


Рис. 9.7. Положение разделяющих линий при $b = 1$ и $\sigma(x_1) = \sigma(x_2) = 0,1$

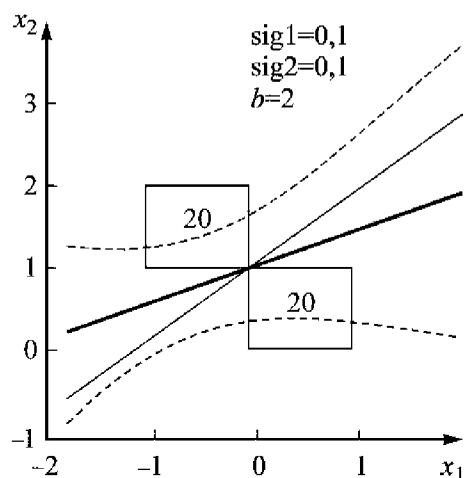


Рис. 9.8. Положение разделяющих линий при $b = 2$ и $\sigma(x_1) = \sigma(x_2) = 0,1$

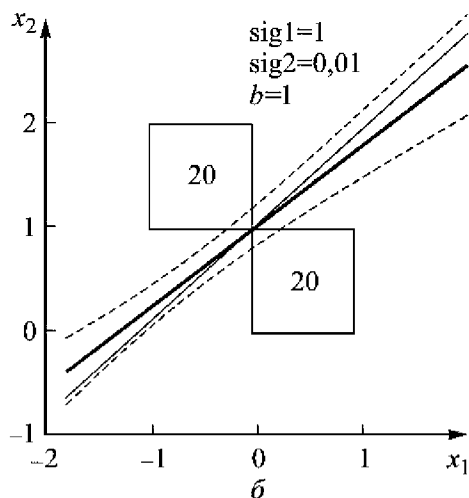
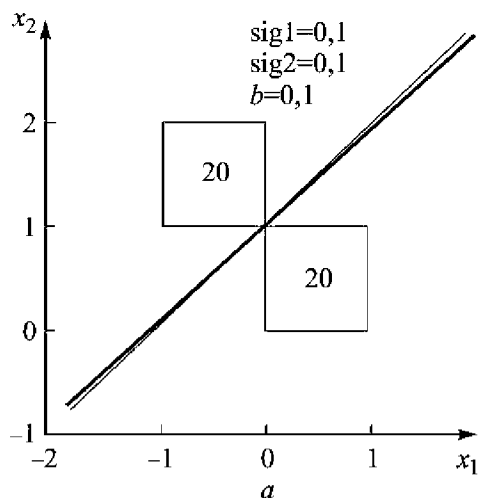


Рис. 9.9. Положение разделяющих линий:

a — при $b = 0,1$, $\sigma(x_1) = \sigma(x_2) = 0,1$; $б$ — при $b = 1$, $\sigma(x_1) = 1$, $\sigma(x_2) = 0,01$

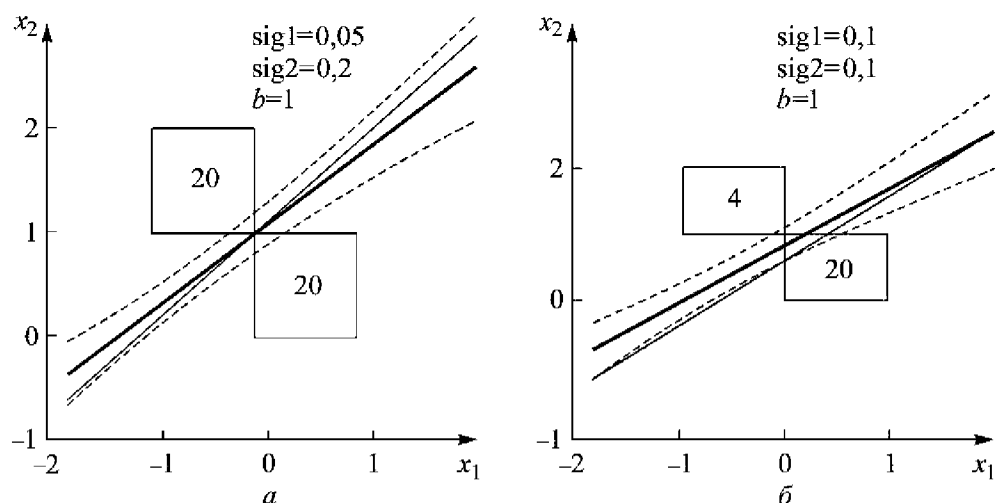


Рис. 9.10. Положение разделяющих линий:

a — при $b=1$, $\sigma(x_1)=0,05$, $\sigma(x_2)=0,2$; $б$ — при $b=1$, $\sigma(x_1)=\sigma(x_2)=0,01$

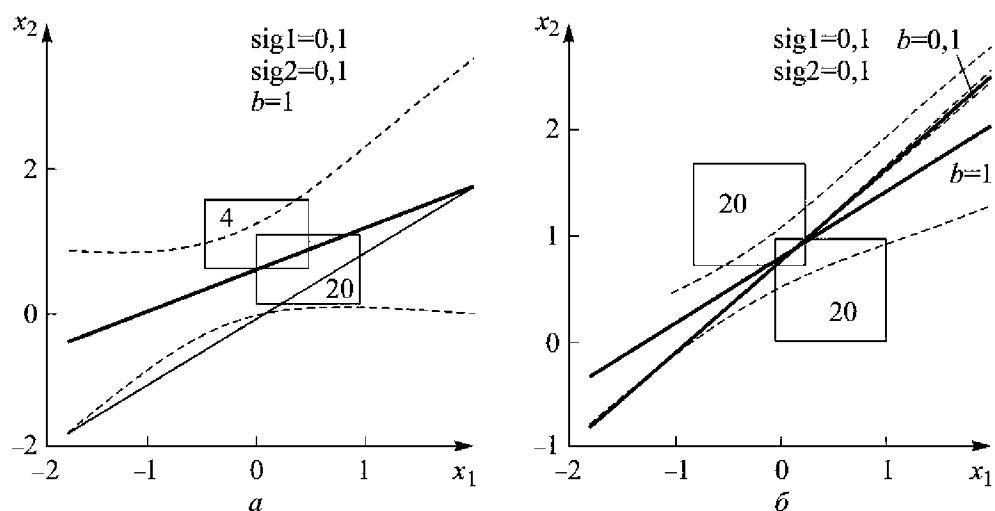


Рис. 9.11. Положение разделяющих линий:

a — при $b=1$, $\sigma(x_1)=\sigma(x_2)=0,1$; $б$ — при $b=0,1$ и $b=1$, $\sigma(x_1)=\sigma(x_2)=0,1$

На рис. 9.7–9.10 представлена ситуация, когда точки двух классов разделяются, а на рис. 9.11 — когда соответствующие области локализации идентифицируемых точек пересекаются.

Исследовалась зависимость решения от вектора допуска b , когда остальные параметры остаются неизменными. С увеличением b от 0,1 до 2 решение отличается от традиционного: увеличиваются интервальные оценки, а также растет количество итераций при поиске минимума функционала.

Зависимость разделяющей линии от величины дисперсии (sig1 и sig2), которая указана на рисунках, слабая, несмотря на значительный диапазон изменения погрешностей измерений. При разном числе точек в разных классах (в данном случае в классе 1 имеется 20 точек, а в классе 2 — 4 точки) происходит смещение разделяющей линии в сторону локализации класса 1.

Для неразделяющихся признаков сохраняются закономерности, описанные для разделяющихся классов. Таким образом, тестовые расчеты дают представление об особенностях применения рассматриваемого метода.

§ 9.8. Идентификация землетрясений и искусственных взрывов по сейсмическим проявлениям

Разработанный метод был применен для идентификации слабых локальных землетрясений и искусственных взрывов, записанных сейсмосетью. Все исходные данные взяты из работ [97, 98]. Оцифровка сейсмограмм основана на относительных энергетических спектральных распределениях фаз P и S , извлеченных из сейсмограмм и автоматически обработанных. Энергетический спектр сейсмических шумов был исключен из данных для улучшения идентификации.

Первоначальная обработка сейсмограмм производилась следующим образом: фазы P и S отфильтровывались в следующих частотных интервалах:

$$D_0 = 1...15 \text{ Гц}, \quad D_1 = 1...3 \text{ Гц}, \quad D_2 = 3...6 \text{ Гц},$$

$$D_3 = 6...10 \text{ Гц}, \quad D_4 = 10...15 \text{ Гц}.$$

Затем вычислялись отношения средних интенсивностей по частотным интервалам к суммарной интенсивности и, аналогично, отношения амплитуд. Таким образом формировалось 18 различных признаков (числовых характеристик сейсмограмм) для процедуры распознавания. Все они подвергались некоторой нормировке.

Отбор признаков основывался на минимизации вероятности ошибки классификации, что позволило автоматически извлечь пять наиболее информативных признаков из 18 исходных:

1) отношение средних энергий S -фазы и P -фазы в частотном диапазоне 10...15 Гц;

2) доля средней энергии S -фазы, принадлежащей частотному диапазону 6...10 Гц;

3) отношение пиковых энергий S -фазы и P -фазы в частотном диапазоне 1...3 Гц;

4) отношение средних энергий S -фазы и P -фазы в частотном диапазоне 6...10 Гц;

5) доля средней энергии S -фазы, принадлежащей частотному диапазону 1...3 Гц.

Каждая точка определяется пятью признаками. Как следует из работы [98], надежное разделение точек при количестве признаков, меньшем 5, не достигается. Этот вывод подтвержден и нашим исследованием, когда рассматривались 25 точек для первого класса и 28 точек для второго.

Несколько вариантов расчетов при количестве признаков, равном 5, представлено на рис. 9.12–9.15. Погрешности измерений не известны и задаются в задаче.

На рис. 9.12 и 9.13 приведены результаты идентификации данных по взрывам и землетрясениям для случая квадратичной разделяющей поверхности

$$y = a_0 + \sum_{i=1}^5 a_i x_i + \sum_{i=1}^5 a_{5+i} x_i^2.$$

На графиках приводятся расстояния от соответствующих точек до разделяющей поверхности. По оси абсцисс отложены номера

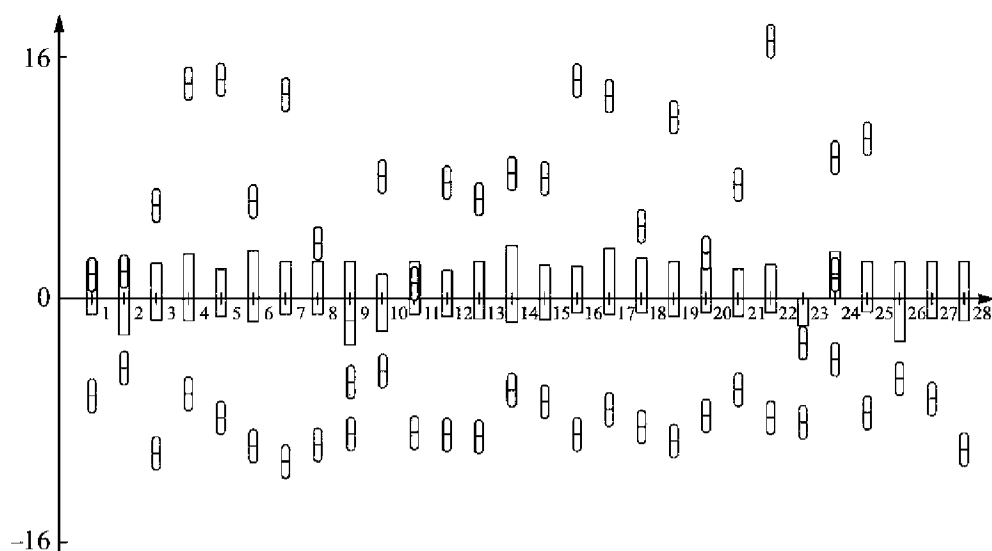


Рис. 9.12. Положение квадратичной разделяющей поверхности при $b_1 = b_2 = 0, 1$

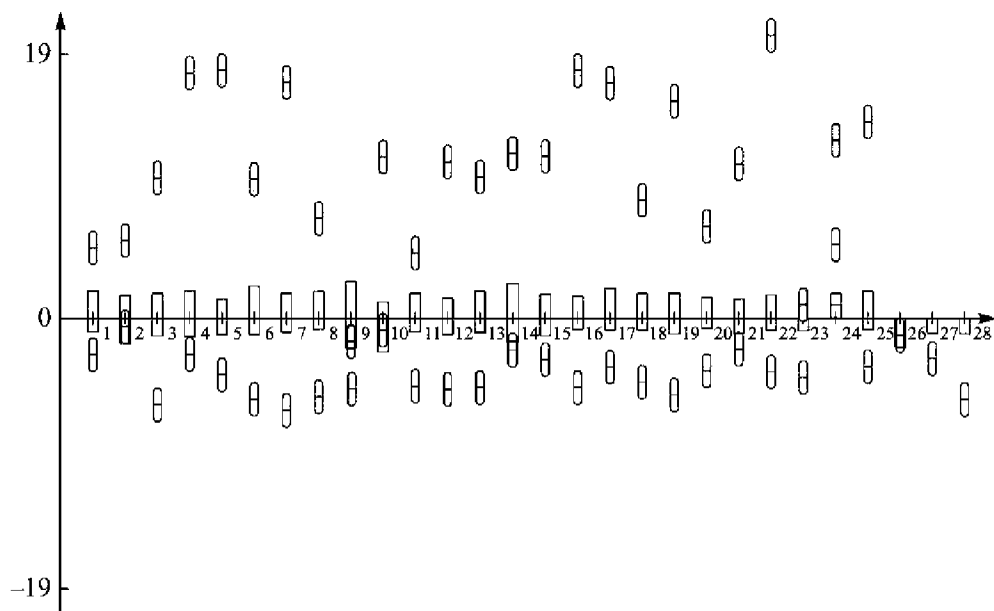


Рис. 9.13. Положение квадратичной разделяющей поверхности при $b_1 = 0,05$ и $b_2 = 0,12$

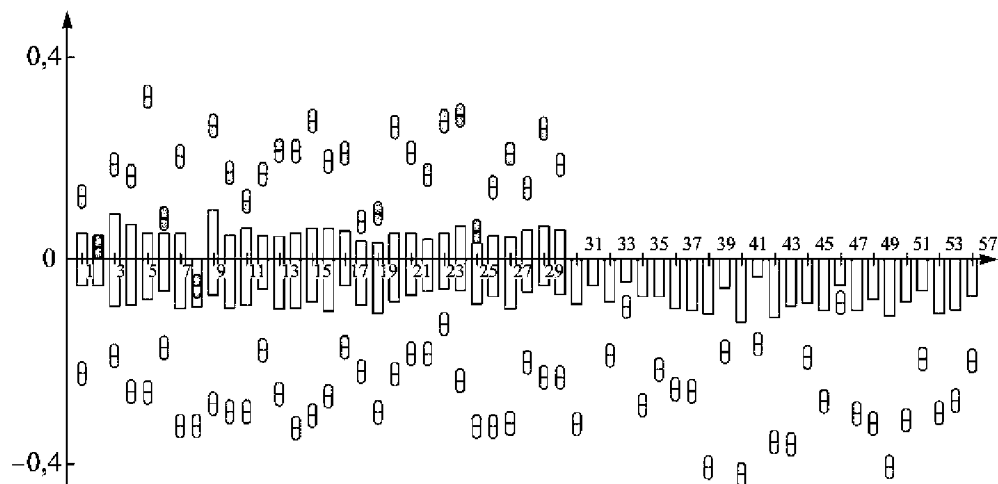


Рис. 9.14. Положение квадратичной разделяющей поверхности при $b_1 = 0,025$ и $b_2 = 0,02$

точек наблюдений. Светлые и темные эллипсы соответствуют точкам первого и второго классов. Их вертикальный размер определяется ошибками измерений. Прямоугольники, примыкающие одной стороной к оси абсцисс, определяют интервальные оценки для соответствующих точек разделяющей поверхности. Каждый прямоугольник расположен выше или ниже горизонтальной оси — в направлении точки, нулевую зону которой он определяет.

Рис. 9.12 соответствует значениям $b_1 = b_2 = 0,1$. В этом случае неправильно идентифицируются точки 24 (класс 1), 9 и 23 (класс 2), как и для линейного случая. Подбор вектора допуска: $b_1 = 0,05$ и $b_2 = 0,12$ (рис. 9.13), позволяет несколько улучшить ситуацию: точка 23 из класса 2 теперь идентифицируется правильно.

Приведем результаты расчетов для тех же классов, но на основе сейсмических данных для другого района (рис. 9.14). Здесь классу 1 (землетрясения) принадлежат 55 точек и классу 2 (взрывы) — 30.

На рис. 9.14 приведены результаты расчетов для квадратичного случая, который оказывается более предпочтительным, чем случай линейной разделяющей функции, когда неверно идентифицируются по три точки в каждом классе. Вариации вектора допуска привели к улучшению результатов распознавания: при $b_1 = 0,025$, $b_2 = 0,02$ теперь только по одной точке из каждого класса идентифицируются неправильно, хотя они и лежат в пределах интервальных оценок для разделяющей поверхности.

§ 9.9. Учет интервальных оценок функций плотности вероятности

в последовательных методах распознавания образов

Пусть известны функции плотности вероятности $p(x|\omega_i)$ некоторого признака x для различных объектов $\{\omega_i\}$, $1, 2, \dots, n$. Кроме того, известны интервальные оценки этих функций, значит, известны интервальные оценки ошибок α и β первого и второго рода, которые имеют место при принятии гипотез $H_1: x \sim \omega_1$ и $H_2: x \sim \omega_2$.

Рассмотрим, как изменится задача последовательного распознавания образов при учете погрешностей в оценке условных функций плотности распределения вероятностей. В частности, как изменятся верхние и нижние пороги (останавливающие границы) и среднее число наблюдений до принятия решений в данном случае.

Воспользуемся для решения данной задачи последовательным критерием отношения вероятностей Вальда. Критерий предназначен для решения задачи о выборе между двумя простыми гипотезами. Рассматривается отношение правдоподобия λ_n между классами ω_1 и ω_2 , имеющее вид

$$\lambda_n = \prod_{i=1}^n \frac{p(x_i | \omega_1)}{p(x_i | \omega_2)}.$$

Наблюдения проводятся до тех пор, пока $B < \lambda_n < A$, или $\ln B < \ln \lambda_n < \ln A$, где $A = \frac{1-\alpha}{\beta}$, $B = \frac{\alpha}{1-\beta}$. Наблюдения прекращаются и принимается гипотеза $H_1: x \sim \omega_1$, как только выполнены неравенства $\lambda_n \geq A$ или $\ln \lambda_n \geq \ln A$. Гипотеза $H_2: x \sim \omega_2$ принимается при выполнении неравенств $\ln \lambda_n \leq \ln B$ или $\lambda_n \leq B$.

В рассматриваемом случае будем учитывать погрешности Δp_1 и Δp_2 в определении условных функций плотности, т. е. для первого измерения x_1 имеем

$$\ln \lambda_1 = \ln \frac{p(x_1 | \omega_1) + \Delta p_1}{p(x_1 | \omega_2) + \Delta p_2}.$$

Если функции плотности распределения вероятностей подчиняются нормальному закону с математическими ожиданиями m_1 и m_2 , одинаковыми дисперсиями σ^2 и погрешностями оценок параметров Δm_1 , Δm_2 , $\Delta \sigma$, то

$$\begin{aligned} p(x | \omega_i) &= \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{(x-m_i)^2}{2\sigma^2}}, \\ p(x | \omega_i) + \Delta p_i &= \\ &= \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{(x-m_i)^2}{2\sigma^2}} \left[1 + \frac{\Delta m_i}{\sigma^2} (x - m_i) + \frac{(x - m_i)^2}{\sigma^3} \Delta \sigma - \frac{\Delta \sigma}{\sigma} \right]. \end{aligned}$$

Здесь приращение $\Delta p_i = \Delta p(x | \omega_i)$ заменили дифференциалом $dp(x | \omega_i)$. Тогда

$$\begin{aligned} \ln \lambda_1 &= \frac{1}{\sigma^2} \left[(m_1 - m_2)x_1 - \frac{1}{2}(m_1^2 - m_2^2) \right] + \\ &+ \ln \frac{\sigma^3 + \sigma \Delta m_1 (x_1 - m_1) + (x_1 - m_1)^2 \Delta \sigma - \sigma^2 \Delta \sigma}{\sigma^3 + \sigma \Delta m_2 (x_1 - m_2) + (x_1 - m_2)^2 \Delta \sigma - \sigma^2 \Delta \sigma} \approx \\ &\approx \frac{1}{\sigma^2} \left\{ x_1 \left[\left(1 - \frac{2\Delta \sigma}{\sigma} \right) (m_1 - m_2) + \Delta m_1 - \Delta m_2 \right] - \right. \end{aligned}$$

$$-\frac{1}{2}\left(1-\frac{2\Delta\sigma}{\sigma}\right)(m_1^2-m_2^2)-m_1\Delta m_1+m_2\Delta m_2\Big\}.$$

Если $\lambda_1 \geq A$ или $\ln \lambda_1 \geq \ln A$, то истинной является гипотеза о том, что $x \sim \omega_1$, т. е. если

$$x_1 \geq \frac{\sigma^2 \ln A}{\left(1-\frac{2\Delta\sigma}{\sigma}\right)(m_1-m_2)+\Delta m_1-\Delta m_2} + \\ + \frac{1}{2} \frac{\left(1-\frac{2\Delta\sigma}{\sigma}\right)(m_1^2-m_2^2)-m_1\Delta m_1+m_2\Delta m_2}{\left(1-\frac{2\Delta\sigma}{\sigma}\right)(m_1-m_2)+\Delta m_1-\Delta m_2},$$

то принимается решение $x \sim \omega_1$.

При $\Delta m_1 = \Delta m_2 = \Delta m$ данное условие примет вид

$$x_1 \geq \frac{\sigma^2 \ln A}{\left(1-\frac{2\Delta\sigma}{\sigma}\right)(m_1-m_2)} + \frac{1}{2}(m_1+m_2) - \frac{\Delta m}{2\left(1-\frac{2\Delta\sigma}{\sigma}\right)}.$$

Если $\Delta\sigma = 0$, то получим

$$x_1 \geq \frac{\sigma^2 \ln A}{m_1-m_2} + \frac{1}{2}(m_1+m_2) - \frac{\Delta m}{2}.$$

Мы здесь считали, что погрешности имеют один знак (направлены в одну сторону). Даже в этом случае получили условие, отличающееся от традиционных. Вернее было бы определить максимальный разброс, сложив модули погрешностей. Чтобы не усложнять формулы, мы этого здесь делать не будем.

После второго измерения при $\Delta m_1 = \Delta m_2 = \Delta m$ имеем

$$\ln \lambda_2 = \ln \frac{p(x_1 | \omega_1) + \Delta p_{11}}{p(x_1 | \omega_2) + \Delta p_{21}} + \ln \frac{p(x_2 | \omega_1) + \Delta p_{12}}{p(x_2 | \omega_2) + \Delta p_{22}} = \\ = \frac{m_1-m_2}{\sigma^2} \left[(x_1+x_2) \left(1-\frac{2\Delta\sigma}{\sigma}\right) - \left(1-\frac{2\Delta\sigma}{\sigma}\right)(m_1+m_2) - \Delta m \right].$$

При n измерениях x_1, x_2, \dots, x_n , когда их истинные значения соответственно равны $\xi_1, \xi_2, \dots, \xi_n$, получаем

$$\ln \lambda_n = \frac{m_1-m_2}{\sigma^2} \left[\sum_{i=1}^n x_i \left(1-\frac{2\Delta\sigma}{\sigma}\right) - \frac{n}{2} \left(1-\frac{2\Delta\sigma}{\sigma}\right)(m_1+m_2) - \frac{n}{2} \Delta m \right],$$

и процедура классификации будет следующей: если

$$\sum_{i=1}^n x_i \geq \frac{\sigma^2 \ln A}{(m_1 - m_2) \left(1 - \frac{2\Delta\sigma}{\sigma}\right)} + \frac{n}{2}(m_1 + m_2) + \frac{n\Delta m}{2 \left(1 - \frac{2\Delta\sigma}{\sigma}\right)},$$

то принимаем гипотезу, что $x \sim \omega_1$; если

$$\sum_{i=1}^n x_i \leq \frac{\sigma^2 \ln B}{(m_1 - m_2) \left(1 - \frac{2\Delta\sigma}{\sigma}\right)} + \frac{n}{2}(m_1 + m_2) + \frac{n\Delta m}{2 \left(1 - \frac{2\Delta\sigma}{\sigma}\right)},$$

то принимаем гипотезу, что $x \sim \omega_2$; если

$$\begin{aligned} & \frac{\sigma^2 \ln B}{(m_1 - m_2) \left(1 - \frac{2\Delta\sigma}{\sigma}\right)} + \frac{n}{2}(m_1 + m_2) + \frac{n\Delta m}{2 \left(1 - \frac{2\Delta\sigma}{\sigma}\right)} < \sum_{i=1}^n x_i < \\ & < \frac{\sigma^2 \ln A}{(m_1 - m_2) \left(1 - \frac{2\Delta\sigma}{\sigma}\right)} + \frac{n}{2}(m_1 + m_2) + \frac{n\Delta m}{2 \left(1 - \frac{2\Delta\sigma}{\sigma}\right)}, \end{aligned}$$

то следует провести новое измерение.

Из условия (7.12) следует, что

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n x_i = \sum_{i=1}^n \xi_i,$$

поэтому погрешность измерения наблюдений x будет заметно влиять только при первых наблюдениях, и в процедурах классификации тогда вместо x_i следует брать $x_i \pm 3\sigma(x_i)$, где $\sigma(x_i)$ — среднее квадратическое отклонение погрешности значения x_i .

Останавливающие границы получены при условии, что

$$A = \frac{1 - \alpha}{\beta}, \quad B = \frac{\alpha}{1 - \beta},$$

где α — ошибка первого рода, β — ошибка второго рода. Пока здесь интервальные оценки α и β не учтены. Чтобы их учесть, выберем $(\ln B)_{\min}$ и $(\ln A)_{\max}$:

$$(\ln A)_{\max} = \ln(A + \Delta A) = \ln A + \ln \left(1 + \frac{\Delta A}{A}\right) \cong \ln A + \frac{|\Delta A|}{A},$$

$$(\ln B)_{\min} = \ln(B - \Delta B) \cong \ln B - \frac{|\Delta B|}{B}.$$

Оценим, как изменится среднее число измерений при учете погрешности наблюдений.

В традиционных методах определено, что среднее число наблюдений, когда истинна гипотеза H_1 , равно \bar{n}_1 :

$$\bar{n}_1 = \frac{1}{\bar{z}} [(1 - \alpha) \ln A + \alpha \ln B],$$

где $\bar{z} = \ln \frac{p(x|\omega_1)}{p(x|\omega_2)}$.

Если истинна гипотеза H_2 , то

$$\bar{n}_2 = \frac{1}{\bar{z}} [\beta \ln A + (1 - \beta) \ln B].$$

В рассматриваемом случае, когда известны интервальные оценки α и β , необходимо взять верхние границы $\ln A$ и $\ln B$ и соответственно верхние или нижние границы α и β . Тогда новое среднее число измерений будет

$$\begin{aligned} \bar{n}_{1H} &= \frac{1}{\bar{z}} \left[(1 - \alpha + \Delta\alpha) \left(\ln A + \frac{|\Delta A|}{A} \right) + (\alpha + \Delta\alpha) \left(\ln B - \frac{|\Delta B|}{B} \right) \right] = \\ &= \frac{1}{\bar{z}} \left[(1 - \alpha + \Delta\alpha) \frac{|\Delta A|}{A} + \Delta\alpha (\ln A + \ln B) - \frac{|\Delta B|}{B} (\alpha + \Delta\alpha) \right] + \bar{n}_1. \end{aligned} \quad (9.7)$$

Здесь первое слагаемое в квадратных скобках — поправка к среднему числу наблюдений.

Рассмотрим

$$\bar{z}_1 = \ln \frac{p(x|\omega_1) + \Delta p_1}{p(x|\omega_2) + \Delta p_2} \cong \bar{z} + \frac{\Delta p_1}{p(x|\omega_1)} - \frac{\Delta p_2}{p(x|\omega_2)}. \quad (9.8)$$

Поскольку мы оцениваем среднее значение (результат многократных наблюдений), то можно считать, что при

$$\frac{\Delta p_1}{p(x|\omega_1)} = \frac{\Delta p_2}{p(x|\omega_2)},$$

т. е. при одинаковой относительной погрешности для условных функций плотности распределения вероятностей, знаменатель дроби в формуле (9.7) при определении \bar{n}_{1H} не изменяется. Поэтому будем считать, что не меняется и \bar{z}_H . (Это не означает, что не будут меняться другие параметры, определяемые из условия (9.8).)

Среднее число наблюдений (если верна гипотеза H_2 , с учетом интервальных оценок α , β , A и B) находится по формуле

$$\begin{aligned}\bar{n}_{2H} &= \frac{1}{z} \left[(\beta + \Delta\beta) \left(\ln A + \frac{|\Delta A|}{A} \right) + (1 - \beta + \Delta\beta) \left(\ln B - \frac{|\Delta B|}{B} \right) \right] = \\ &= \bar{n}_2 + \frac{1}{z} \left[(\beta - \Delta\beta - 1) \frac{|\Delta B|}{B} + \Delta\beta \ln(AB) + \frac{|\Delta A|}{A} (\beta + \Delta\beta) \right],\end{aligned}$$

где $\Delta A/A$ и $\Delta B/B$ находят из предположения, что погрешности ΔA , ΔB , $\Delta\alpha$, $\Delta\beta$ независимы и подчиняются нормальному закону распределения с дисперсиями, соответственно равными $(\Delta A)^2$, $(\Delta B)^2$, $(\Delta\alpha)^2$, $(\Delta\beta)^2$.

По формулам переноса ошибок получим

$$\begin{aligned}\left(\frac{\Delta A}{A} \right)^2 &= \left(\frac{\Delta\alpha}{1 - \alpha} \right)^2 + \left(\frac{\Delta\beta}{\beta} \right)^2, \\ \left(\frac{\Delta B}{B} \right)^2 &= \left(\frac{\Delta\alpha}{\alpha} \right)^2 + \left(\frac{\Delta\beta}{1 - \beta} \right)^2.\end{aligned}$$

§ 9.10. Сравнение зон неопределенности. Общий алгоритм принятия решений

Как было показано, поскольку в реальных условиях невозможно провести измерение без погрешностей, процедуру методов последовательного принятия решений используют и в процессе принятия решений по выборке фиксированного объема. Представляет интерес сравнить зоны неопределенности, полученные в процессах принятия решений по выборке фиксированного объема и в последовательных методах. Другими словами, не является ли зона неопределенности (нулевая зона) при анализе выборки фиксированного объема столь незначительной, что ее учет представляет только теоретический интерес.

Одномерный случай

Пусть некоторый признак объекта x распределен по нормальному закону с математическим ожиданием m_1 и дисперсией σ_1^2 для гипотезы H_0 и с математическим ожиданием m_2 и дисперси-

ей σ_2^2 для гипотезы H_1 . При равных вероятностях гипотез H_0 и H_1 решающая граница x_0 имеет вид

$$x_0 = \frac{1}{2}(\hat{m}_1 + \hat{m}_2).$$

В этом случае дисперсия определяется по формуле

$$D(x_0) = \frac{1}{4} [\sigma^2(\hat{m}_1) + \sigma^2(\hat{m}_2)],$$

где $\sigma^2(\hat{m}_i)$ — дисперсия оценки математического ожидания \hat{m}_i , $i=1, 2$. При одинаковых дисперсиях $\sigma^2(\hat{m}_1) = \sigma^2(\hat{m}_2) = \sigma^2(\hat{m})$ имеем

$$\sigma(x_0) = \frac{\sigma(\hat{m})}{\sqrt{2}}.$$

Во многих случаях, когда инструментальная погрешность измерения не является определяющей, оценкой математического ожидания нормального закона распределения является среднее арифметическое результатов наблюдений \bar{x} :

$$\hat{m} = \bar{x} = \sum_{i=1}^n x_i,$$

отсюда получаем среднее квадратическое отклонение

$$\sigma(\hat{m}) = \sigma(\bar{x}) = \frac{\sigma(x)}{\sqrt{n}},$$

где $\sigma^2(x)$ — дисперсия наблюдаемых значений x .

Ширина зоны неопределенности в этом случае будет такой

$$\frac{2t\sigma(\hat{m})}{\sqrt{2}} = \frac{\sqrt{2} \cdot t\sigma(x)}{\sqrt{n}},$$

где t — квантиль нормального распределения, определяемая доверительной вероятностью, $t=1, 2, \dots$

В методе последовательного принятия решений, когда решение принимается по среднему значению \bar{x} , ширина полосы неопределенности в случае нормального распределения признака с математическими ожиданиями m_1 и m_2 для разных гипотез и с одинаковыми дисперсиями σ^2 для обеих гипотез при числе наблюдений, равном n , определяется выражением

$$L = \frac{\sigma^2}{|m_1 - m_2| n} \ln \frac{A}{B},$$

где A и B — соответственно останавливающие верхние и нижние границы:

$$A = \frac{1 - \alpha}{\beta}, \quad B = \frac{\alpha}{1 - \beta}.$$

Здесь α и β — ошибки первого и второго рода.

Пусть $|m_1 - m_2| = k\sigma$. Тогда ширина полосы неопределенности определяется по формуле

$$L = \frac{\sigma}{kn} \ln \frac{(1 - \alpha)(1 - \beta)}{\beta\alpha}.$$

Чем больше разность математических ожиданий m_1 и m_2 , т. е. чем больше k , тем меньше α и β и тем уже полоса неопределенности. Найдем конкретные значения ширины полосы неопределенности для равновероятных гипотез. Пусть $|m_1 - m_2| = 4\sigma$; тогда $\alpha = \beta \cong (1 - 0,95)/2 = 0,025$. При $|m_1 - m_2| = 2\sigma$ получим ошибки $\alpha = \beta = (1 - 0,68)/2 = 0,16$. В первом случае ширина полосы неопределенности будет $\approx 1,5\sigma/n$, во втором $\approx 2\sigma/n$.

Приведенные результаты показывают, что зоны неопределенности в задачах принятия решений при фиксированном объеме выборки и в последовательных методах соизмеримы, поскольку $\sigma(x)$ наиболее часто совпадает с σ . Причем отношение ширины полос в этих методах принятия решений равно

$$\frac{\sqrt{2} \cdot t\sigma(x)kn}{\sqrt{n} \cdot \sigma \ln \frac{(1 - \alpha)(1 - \beta)}{\alpha\beta}} \cong \frac{\sqrt{2} \cdot tk\sqrt{n}}{\ln \frac{(1 - \alpha)(1 - \beta)}{\alpha\beta}} \approx \sqrt{n}.$$

Следовательно, в процессе принятия решения при фиксированном объеме выборки ширина зоны неопределенности может быть даже больше, чем в методе последовательного принятия решений.

Двумерный случай

Рассмотрим задачу принятия решений по двум признакам x и y , имеющим распределение Гаусса. Для простоты будем считать, что решающая граница — прямая линия.

Уравнение линейной решающей функции приводилось в § 6.3:

$$x^T K^{-1} (M_1 - M_2) - \frac{1}{2} (M_1 + M_2)^T K^{-1} (M_1 - M_2) + \ln \frac{P(\omega_1)}{P(\omega_2)} = 0,$$

где x — текущий вектор; M_i — вектор математического ожидания для i -й гипотезы, $i = 1, 2$; K — ковариационная матрица, одинаковая для обеих гипотез; $P(\omega_i)$ — вероятность i -й гипотезы, $i = 1, 2$.

В рассматриваемом случае при равновероятных гипотезах и единичной ковариационной матрице уравнение решающей прямой имеет вид

$$x(m_{1x} - m_{2x}) + y(m_{1y} - m_{2y}) - \frac{1}{2}(m_{1x}^2 - m_{2x}^2 + m_{1y}^2 - m_{2y}^2) = 0.$$

Здесь m_{ix} , m_{iy} — математические ожидания соответственно координат x и y для i -й гипотезы, $i = 1, 2$.

В процессе наблюдений необходимо найти оценки \hat{m}_{ij} и их дисперсии $D(\hat{m}_{ij})$, а также оценки и дисперсии оценок элементов ковариационной матрицы K (если они не исключены из рассмотрения).

Представим уравнение прямой в традиционном виде:

$$y = \theta_0 + \theta_1 x,$$

где

$$\theta_0 = \frac{m_{1x}^2 - m_{2x}^2 + m_{1y}^2 - m_{2y}^2}{2(m_{1y} - m_{2y})}, \quad \theta_1 = -\frac{m_{1x} - m_{2x}}{m_{1y} - m_{2y}}.$$

Интервальную оценку прямой (интервальную оценку значений y) можно найти, пользуясь формулой

$$D(\hat{y}) = (1, x) D(\hat{\theta}) \begin{pmatrix} 1 \\ x \end{pmatrix};$$

здесь $D(\hat{\theta})$ выражается через \hat{m}_{ij} и $D(\hat{m}_{ij})$. Последняя формула определяет дисперсию значений \hat{y} для любого значения x , тем самым и интервальные оценки решающей границы.

В методе последовательного принятия решений в двумерном случае, когда признаки независимы и подчиняются нормальному закону распределения с математическими ожиданиями $M_1 = (m_{1x}, m_{1y})$ и $M_2 = (m_{2x}, m_{2y})$ и общей ковариационной матрицей

$K = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$, после проведения первого измерения (x_1, y_1) выражение $\ln \lambda_1$ будет иметь следующий вид:

$$\begin{aligned} \ln \lambda_1 &= \ln \frac{p(x_1, y_1 | \omega_1)}{p(x_1, y_1 | \omega_2)} = \\ &= \frac{(m_{1x} - m_{2x}) \left(x_1 - \frac{m_{1x} + m_{2x}}{2} \right)}{\sigma_x^2} + \frac{(m_{1y} - m_{2y}) \left(y_1 - \frac{m_{1y} + m_{2y}}{2} \right)}{\sigma_y^2}. \end{aligned}$$

После n наблюдений $x_1, y_1, \dots, x_n, y_n$ получим

$$\ln \lambda_n = \sum_{i=1}^n \ln \frac{p(x_i, y_i | \omega_1)}{p(x_i, y_i | \omega_2)},$$

или

$$\begin{aligned} \ln \lambda_n &= \frac{(m_{1x} - m_{2x}) \left[\sum_{i=1}^n x_i - \frac{n}{2}(m_{1x} + m_{2x}) \right]}{\sigma_x^2} + \\ &+ \frac{(m_{1y} - m_{2y}) \left[\sum_{i=1}^n y_i - \frac{n}{2}(m_{1y} + m_{2y}) \right]}{\sigma_y^2}. \end{aligned}$$

Рассматриваем неравенства $\ln \lambda_n \geq \ln A$ и $\ln \lambda_n \leq \ln B$ и принимаем решение о прекращении или продолжении наблюдений. Несмотря на то, что выражения получились более громоздкими, чем в одномерном случае, их структура не изменилась: останавливающие границы будут представлять собой две параллельные прямые.

Границы зоны неопределенности, полученные в процессе принятия решений по выборке фиксированного объема, когда решающая функция есть прямая линия, описываются гиперболами. Ширина зоны неопределенности зависит от значений признаков x и y . Если $x = \bar{x}$ и $y = \bar{y}$, то ширина зоны будет минимальной. По мере удаления от точки с координатами (\bar{x}, \bar{y}) зона неопределенности будет резко увеличиваться.

Для других видов решающих функций $y = f(x_1, \dots, x_m)$ от случайных значений признаков x_1, x_2, \dots, x_m интервальная оценка так-

же находится с помощью вычисления дисперсии значений \hat{y} . Дисперсия $D(\hat{y})$ может быть определена по формуле

$$D(\hat{y}) = \sum_{i=1}^m \left(\frac{\partial y}{\partial x_i} \right)^2 D(x_i) + 2 \sum_{i=1}^m \sum_{\substack{j=2 \\ j>i}}^m \frac{\partial y}{\partial x_i} \frac{\partial y}{\partial x_j} D(x_i, x_j),$$

где $D(x_i, x_j)$ — корреляционный момент для случайных величин x_i и x_j .

Итак, показано, что в любом методе принятия решений из-за погрешностей исходных данных образуется область неопределенности с останавливающими границами: верхними $D_v(x)$ и нижними $D_n(x)$. Это могут быть или границы интервальных оценок решающих функций, или обычные останавливающие границы, получающиеся в методе последовательного принятия решений. Пусть разность решающих границ между i - и j -областями $D(x) = D_i(x) - D_j(x)$ равна нулю; тогда для любой задачи принятия решений (для последовательных методов или при фиксированном объеме выборки) справедлив следующий алгоритм:

- 1) если истинные значения точки наблюдения «выше» границы $D_v(x)$, то $x \sim \omega_i$;
- 2) если истинные значения точки наблюдения «ниже» границы $D_n(x)$, то $x \sim \omega_j$;
- 3) если истинные значения точки наблюдения расположены «между» границами $D_v(x)$ и $D_n(x)$, то следует продолжить наблюдения.

ПОСТРОЕНИЕ ПРОГНОЗОВ

§ 10.1. Особенности процедуры прогнозирования

Процедуры построения прогнозов используются почти во всех областях знания, в том числе в экономике, социологии, технике, образовании и т. д. Процесс построения прогноза можно представить в виде двух взаимно связанных задач:

- 1) построение модели исследуемого явления;
- 2) оценка основных характеристик (параметров) модели по базовым данным и получение по этой модели интервальной оценки прогноза.

Как правило, эти задачи дополняют друг друга, процесс построения прогноза часто бывает итерационным, состоящим в оценке параметров модели по базовым данным, и полученные интервальные оценки прогноза могут послужить основанием для изменения исходной модели и последующего пересчета прогноза. В задаче расчета прогноза особое внимание должно быть уделено учету погрешностей исходных данных при оценке характеристик (параметров) модели и процедуре получения интервальных оценок прогнозов.

Решение первой задачи базируется на использовании физических законов, законов развития общества, а часто и на интуиции, вторая задача может быть решена методами математической статистики. Таким образом, если в силу законов природы, технических закономерностей и законов общественного развития может быть получена модель явления, то основная проблема получения достоверного прогноза переносится на выбор математического метода.

В процессе построения модели какого-либо явления опираются, как правило, на наиболее устойчивые события. Например, наилучшим условием для построения модели и последующего прогноза является постоянство процесса или параметров модели. Поэтому при построении модели наряду с интегральными методами и элементарными функциями используют конечные разности такого порядка, при котором процесс считается стационарным. Однако следует иметь в виду, что погрешности значений конечных

разностей намного больше, чем погрешности значений исходных данных, по которым получены конечные разности.

Модели бывают концептуальные, физические или математические (иначе: феноменологические, эмпирические и аналитические), в зависимости от того, какой аспект явления в данном случае наиболее существен, от методов, которые можно использовать при построении модели, от количества и качества имеющейся информации.

Относительная простота вида модели является главной характеристикой модели. Во многих случаях для того, чтобы модель была применима, ее сложность должна находиться в определенном соответствии со сложностью описываемого объекта. Если физический механизм явления полностью ясен, можно составить математическое выражение, точно описывающее это явление. Часто для получения таких описаний (моделей) необходимы подробные сведения о явлении, которых может не быть, тогда приходится прибегать к построению эмпирической модели. Оба эти случая представляют собой крайности. Обычно используемые модели занимают промежуточное положение между ними. В частности, можно использовать некоторые теоретические представления для указания подходящего класса математических функций, для которых могут быть эмпирически подобраны число и параметры моделей по экспериментальным данным.

Теоретический анализ не только помогает выбрать подходящий вид модели, но и получить хорошие оценки числовых значений ее параметров. Эти значения затем проверяют, анализируя реальные данные. Результат такой проверки служит, в свою очередь, основанием для пересмотра модели.

Итерационный подход к построению моделей включает в себя следующие этапы.

1. На основе теории и практических сведений о явлении выбирается класс моделей с учетом тех целей, для которых создается модель.

2. Разрабатываются простейшие методы идентификации подклассов этих моделей. Процесс идентификации может быть использован для получения предварительных оценок параметров моделей.

3. Пробная модель подгоняется к экспериментальным данным, оцениваются ее параметры. Предварительные оценки, полученные на этапе идентификации, теперь можно использовать как началь-

ные значения в более точных итеративных методах оценивания параметров.

4. Диагностические проверки позволяют выявить возможные дефекты выбранной модели и диагностировать их причины. Если такие дефекты не выявлены, модель можно использовать. Если обнаружено какое-либо несоответствие оценок параметров модели, то итеративные циклы идентификации, оценок и диагностической проверки повторяют до тех пор, пока не будет найден подходящий вид модели.

В практике широко используют *параметрические модели*, когда общий вид модели известен, а оцениваются только параметры модели. Это повысило интерес к задачам оценивания параметров при построении таких моделей по экспериментальным данным.

При построении модели необходимо стремиться ответить на следующие вопросы:

- 1) как оценить качество модели?
- 2) как учесть всю имеющуюся информацию?
- 3) в чем состоит оптимальная стратегия получения недостающей информации?
- 4) что делать с нелинейностями?
- 5) можно ли аппроксимировать сложную систему простой моделью?

Ответы на эти вопросы зависят от конкретного класса систем. На практике отыскание подходящей модели может быть достаточно сложной задачей.

Математическую модель системы называют *детерминированной*, если входящие в нее описания воздействия и параметры модели являются постоянными, или детерминированными, функциями переменных состояния и времени. Математическую модель системы называют *статистической (стохастической)*, если функции, описывающие воздействия, и параметры модели являются случайными. Для стохастических (вероятностных) динамических систем текущее состояние $x(t_1)$ в момент t_1 и входное воздействие $\omega = \omega(t_1, t_2)$ определяют в момент t_2 не состояние $x(t_2)$, а лишь его вероятностное распределение.

Модели временных рядов и исследуемых процессов, необходимые для получения оптимального прогнозирования, в действительности являются стохастическими, поскольку на изучаемый процесс действует большое число неизвестных факторов и нельзя предложить детерминированную модель, допускающую точное

вычисление поведения объекта в будущем. Можно лишь вычислить вероятность того, что некоторое будущее значение принадлежит определенному интервалу. В дальнейшем будем различать вероятностную модель, или стохастический процесс, и наблюдаемый временной (вариационный) ряд z_1, z_2, \dots, z_N , который рассматривается как выборочная реализация стохастического процесса.

Важным классом стохастических процессов, рассматриваемых в практических задачах, является класс *стационарных процессов*. Системы называют *стационарными*, если их динамические свойства не изменяются с течением времени; если такое изменение имеет место, то системы называют *нестационарными*. Стационарность означает, что процесс преобразования системой входных возмущений обладает свойством инвариантности относительно сдвига по времени входных возмущений. Стационарный процесс остается в равновесии относительно постоянного среднего уровня, а нестационарный не имеет естественного среднего значения. Например, стохастическая модель, для которой прогнозирование экспоненциально взвешенным скользящим средним является оптимальным, относится к классу нестационарных процессов, называемых *процессами авторегрессии — проинтегрированного скользящего среднего* (АРПСС). Этот класс процессов обеспечивает множество как стационарных, так и нестационарных моделей, которые адекватно описывают многие встречающиеся на практике ряды наблюдений.

Полезным инструментом для описания поведения стационарных процессов является автокорреляционная функция. При гипотезе, что автокорреляция равна нулю, отношение оценки ρ_k к стандартной ошибке будет распределено примерно по нормальному закону с единичной дисперсией. То же справедливо и для частных автокорреляций. Эти факты можно использовать для получения нестрогих правил проверки того, являются ли теоретические автокорреляции и частные автокорреляции при задержках, больших некоторой, практически нулевыми.

Необходимо иметь в виду следующее:

- 1) по рядам наблюдений ограниченной длины не удастся доказать, что корень оператора авторегрессии, определяющий скорость стремления автокорреляций к нулю, точно равен единице;
- 2) не существует, конечно, резкого перехода от стационарного поведения необходимого ряда к нестационарному.

В сложных случаях некоторым преимуществом может обладать использование нестационарных моделей в задачах прогнозирования. Если использовать нестационарную модель, которая не содержит среднего значения μ , то прогнозы будущего поведения системы не будут зависеть от выборочного среднего, найденного по предыдущему периоду, которое может не иметь никакого отношения к будущему среднему уровню ряда.

Как было указано выше, этапу прогнозирования предшествует итеративная процедура построения математической модели, состоящая из этапов идентификации, оценивания и диагностической проверки.

При идентификации используется любая информация о том, как были получены рассматриваемые числовые значения (как был генерирован ряд наблюдений), с целью нахождения набора экономичных моделей, заслуживающих проверки. Экономичные модели должны обладать максимальной простотой и минимальным числом параметров, но при этом адекватно описывать наблюдения. Методы идентификации определяют класс пробных моделей, для которых применяются более формальные и эффективные методы оценивания.

Оценивание — это процедура получения оценок параметров моделей. Оценки определяют адекватность моделей. Неадекватность выбранной модели может быть вызвана неэффективностью процедуры оценки параметров модели, а не тем, что неадекватен вид модели. Процедуры получения оценок моделей рассмотрены в гл. 7.

Диагностическая проверка — это проверка согласования подобранной модели с исходными данными для того, чтобы обнаружить недостатки модели и улучшить ее. В частности, диагностическая проверка модели может базироваться на введении избыточного числа параметров, т. е. на оценивании параметров для несколько более общей модели, чем ожидаемая. Идея этого способа состоит в том, что можно предсказать неадекватные свойства модели и проводить исследования остаточных ошибок после подгонки модели, что позволяет определить, какие необходимы изменения в модели.

При проведении процедуры прогнозирования необходимо иметь в виду, что все факторы, влияющие на поведение системы в базовом (исследуемом) и прогнозируемом периодах, должны быть постоянны или изменяться по известному закону. Первый случай

реализуется в однофакторном прогнозировании, второй — при многофакторном.

Можно выделить два вида прогнозируемых характеристик системы, зависящих от времени: переменные состояния и переменные интенсивности. Переменная состояния определяется периодически, и ее значение в течение небольшого интервала времени не зависит от времени, прошедшего с момента начала наблюдения. Переменная интенсивности также определяется периодически, но ее значение пропорционально времени, прошедшему с момента предыдущего наблюдения. Если переменная состояния характеризует количество, то переменная интенсивности — скорость его изменения.

Величина промежутков времени между измерениями входных переменных системы с целью проверки и уточнения ранее сделанных прогнозов о выходных переменных зависит главным образом от длительности времени упреждения и наибольшей частоты циклических изменений в системе, которые должна отражать модель. Поэтому временные интервалы пересмотров прогнозов могут изменяться в широких пределах. Однако эти интервалы должны быть достаточно велики, чтобы обеспечивалась вероятность осуществления ожидаемых изменений системы. Когда имеют место какие-то периодические процессы, то частота наблюдений должна быть по крайней мере вдвое больше частоты изучаемого процесса. Если случайная ошибка при определении входных переменных велика по сравнению с измеряемой величиной, интервал уточнения прогноза для переменной интенсивности целесообразно увеличить, усредняя таким образом случайную ошибку. Однако для переменной состояния в аналогичном случае интервал уточнения прогноза лучше уменьшить, что позволяет для выделения полезной информации использовать соответствующие методы фильтрации. Использование доступных к моменту t наблюдений динамического ряда для прогнозирования его значений в некоторый момент $t+l$ в будущем является основой для управления и оптимизации промышленных процессов, экономических систем и т. д. Интервал l называют *временем упреждения*. Функция $y_t(l)$, позволяющая получить в момент t прогнозы будущих времен упреждения l , называется *прогнозирующей функцией* в момент t .

Прогнозирующая функция определяется заданной априори целью. Например, находится такая прогнозирующая функция $\hat{y}_t(l)$, у которой среднее значение квадрата отклонения $(y_{t+l} - \hat{y}_t(l))^2$ ис-

тинного значения y_{t+1} от прогнозируемого значения является наименьшим для каждого упреждения l .

Вычисление наилучшего прогноза должно сопровождаться указанием его точности, например, чтобы можно было оценить риск, связанный с принятием решения по данному прогнозу. Точность прогноза может быть выражена вероятностными пределами (интервальными оценками).

Динамический ряд рассматривается как сумма детерминированной и случайной компонент. Детерминированная компонента выражается некоторой аппроксимирующей функцией, отражающей закономерность развития исследуемого явления. Появление случайной компоненты определяется сложным взаимным влиянием параметров системы, влиянием на их величину большого числа неизвестных факторов, действующих в разных направлениях, что находит свое выражение в отклонении значений показателей системы от аппроксимирующей функции. Дополнительный вклад в величину случайного компонента вносит и аппроксимирующая модель, с помощью которой невозможно описать все особенности системы.

Наиболее часто отклонения наблюдаемых величин от аппроксимирующей функции, описывающей развитие события, рассматриваются как стационарный случайный процесс, к которому применимы методы прогнозирования стационарных случайных процессов. Если случайная компонента не является стационарной, то производят определенные преобразования, чтобы сделать случайную компоненту стационарной хотя бы в определенных условиях.

В прогнозировании важное значение имеет предварительный анализ характера изучаемого явления для определения вида его описания: процесс хорошо описывается основной тенденцией (трендом) или процесс зависит от изменения некоторого набора показателей, отражающих структуру процесса. Выбор вида описания предопределяет точность прогноза на будущее.

Методы прогнозирования иногда разделяют на три группы: статистические (описательные), причинно-следственные и их комбинаций. Для изучения исследуемого процесса необходимо задать закон изменения входных переменных по времени. Выходные переменные системы могут быть описаны с помощью некоторой модели, значения коэффициентов которой определяются подбором. При этом различные наблюдения учитываются с различными весовыми множителями. По таким моделям, включающим описание

предыстории системы, прогноз можно составить путем расчета состояния системы для некоторого будущего момента времени.

Если удастся построить модель окружающей среды, позволяющую выявить причины изменений в системе (вторая группа методов), то прогноз, полученный с помощью такой модели, объясняет будущее системы. Подобные методы охватывают широкий круг моделей.

Между двумя любыми автокоррелированными временными рядами всегда существует статистическая корреляция. Следовательно, существует опасность бессмысленного использования множественной регрессии в поисках «хорошего» коэффициента корреляции между прогнозируемыми (выходными) переменными и различными потенциально информативными входными переменными. Известная предыстория представляет собой только часть полного временного ряда. При длинных рядах наблюдаемых величин связь между переменными в любом случае будет найдена. Если с увеличением объема информации коэффициенты модели становятся равными нулю, то модель не пригодна для принятия правильного решения. Наилучшие результаты получаются при использовании комбинации статистических и причинно-следственных методов прогнозирования. Исходные данные обычно представляют собой результаты выборочных наблюдений либо переменной интенсивности, либо переменной состояния. Результаты наблюдений регистрируются с ошибками, которые возникают как при наблюдениях, так и при регистрации данных. Кроме того, изучаемый процесс может иметь стохастическую природу. Результаты наблюдений могут содержать и аномальные эффекты. Поэтому не каждую совокупность зарегистрированных по мере поступления реальных данных следует считать подходящим рядом значений, на основании которого можно составлять прогноз. Перед тем как подбирать коэффициенты модели по исходным данным, из последних должны быть исключены выбросы, т. е. результаты наблюдений, которые не характеризуют прогнозируемый процесс.

Для описания стохастических элементов рядов и их прогноза используются три различных понятия: помехи, остатки и ошибки. Понятие помехи связано с собственной изменчивостью процесса и неопределенностью, вносимой при наблюдении за этим процессом. Следовательно, помеха является составной частью используемых данных. Под остатками понимается разность между результатами наблюдений и соответствующими значениями, вычисленными с помощью прогнозирующей их модели. Таким образом,

остатки связаны с прошлыми данными и моделью, которая использовалась для их оценок. Ошибки прогноза представляют собой разность между прогнозом, сделанным в настоящее время, и тем прогнозом, что будет наблюдаться позднее, в момент времени, для которого составлен прогноз.

Любой процесс, представленный рядом результатов наблюдений, можно описать системой разностных уравнений (когда процесс дискретен) или дифференциальных уравнений (когда он непрерывен). Если в этих уравнениях коэффициенты не зависят от времени, процесс называется *стационарным*, если же зависят, то — *нестационарным*. Эта зависимость может носить вероятностный или регулярный характер. Известное регулярное изменение коэффициентов во времени может быть описано дополнительными уравнениями.

Для достаточно больших интервалов времени многие из прогнозируемых рядов являются нестационарными. Однако их все же можно считать квазистационарными, если прогноз составлять для какого-то одного момента времени.

Ни в одном из статистических методов прогнозирования не может быть заранее предусмотрено изменение модели прогнозируемого процесса. Существуют методы быстрого обнаружения изменений в последующих процессах с соответствующей реакцией на эти изменения. Модель прогноза может все более усложняться, когда это экономически оправдано, что позволяет глубже проникнуть в механизм наблюдаемых явлений.

Во многих случаях изменения изучаемого процесса можно предвидеть заранее, но в модель прогноза они не включаются, так как последствия таких изменений не могут быть точно рассчитаны. Тем не менее, на основе тщательного анализа различных вариантов можно предсказать характер изменений. В любой отдельный период времени существует, очевидно, несколько серий прогнозов, отличных от простого описательного прогноза. Это позволяет минимизировать время, затрачиваемое на внесение изменений.

§ 10.2. Модели для получения прогнозов

Для построения прогнозов необходимо в базисном периоде получить результаты наблюдений над прогнозируемой величиной (фактором). Эти наблюдения могут быть представлены в различных видах: мгновенные значения, проинтегрированные величины,

конечные разности. Вид полученных результатов наблюдений определяет последующий выбор модели для получения прогноза.

Построение прогнозов по векторной модели

Пусть в момент времени T задана последовательность результатов наблюдений y_t для некоторых моментов времени $t \leq T$. Прогнозирующая модель задает множество выходных переменных y_{t+l} , $l > 0$, которые могут быть выражены в векторной форме. В общем виде выражение для модели записывается в виде $y_{T+l} = a_T F(l)$, где вектор a представляет собой коэффициенты модели, получаемые по результатам наблюдений до момента T включительно, а матрица F размерностью $m \times T$ является набором аппроксимирующих функций, строки которой соответствуют каждому из m членов модели и столбцы — каждому из интервалов прогнозирования до момента T . Компоненты вектора y представляют собой прогнозируемые значения для каждого из интервалов на момент l . В большинстве практических задач из интервалов на момент l элементы матрицы F определяют время относительно момента самого последнего наблюдения, а значения коэффициентов a_T зависят от выбора начала отсчета времени. Широкий класс составляют модели, которые могут быть представлены с помощью полиномов F_i : $F_1(t) = 1$, $F_2(t) = t$, $F_3(t) = t(t-1)/2$ и т. д. Применение различных методов прогнозирования позволяет получить различные модели, например, за счет использования разностей или сумм функций F_i .

Коэффициенты a_T вычисляют сначала по методу наименьших квадратов в соответствии с N исходными данными $(m \times N)$ -матрицы F . Гармоники включаются в модель вплоть до той частоты, при которой квадрат амплитуды составляет по крайней мере четвертую часть от величины дисперсии остатков, получаемых при подгонке этой модели с учетом имеющихся степеней свободы для ошибок.

С каждым новым наблюдением вектор a_T заменяется вектором

$$a_T = L^T a_{T-1} + h e,$$

где L — матрица перехода размерностью $N \times m$, которая определяется с помощью выражения $f(t) = Lf(t-1)$ и переводит значения соответствующих функций из одного периода времени в следующий. Столбец матрицы F , имеющий номер t , обозначается

как $f(t)$, и e представляет собой ошибку (скаляр) самого последнего прогноза, сделанного на предыдущем интервале: $e = y_T - \hat{y}_T$. Вектор h , характеризующий m постоянных сглаживания, определяется следующим образом. Используя z -преобразования и некоторые элементарные тригонометрические соотношения, вводится матрица

$$H = \sum_j \alpha^j f(-j) f^T(-j).$$

Поскольку параметр α выбирается согласно неравенству $\alpha < 1$, то значимость более старых данных снижается и увеличивается вес, приписываемый более поздним данным. Следовательно, вектор сглаживания можно представить в виде $h = H^{-1}f(0)$. Величины L и h необходимо вычислить только один раз, так как при дальнейшем пересмотре прогнозов они не изменяются. Если прогнозы составляются для переменных интенсивности и результаты наблюдений получены в нерегулярные интервалы времени, то исходные данные преобразуются к величинам, характеризующим скорость их изменения на равномерных интервалах времени. Получаемые в результате прогнозы могут быть приведены в соответствие с величиной будущих интервалов времени. Заметим, что подобное использование сечения какой-либо входной переменной одинаково пригодно для любых рядов и не приводит к увеличению дисперсии, какое имело бы место в случае получения сечения из данных предыстории. Поскольку решение линейного дифференциального уравнения может быть представлено в виде полинома, то различные математические описания моделей означают лишь возможность выбора удобного метода анализа, а не фактическое различие в математических свойствах таких моделей. Тем не менее точность прогноза будет зависеть от способа математического описания явления.

Математические методы позволяют представить прогнозирующую модель в виде полинома любого порядка. Применение полиномов второго порядка для описания прогнозируемых рядов часто оказывается достаточным.

В процессе прогнозирования, как правило, нельзя предложить детерминированную модель, допускающую точное вычисление будущего поведения объекта, так как в ней может участвовать ряд неизвестных факторов. Однако можно предложить модель, позволяющую вычислять вероятность того, что некоторое будущее зна-

чение будет лежать в определенном интервале, — стохастическую вероятностную модель (или стохастический процесс).

Чтобы исследовать возможности прогнозирования по выбранной модели, ряд наблюдаемых значений разбивают на две части. Первую часть ряда рассматривают как предысторию, а вторую — как прогноз. Получив прогнозирующую функцию по базисному периоду, по второй части ряда можно оценить реальные ошибки прогноза. Изменяя число элементов рядов предыстории и прогноза, получим зависимость точности прогноза от периода предыстории и величины прогнозируемого периода.

Основные разностные модели

Для построения моделей будем использовать некоторые операторы:

- оператор сдвига назад B , $By_t = y_{t-1}$, $B^m y_t = y_{t-m}$;
- оператор сдвига вперед $F = B^{-1}$, $Fy_t = y_{t+1}$, $F^m y_t = y_{t+m}$;
- разностный оператор со сдвигом назад Δ :

$$\Delta y_t = y_t - y_{t-1} = (1 - B)y_t;$$

- оператор суммирования $S = \Delta^{-1}$:

$$Sy_t = \Delta^{-1} y_t = y_t + y_{t-1} + y_{t-2} + \dots = (1 + B + B^2 + \dots)y_t = (1 - B)^{-1} y_t.$$

1. Модель авторегрессии. Рассмотрим основные разностные модели для построения прогнозов. Здесь текущее значение процесса выражается как конечная линейная совокупность предыдущих значений процесса и независимого импульса a_t , который является реализацией случайной величины с фиксированным распределением, часто полагаемым нормальным с нулевым средним и дисперсией $\sigma^2(a)$ (белый шум), т. е.

$$y_t = \mu + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \dots + \Phi_p y_{t-p} + a_t,$$

где μ — параметр, определяющий уровень процесса (тренд); Φ_i , $i = 1, 2, \dots, p$, — неизвестные параметры. Пусть $\tilde{y}_t, \tilde{y}_{t-1}, \tilde{y}_{t-2}, \dots$ — отклонения от μ , например $\tilde{y}_t = y_t - \mu$. Тогда значения \tilde{y}_t находят по формуле

$$\tilde{y}_t = \Phi_1 \tilde{y}_{t-1} + \Phi_2 \tilde{y}_{t-2} + \dots + \Phi_p \tilde{y}_{t-p} + a_t,$$

которая называется *процессом авторегрессии (АР) порядка p* .

Введем оператор авторегрессии порядка p :

$$\Phi(B) = 1 - \Phi_1 B - \Phi_2 B^2 - \dots - \Phi_p B^p;$$

тогда модель авторегрессии имеет вид $\Phi(B)\tilde{y}_t = a_t$. Она содержит $p + 2$ неизвестных параметров μ , $\Phi_1, \Phi_2, \dots, \Phi_p$, $\sigma^2(a)$, которые необходимо оценить перед вычислением прогноза по наблюдениям. Применяя линейный оператор

$$\psi(B) = 1 + \psi_1 B + \psi_2 B^2 + \dots = \Phi^{-1}(B),$$

модель авторегрессии можем записать в виде

$$\tilde{y}_t = \psi(B)a_t;$$

здесь \tilde{y}_t выражается как бесконечная взвешенная сумма импульсов a_t, a_{t-1}, \dots . Процессы авторегрессии могут быть стационарными или нестационарными. Чтобы процесс был стационарным, необходимо чтобы коэффициенты ψ_1, ψ_2, \dots оператора $\psi(B) = \Phi^{-1}(B)$ образовывали сходящийся ряд.

2. Модель скользящего среднего. Пусть \tilde{y}_t линейно зависит от конечного числа q предыдущих значений a_{t-i} . Процесс

$$\tilde{y}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q}$$

называется *процессом скользящего среднего порядка q* ; θ_i , $i = 1, 2, \dots, q$, — оцениваемые параметры.

Введем оператор скользящего среднего порядка q , который имеет вид

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q.$$

Тогда $\tilde{y}_t = \theta(B)a_t$. Модель содержит $q + 2$ неизвестных параметров μ , $\theta_1, \theta_2, \dots, \theta_q$, $\sigma^2(a)$, подлежащих оценке по наблюдениям. Последовательность независимых случайных величин a_t, a_{t-1}, \dots (белый шум) вводится для генерации динамических рядов, в которых последовательные значения сильно зависимы. Например, белый шум a_t можно трансформировать в процесс y_t при помощи линейного фильтра $\psi(B)$:

$$y_t = \mu + a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \dots = \mu + \psi(B)a_t.$$

3. Смешанные модели авторегрессии — скользящего среднего. Для достижения большей гибкости процедуры подгонки модели к наблюдениям вводят комбинированную модель авторегрессии типа скользящего среднего, имеющую вид

$$\tilde{y}_t = \Phi_1 \tilde{y}_{t-1} + \dots + \Phi_p \tilde{y}_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q},$$

или

$$\Phi(B) \tilde{y}_t = \theta(B) a_t,$$

в которой содержится $p + q + 2$ параметров μ , Φ_1, \dots, Φ_p , $\theta_1, \dots, \theta_q$, $\sigma^2(a)$, подлежащих оценке по наблюдениям.

4. Нестационарные модели. Для многих рядов наблюдений уровни, относительно которых происходят флуктуации, являются различными для разных элементов ряда, однако при этом поведения их оказываются схожими. Так, для однородного нестационарного процесса можно ввести ряд, составленный из разностей Δ^d порядка d , который может быть стационарным. Для описания однородного нестационарного процесса вводят обобщенный оператор авторегрессии

$$\varphi(B) = \Phi(B)(1 - B)^d,$$

где $\Phi(B)$ — стационарный оператор.

Обобщенная модель для описания однородного нестационарного процесса имеет вид

$$\varphi(B) y_t = \theta(B)(1 - B)^d y_t = \theta(B) y_t,$$

или

$$\varphi(B) \omega_t = \theta(B) y_t, \quad \omega_t = \Delta^d y_t,$$

т. е.

$$\omega_t = \Phi_1 \omega_{t-1} + \dots + \Phi_p \omega_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}.$$

Этот процесс называется *процессом авторегрессии — проинтегрированного* (просуммированного) *скользящего среднего* (АРПСС) *порядка* (p, d, q) . Очевидно, что модель АРПСС является общей для всех рассмотренных здесь моделей. На практике адекватное описание стационарных и нестационарных процессов происходит при значениях p и q не больше чем два и при d , равных нулю или единице (максимум — двум).

§ 10.3. Сглаживание рядов с помощью скользящей средней

При представлении совокупности результатов наблюдений в виде рядов фактически используется предположение о том, что наблюдаемые величины имеют некоторое распределение, параметры которого и их изменение можно оценить. По этим параметрам (как правило, по среднему значению и дисперсии, хотя иногда используется и более полное описание) можно построить одну из моделей вероятностного представления процесса.

Другим вероятностным представлением является модель в виде частотного распределения с параметрами p_j для относительной частоты наблюдений, попадающих в j -й интервал. При этом если в течение принятого времени упреждения не ожидается изменение распределения, то решение принимается на основании имеющегося эмпирического частотного распределения.

Важнейшей задачей анализа является выявление основной тенденции поведения системы как результат влияния комплекса причин, действующих на изучаемый процесс. Основная тенденция поведения ряда характеризуется трендом. Для выявления основной тенденции поведения ряда применяют процесс сглаживания рядов динамики. Процесс сглаживания требует тщательного анализа, чтобы исключить возможность «сгладить» под видом случайных выбросов и отклонений существенные кратковременные изменения показателей, отражающих важные моменты в поведении системы.

Для сглаживания рядов динамики часто применяется метод наименьших квадратов. Заметим, что процесс сглаживания рядов динамики в действительности может быть описан гладкой кривой, а наблюдаемые выбросы есть результат наложения случайных помех. В этом случае процедура проведения сглаживания методом наименьших квадратов имеет четкое статистическое объяснение. Однако в других случаях отклонение сглаженной кривой от экспериментальной нельзя объяснить только влиянием случайных помех. В подобных случаях применяют метод наименьших квадратов, но здесь он определяет некоторую гладкую кривую, наилучшим образом (в выбранной метрике) проходящую по исходным данным. Ничего общего эта процедура со статистическим толкованием метода наименьших квадратов в данном случае не имеет.

В результате предварительного анализа (из физических условий задачи) выбирается класс функций, которыми может быть описано изучаемое явление. Параметры этих функций подлежат

определению. В ряде случаев параметры функций могут иметь определенный физический или экономический смысл. Для процедуры прогнозирования при выборе сглаживающей функции необходимо иметь в виду следующее: функция должна отражать основную закономерность развития явления, в отношении которой можно выдвинуть гипотезу, что эта закономерность сохранится и в будущем. Например, для процесса интерполяции достаточно, чтобы выбранная функция наиболее точно описывала значения ряда.

Для сглаживания рядов динамики нецелесообразно брать функции с большим числом параметров, в то же время выбранная функция должна быть адекватна исследуемому процессу. Следует иметь в виду, что короткие ряды в редких случаях дают возможность получить объективную информацию. Если рассматривать достаточно длинный ряд, то за это время может измениться поведение системы.

После определения параметров выбранных функций необходимо отобрать из них наиболее приемлемую. Знание оценок свободных параметров функций и их дисперсий дает возможность применять критерии согласия для выбора основной функции. В крайнем случае этот выбор можно провести по величине среднего квадратического отклонения значений элементов ряда и тех же значений, вычисленных по аппроксимирующей функции.

Как было уже отмечено, метод наименьших квадратов применим при детерминированных значениях аргументов (факторов) выбранных функций. Иногда условие детерминированности факторов не выполняется и следует использовать описанные в гл. 7 методы конфлюэнтного анализа. Все необходимые для сглаживания рядов динамики формулы приведены в гл. 7. Отметим еще раз необходимость учета всех погрешностей и неопределенностей исходных данных. В § 7.14 показано, например, что для функции $\eta = \xi_1 \xi_2^3 - \xi_1^2 \xi_2^2 + 2\xi_1^3$ регрессионные оценки параметров отличались до 10 % от истинных значений, а конфлюэнтные — до 2 %. При этом отличие в значениях функции от идеальных в базисном периоде составляет 2–3 %. Нетрудно теперь будет указать те значения ξ , при которых ошибка прогноза по регрессионной модели достигнет 10 %, а по конфлюэнтной модели — только 2 %.

Практически во всех применяемых в настоящее время методах прогнозирования коэффициенты моделей сначала определяются путем подгонки модели к некоторым данным предыстории, а затем проверяются и уточняются по мере поступления новых данных.

Выбор коэффициентов, как правило, осуществляется из условия минимизации суммы квадратов остаточных разностей между данными y_{T-j} и результатами расчета по модели \hat{y}_{T-j} с учетом различных весовых множителей ω_j , приписываемых остаткам в различные моменты времени. Так, в рамках метода МНК для векторной формы модели, записанной в общем виде, вектор коэффициентов a выбирается на основе минимизации величины

$$S = \sum_j \omega_j (y_{T-j} - \hat{y}_{T-j})^2,$$

где суммирование проводится по всем j вплоть до самого последнего наблюдения в момент времени T . При оценке тех или иных преимуществ различных систем весовых множителей ω_j необходимо учитывать не только достигаемую при их использовании точность прогноза, но и степень сложности соответствующих вычислений.

Скользящее среднее представляет собой оценку по методу наименьших квадратов единственной константы для представления исходных данных с одинаковыми весовыми множителями. Этому случаю соответствует простая функция $F(t)=1$ для всех t . При этом весовые множители ω_j равны 1 для N последних наблюдений в интервале $0 \leq j \leq N-1$ и равны 0 для $j \geq N$. Некоторые коэффициенты модели можно определять с помощью полиномов более высокого порядка путем подгонки модели к результатам N последних наблюдений в каждый момент поступления новых данных. Таким образом, в самом общем случае значения коэффициентов в любой заданный момент времени зависят от предыдущих значений коэффициентов, от ошибки в прогнозе при использовании самого последнего наблюдения, а также от вида используемого полинома и, конечно, числа N результатов наблюдений.

В методе скользящей средней первоначальные значения элементов ряда заменяются их средней арифметической величиной внутри выбранного интервала. Полученное значение относится к середине выбранного интервала. Затем интервал сдвигается на одно наблюдение и расчет средней величины ряда повторяется. Интервалы определения средней величины ряда берутся в течение всего времени одинаковыми. Чем шире интервал, тем более плавный вид имеет огибающая, проведенная по значениям элементов нового ряда. Сглаженный ряд короче первоначального на $k-1$ наблюдений (k — величина интервала сглаживания).

Величина интервала сглаживания определяется конкретной задачей. Если число членов интервала сглаживания нечетное, то полученное значение скользящей средней

$$\hat{y}_{i+m} = \frac{\sum_{k=0}^{2m} y_{i+k}}{2(m+1)}$$

приходится на средний член интервала сглаживания. При четном числе членов интервала сглаживания значения скользящих средних будут располагаться в промежутках между элементами ряда

$$\hat{y}_{i+m} = \frac{\frac{y_i}{2} + \sum_{k=0}^{2m-1} y_{i+k} + \frac{1}{2} y_{i+2m}}{2m}.$$

Здесь y_i — значение i -го члена ряда, m — целое положительное число, определяющее величину интервала сглаживания, $1 \leq m \leq n/2$:

$$m = \begin{cases} \frac{n}{2}, & \text{если } n \text{ — четное число,} \\ \frac{n-1}{2}, & \text{если } n \text{ — нечетное число,} \end{cases}$$

n — число членов ряда.

Если известно, что внутри интервала сглаживания имеет место нелинейный тренд, применяют взвешенные скользящие средние: внутри каждого интервала сглаживания значения элементов ряда описывают полиномом p -й степени:

$$y = a_0 + \sum_{i=1}^p a_i t^i.$$

Параметры a_0, a_1, \dots, a_p полинома находят, например, методом наименьших квадратов или другими методами, описанными в гл. 7. Взвешенную скользящую среднюю для выбранного интервала определяют как средний член сглаженных значений исходного ряда с помощью полинома.

Простота и наглядность — достоинства метода скользящей средней. Однако при малом числе наблюдений метод приводит к искажению тренда, величина интервала сглаживания влияет на форму тренда, теряются начальные и конечные элементы ряда.

Сглаживание рядов можно проводить и с помощью других функций и полиномов, а также с помощью метода конечных разностей. Для элементов ряда можно вычислить первые разности $\Delta_i^1 = y_{i+1} - y_i$, вторые разности $\Delta_i^2 = \Delta_{i+1}^1 - \Delta_i^1$ и т. д.

Если предположить, что элементы ряда можно описать полиномом p -й степени, то $(p+1)$ -я производная этого полинома будет всюду равна нулю, т. е. $(p+1)$ -я разность элементов такого ряда будет равна нулю. Таким образом, взяв p -ю разность от членов ряда, рассмотрим ряд, в котором исключен тренд, выраженный полиномом p -й степени.

§ 10.4. Прогнозирование с помощью экспоненциального сглаживания

Для стационарных рядов существует система весовых множителей, позволяющая обеспечить минимальную ошибку прогноза. Эти множители определяются видом автоковариационной функции. Известен метод вычисления таких оптимальных весовых функций для постоянного уровня, тренда и сезонных коэффициентов. В каждом из этих случаев весовые множители уменьшаются согласно поведению показательной функции α^j при $\alpha < 1$, а различные значения α , получаемые для уровня, тренда и сезонных коэффициентов, определяются систематическими исследованиями точности прогнозов, получаемых при различных комбинациях весовых функций.

Во многих случаях целесообразно использовать последовательность $\omega_j = \alpha^j$, $\alpha < 1$, $j = 1, 2, \dots$, придающую более высокий вес более поздней информации и позволяющую относительно просто оценивать значения коэффициентов даже достаточно сложных моделей, таких, в которых для описания сезонных циклов используются полиномы в сочетании с преобразованиями Фурье (подобное представление можно рассматривать как сложные полиномы). Для модели экспоненциально взвешенного скользящего среднего предложены способы, с помощью которых в те периоды времени, когда средняя ошибка прогноза близка к нулю (благодаря верно выбранным модели и ее коэффициентов), скорость убывания последовательности α^j может быть увеличена, а в те периоды времени, когда средняя ошибка прогноза значительно больше нуля и

существует опасность того, что модель может «забыть» старую информацию (в этом случае требуется уточнение прогноза), скорость убывания α^j может быть уменьшена.

Уточнение прогноза проводят по принципу обратной связи: новые прогнозы корректируют на основе учета ошибок в предшествующих прогнозах. Если при выборе весовых множителей в процессе составления прогноза также используется обратная связь, то не только строгий анализ областей устойчивости данной системы, но и любой другой анализ становятся фактически невозможными. Для анализа эффективности какого-либо метода недостаточно привести примеры, подтверждающие его полезность. Необходимо также выявить области (если они существуют), в которых применение рассматриваемого метода невозможно или неэффективно.

Многие методы позволяют отыскать наилучшее значение скорости затухания весовых множителей путем многократного анализа имеющегося ряда данных. При этом в качестве критерия используется достигаемая точность прогноза (минимальная дисперсия ошибок). Однако такой подход содержит и недостатки.

Во-первых, если в средней ошибке есть значимые разности (они должны быть равны нулю), то более вероятно, что эти разности больше зависят от способа выбора начальных значений коэффициентов модели, чем от различий в скорости затухания весовых множителей или постоянной сглаживания.

Во-вторых, еще более важный источник возможной ошибки можно проиллюстрировать с помощью следующего примера. Будем рассматривать очень длинный ряд чисел как некоторый коррелированный процесс. Этот ряд, стационарность и однородность которого гарантирована самим способом его получения, разобьем на короткие отрезки, содержащие достаточно данных для отыскания наилучшего значения скорости затухания весовых множителей. Проведем анализ результатов, полученных для каждого из этих отрезков. Несмотря на то, что все отрезки относятся к одному и тому же процессу, тем не менее, существует широкое распределение соответствующих значений скорости затухания. Определить скорость затухания для данного отрезка можно только после того, как он стал предысторией. Величина скорости затухания на следующем отрезке, которая может быть другой, будет известна только после определения ее для данного отрезка, и, следовательно, для прогноза такие значения вообще бесполезны.

Разность между точностью, получаемой при постоянном использовании некоторой стандартной скорости затухания весовых множителей, например при $\alpha = 0,9$, и точностью, которая может быть достигнута при заранее известном оптимальном значении скорости затухания, относительно мала по сравнению с выборочной ошибкой скорости затухания в том случае, когда наилучшее значение весовых множителей для длинного ряда выбирается по данным для коротких отрезков этого ряда.

Рассмотрим метод экспоненциального сглаживания полиномом степени p :

$$y_t = \sum_{i=0}^p \frac{a_i}{i!} t^i,$$

представленного n членами ($t = 1, 2, \dots, n$).

Экспоненциальной средней первого порядка для ряда y_t назовем

$$S_t^{[1]}(y) = \alpha \sum_{i=0}^n (1-\alpha)^i y_{t-i},$$

где $0 < \alpha < 1$ — параметр сглаживания, который выбирается априори.

Экспоненциальная средняя k -го порядка для ряда y_t имеет вид

$$S_t^{[k]}(y) = \alpha \sum_{i=0}^n (1-\alpha)^i S_{t-i}^{[k-1]}(y).$$

Справедлива следующая рекуррентная формула [41]:

$$S_t^{[k]}(y) = \alpha S_t^{[k-1]}(y) + (1-\alpha) S_{t-1}^{[k]}(y),$$

откуда имеем

$$\begin{aligned} S_t^{[1]}(y) &= \alpha y_t + (1-\alpha) S_{t-1}^{[1]}(y) = \alpha y_t + (1-\alpha) [\alpha y_{t-1} + (1-\alpha) S_{t-2}^{[1]}(y)] = \\ &= \alpha y_t + \alpha(1-\alpha) y_{t-1} + (1-\alpha)^2 [\alpha y_{t-2} + (1-\alpha) S_{t-3}^{[1]}(y)] = \\ &= \alpha \sum_{i=0}^{t-1} (1-\alpha)^i y_{t-i} + (1-\alpha)^t y_0. \end{aligned}$$

Веса (коэффициенты при $S_t^{[k]}(y)$), присвоенные наблюдениям, убывают в геометрической прогрессии:

$$S_t^{[2]}(y) = \alpha S_t^{[1]}(y) + (1 - \alpha) S_{t-1}^{[2]}(y),$$

$$\dots\dots\dots$$

$$S_t^{[n]}(y) = \alpha S_t^{[n-1]}(y) + (1 - \alpha) S_{t-1}^{[n]}(y).$$

Прогноз значений динамического ряда y_t в момент $t+l$ после $t=n$ строится с помощью ряда Тейлора:

$$\hat{y}_{t+l} = y_t^{(0)} + l y_t^{(1)} + \dots + \frac{l^p}{p!} y_t^{(p)},$$

где $y_t^{(p)}$ — p -я производная, взятая в момент t .

Для линейной модели $y_t = a_0 + a_1 t$ прогноз вычисляется по формуле $\hat{y}_{t+l} = \hat{a}_0 + l \hat{a}_1$; для квадратичной модели $y_t = a_0 + a_1 t + \frac{1}{2} a_2 t^2$ получим

$$\hat{y}_{t+l} = \hat{a}_0 + l \hat{a}_1 + \frac{1}{2} l^2 \hat{a}_2.$$

Оценки параметров \hat{a}_0 , \hat{a}_1 для линейной модели и \hat{a}_0 , \hat{a}_1 , \hat{a}_2 для квадратичной модели находятся соответственно из систем уравнений, связывающих искомые параметры с экспоненциальными средними.

В первом случае получаем систему уравнений

$$S_t^{[1]}(y) = \hat{a}_0 - \frac{1-\alpha}{\alpha} \hat{a}_1,$$

$$S_t^{[2]}(y) = \hat{a}_0 - \frac{2(1-\alpha)}{\alpha} \hat{a}_1,$$

откуда имеем

$$\hat{a}_0 = 2S_t^{[1]}(y) - S_t^{[2]}(y),$$

$$\hat{a}_1 = \frac{\alpha}{1-\alpha} (S_t^{[1]}(y) - S_t^{[2]}(y)).$$

Ошибка прогноза при этом определяется следующим образом:

$$\sigma(\hat{y}_{t+l}) = \sigma_0 \sqrt{\frac{\alpha}{(1-\alpha)^3} (1 + 4(1-\alpha) + 5(1-\alpha^2) + 2\alpha(4-3\alpha)l + 2\alpha^2 l^2)},$$

где σ_0 — среднее квадратическое отклонение для наблюдаемого ряда \hat{y}_t с n элементами и p параметрами полинома:

$$\sigma_0^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - p - 1}.$$

Во втором случае получаем систему уравнений

$$\begin{aligned} S_t^{[1]}(y) &= \hat{a}_0 - \frac{1-\alpha}{\alpha} \hat{a}_1 + \frac{(1-\alpha)(2-\alpha)}{2\alpha^2} \hat{a}_2, \\ S_t^{[2]}(y) &= \hat{a}_0 - \frac{2(1-\alpha)}{\alpha} \hat{a}_1 + \frac{(1-\alpha)(3-2\alpha)}{2\alpha^2} \hat{a}_2, \\ S_t^{[3]}(y) &= \hat{a}_0 - \frac{3(1-\alpha)}{\alpha} \hat{a}_1 + \frac{3(1-\alpha)(4-3\alpha)}{2\alpha^2} \hat{a}_2, \end{aligned}$$

откуда имеем

$$\begin{aligned} \hat{a}_0 &= 3(S_t^{[1]}(y) - S_t^{[2]}(y)) + S_t^{[3]}(y), \\ \hat{a}_1 &= \frac{\alpha}{2(1-\alpha)^2} ((6-5\alpha)S_t^{[1]}(y) - 2(5-4\alpha)S_t^{[2]}(y) + (4-3\alpha)S_t^{[3]}(y)), \\ \hat{a}_2 &= \frac{\alpha^2}{(1-\alpha)^2} (S_t^{[1]}(y) - 2S_t^{[2]}(y) + S_t^{[3]}(y)). \end{aligned}$$

Ошибка прогноза рассчитывается по формуле

$$\sigma(\hat{y}_{t+l}) \approx \sigma_0 \sqrt{2\alpha + 3\alpha^2 + 3\alpha^3 l^3}.$$

Для применения этих формул необходимо знать начальную величину в рекуррентной формуле $S_{t-1}^{[k]}(y)$. Обычно она определяется из анализа только части динамического ряда. При отсутствии предварительных рекомендаций для нахождения начальных условий можно пользоваться следующими формулами:

$$\begin{aligned} S_{t-1}^{[1]}(y) &= a_0 - \frac{1-\alpha}{\alpha} a_1, \\ S_{t-1}^{[2]}(y) &= a_0 - \frac{2(1-\alpha)}{\alpha} a_1 \end{aligned}$$

для линейной модели,

$$S_{t-1}^{[1]}(y) = a_0 - \frac{1-\alpha}{\alpha} a_1 + \frac{(1-\alpha)(2-\alpha)}{2\alpha^2} a_2,$$

$$S_{t-1}^{[2]}(y) = a_0 - \frac{2(1-\alpha)}{\alpha}a_1 + \frac{(1-\alpha)(3-2\alpha)}{2\alpha^2}a_2,$$

$$S_{t-1}^{[3]}(y) = a_0 - \frac{3(1-\alpha)}{\alpha}a_1 + \frac{3(1-\alpha)(4-3\alpha)}{2\alpha^2}a_2$$

для квадратичной модели.

Значения параметров для этих формул рекомендуется получать методом наименьших квадратов. Выбор начальных условий влияет на оценки погрешностей метода. Другой важной проблемой является выбор параметра сглаживания α . При значениях α , близких к единице, учитываются только последние члены ряда, при α , близких к нулю, — практически все члены ряда. Точных рекомендаций по выбору значений α не существует.

Метод экспоненциального сглаживания разработан для рядов, состоящих из большого числа наблюдений, при увеличении числа наблюдений точность прогноза должна возрастать. При анализе коротких рядов метод не «срабатывает», так как часто не «успевает» учесть изменения при быстрых темпах роста процесса. Если явление протекает в одних и тех же условиях, то точность прогноза определяется величиной периода предыстории явления (базисного периода) и длительностью прогнозируемого периода. Количественно точность прогноза можно оценить по имеющемуся динамическому ряду.

§ 10.5. Многофакторное прогнозирование

Рассмотрим динамическое многофакторное прогнозирование. Многофакторные динамические модели должны учитывать пространственные и временные изменения факторов (аргументов), а также (при необходимости) запаздывание влияния этих факторов на зависимую переменную (функцию). Многофакторное прогнозирование позволяет учитывать развитие взаимосвязанных процессов и явлений. Основой его является системный подход к изучению исследуемого явления, а также процесс осмысления явления как в прошлом, так и в будущем. В многофакторном прогнозировании одной из основных проблем является проблема выбора факторов, обуславливающих поведение системы, которая не может быть решена чисто статистическим путем, а только при помощи детального изучения существа явления. Здесь следует подчеркнуть преимуще-

ство анализа (осмысления) перед использованием чисто статистических (математических) методов изучения явления.

В традиционных методах (например, в методе наименьших квадратов) считается, что наблюдения независимы друг от друга (по одному и тому же аргументу). В действительности существует автокорреляция и, если ее не учитывать, это приводит к неоптимальности статистических оценок, затрудняет построение доверительных интервалов для коэффициентов регрессии, а также проверку их значимости.

Автокорреляция определяется по отклонениям от трендов. Она может иметь место, если не учтено влияние существенного фактора или нескольких менее существенных факторов, действия которых направлены «в одну сторону», либо если неверно выбрана модель, устанавливающая связь между факторами (аргументами) и функцией. Для выявления наличия автокорреляции применяется критерий Дурбина — Уотсона

$$d = \frac{\sum_{i=1}^n (\varepsilon_{i+1} - \varepsilon_i)^2}{\sum_{i=1}^n \varepsilon_i^2},$$

где ε_i — случайные отклонения от регрессионной модели.

Для исключения или уменьшения автокорреляции применяется переход к случайной компоненте (исключение тренда) или введение времени в уравнение множественной регрессии в качестве аргумента.

В многофакторных моделях *возникает проблема мультиколлинеарности* — наличие сильной корреляции между факторами, которая может существовать вне всякой зависимости между функцией и факторами.

При мультиколлинеарности между аргументами существует (линейная) связь и наряду с уравнением регрессии имеются и другие (линейные) отношения, искажается физический смысл коэффициентов регрессии, появляется некорректность — слабая обусловленность систем алгебраических уравнений. Соответствующие диагональные элементы матрицы, обратной матрице вторых производных от функции правдоподобия, определяющие дисперсии оценок коэффициентов регрессии, стремятся к бесконечности и порождают ошибочную (в данном случае) тенденцию к исключению существенных переменных из уравнения регрессии. Муль-

тиколлинеарность определяется как степень отклонения независимых переменных (факторов) от ортогональных переменных. Для проверки наличия мультиколлинеарности во множестве независимых переменных используется показатель

$$\chi^2 = -\left(n-1 - \frac{1}{6}(2p+5)\right) \ln(X^T X),$$

где X — матрица факторов (аргументов), p — число факторов, n — число наблюдений. Показатель χ^2 приближенно имеет χ^2 -распределение.

Главное в процедуре прогнозирования состоит в последующем выявлении тех факторов, которые наиболее сильно взаимосвязаны. Это можно сделать по диагональным элементам матрицы, обратной матрице вторых производных от функции правдоподобия.

Вводится величина

$$\Phi_i = (c_{ii} - 1) \left(\frac{n-p}{p-1} \right),$$

где c_{ii} — i -й диагональный элемент матрицы. Величина Φ_i имеет F -распределение с $n-p$ и $p-1$ степенями свободы.

Выявив, какие факторы являются мультиколлинеарными, можно определить характер взаимосвязанности между мультиколлинеарными элементами множества независимых переменных. Для этой цели используются недиагональные элементы обратной матрицы и полученные в них коэффициенты частной корреляции между мультиколлинеарными переменными и всеми остальными переменными. Анализ проводится по t -критерию Стьюдента для коэффициента частной корреляции.

В многофакторном прогнозировании без методов конфлюэнтного анализа (без учета погрешностей аргументов) нельзя обойтись в принципе. Так, в многофакторном анализе необходимо наряду с оценкой параметров сглаживающей (исследуемой) функции построить прогноз каждого фактора (по неким другим функциям или моделям). Естественно, что значения факторов, полученные в эксперименте в базисном периоде, не совпадают с аналогичными значениями, найденными по прогнозирующим моделям для факторов. Поэтому либо это различие должно быть объяснено случайными отклонениями, величина которых выявлена указанными различиями и должна быть учтена сразу же при оценке параметров сглаживающей функции, либо это различие не случайно

и никакого прогноза делать нельзя, т. е. в задаче многофакторного прогнозирования исходные значения факторов, как и значения сглаживающей функции, должны быть взяты с соответствующими ошибками, закон распределения которых должен быть определен при соответствующем анализе, предшествующем процедуре прогнозирования.

Пример (построение многофакторного прогноза). Рассмотрим задачу построения прогноза значений показателя α использования производственной мощности отрасли связи в 1986–1990 гг. как функцию факторов d_a , u , d_{ab} . В качестве базового периода был взят период 1977–1985 гг. (табл. 10.1).

Таблица 10.1

Исходные данные для определения зависимости α от d_a , u , d_{ab}

α , $\frac{\text{тысяч разговоров}}{\text{канал}}$	d_a	u	d_{ab}
6,72	0,112	5,4	0,431
6,97	0,148	4,8	0,484
7,17	0,190	6,3	0,552
7,37	0,234	6,8	0,593
7,73	0,273	7,2	0,630
7,99	0,312	7,6	0,649
8,16	0,347	8,1	0,663
8,50	0,361	8,5	0,687
8,23	0,423	8,9	0,724

Погрешность исходных данных принималась равной 5 единицам первого не указанного разряда после запятой.

Первоначально для каждого фактора необходимо было выбрать модель, по которой определяют «истинные» значения факторов в базовом периоде (табл. 10.2) и которую используют в дальнейшем для построения прогноза. Отбор наиболее приемлемых моделей проводился по χ^2 -критерию Пирсона.

Таблица 10.2

«Истинные» значения факторов

Параметр	Годы								
	1977	1978	1979	1980	1981	1982	1983	1984	1985
\hat{d}_a	0,1121	0,1479	0,1903	0,2346	0,2730	0,3103	0,3464	0,3610	0,4248
\hat{u}	5,3811	5,8120	6,2622	6,7592	7,200	7,6456	8,1396	8,5017	8,8466
\hat{d}_{ab}	0,4537	0,5028	0,5460	0,5840	0,6175	0,6470	0,6730	0,6959	0,7161

В процессе анализа для прогноза показателя α была выбрана функция

$$\alpha = a_1 + a_2 d_{a6} + a_3 u + a_4 d_a + a_5 \frac{d_a}{u}.$$

Оценки параметров вектора a и их дисперсий получены по алгоритмам, описанным в § 7.12, и были следующими: $\hat{a}_1 = 0,067$, $\hat{a}_2 = 0,515$, $\hat{a}_3 = 1,411$, $\hat{a}_4 = -11,719$, $\hat{a}_5 = 7,513$, $D(\hat{a}_1) = 0,010$, $D(\hat{a}_2) = 0,018$, $D(\hat{a}_3) = 0,0003$, $D(\hat{a}_4) = 0,085$, $D(\hat{a}_5) = 0,008$.

Таблица 10.3

Прогнозируемые значения $\hat{\alpha}$ и $\sigma(\hat{\alpha})$

Параметр	Годы				
	1986	1987	1988	1989	1990
$\hat{\alpha}, \frac{\text{тысяч разговоров}}{\text{канал}}$	10,64	11,01	11,22	11,66	12,31
$\sigma(\hat{\alpha})$	0,29	0,31	0,32	0,33	0,34

Значения $\hat{\alpha}$ в базовом периоде при найденных оценках вектора a не отличались от заданных значений α больше, чем на 0,9 %; величина χ^2 была равна 0,0034; вероятность того, что модель не может быть отвергнута, была равна 0,999. Прогнозируемые значения $\hat{\alpha}$ и их средние квадратические отклонения за период 1986–1990 гг. приведены в табл. 10.3.

§ 10.6. Идентификация моделей типа АРПСС

Приступим к процедуре идентификации модели, записанную в конечных разностях. Рассмотрим общую разностную модель — модель АРПСС. Конкретная цель состоит в том, чтобы получить некоторое указание на то, какие значения p , d и q нужны в общей линейной модели АРПСС, и выбрать некоторые начальные значения параметров этой модели. Полученная таким образом пробная модель является отправной точкой для применения более формальных и эффективных методов оценивания, описанных в гл. 7.

Идентификация и оценивание взаимосвязаны. Идентификация неизбежно не точна. Она неточна потому, что возникновение вопроса о том, какие типы моделей встречаются на практике и в каких обстоятельствах, — это следствие поведения объектов физического мира, и проблема идентификации не может быть решена

только математическими методами. Поскольку на этапе идентификации нельзя точно сформулировать задачу, приходится использовать статистически «неэффективные» методы. На этом этапе особенно полезны графические методы. Предварительная идентификация не требует ничего, кроме анализа класса моделей, которые будут позднее эффективно подгоняться к результатам наблюдений и проверяться.

Например, идентифицировать подходящий класс моделей из общего семейства моделей АРСС

$$\Phi(B)\Delta^d y_t = \theta_0 + \theta(B)a_t,$$

который применен для описания данного ряда, можно следующим образом:

а) вычислим конечную разность от y_t столько раз, сколько необходимо, чтобы обеспечить стационарность модели, и сведем изучаемый процесс к смешанному процессу авторегрессии — скользящего среднего

$$\Phi(B)\omega_t = \theta_0 + \theta(B)a_t,$$

где $\omega_t = (1 - B)^d y_t = \Delta^d y_t$;

б) идентифицируем конечный ряд с помощью автокорреляционной и частной автокорреляционной функций; эти функции используют также для нахождения приближенных оценок параметров, чтобы получить начальные значения параметров для этапа оценивания итеративных процедур.

В процессе идентификации прежде всего надо оценить порядок разности d .

Автокорреляционная функция стационарного смешанного процесса авторегрессии — скользящего среднего удовлетворяет разностному уравнению

$$\Phi(B)\rho_k = 0, \quad k > q,$$

где ρ_k — автокорреляция (автокорреляционная функция) с задержкой k , которая для вариационного ряда z_1, z_2, \dots, z_n определяется по формуле

$$\rho_k = \frac{M[(z_t - \mu)(z_{t+k} - \mu)]}{\sqrt{M[(z_t - \mu)^2]M[(z_{t+k} - \mu)^2]}} = \frac{M[(z_t - \mu)(z_{t+k} - \mu)]}{\sigma_z^2} = \frac{\gamma_k}{\gamma_0}.$$

Функция γ_k от задержки k называется *автоковариационной функцией стохастического процесса*; функция ρ_k от задержки k называется *автокорреляционной функцией стохастического процесса*. На практике определяют только выборочные оценки этих функций. Например, оценку $\hat{\gamma}_k$ определяют по формуле

$$\hat{\gamma}_k = \frac{1}{n} \sum_{t=1}^{n-k} (z_t - \bar{z})(z_{t+k} - \bar{z}), \quad k = 0, 1, \dots, m,$$

где \bar{z} — среднее значение ряда наблюдений, n — число наблюдений.

Дисперсия оценки выборочного коэффициента автокорреляции имеет вид

$$D(\hat{\rho}_k) \approx \frac{1}{n} \sum_{v=-\infty}^{+\infty} (\rho_v^2 + \rho_{v+k}\rho_{v-k} - 4\rho_k\rho_v\rho_{v-k} + 2\rho_v^2\rho_k^2).$$

Если $\hat{\rho}_k = C^{|k|}$, $-1 < C < 1$, т. е. функция $\hat{\rho}_k$ затухает экспоненциально, то дисперсия оценки $\hat{\rho}_k$ определяется по формуле

$$D(\hat{\rho}_k) \approx \frac{1}{n} \left[\frac{(1+C^2)(1-C^{2k})}{1-C^2} - 2kC^{2k} \right].$$

Автокорреляционная функция стационарного смешанного процесса авторегрессии — скользящего среднего удовлетворяет разностному уравнению $\Phi(B)\rho_k = 0$.

Кроме того, если $\Phi(B) = \prod_{i=1}^p (1 - G_i B)$, то решение этого разностного уравнения для k -й автокорреляции ρ_k в предположении отсутствия кратности корней имеет вид

$$\rho_k = A_1 G_1^k + A_2 G_2^k + \dots + A_p G_p^k, \quad k > q - p.$$

Условие стационарности требует, чтобы нули $\Phi(B)$ лежали вне единичного круга, и приводит к тому, что корни G_1, G_2, \dots, G_p лежат внутри единичного круга. Для стационарной модели, ни один из корней которой не лежит близко к границе единичного круга, автокорреляционная функция быстро затухает при средних и больших k . Если хотя бы один действительный корень, например G_1 , приближается к 1, так что $G_1 = 1 - \delta$, где δ — малое положительное число, то, поскольку $\rho_k \approx A_1(1 - k\delta)$ для больших k ,

автокорреляционная функция не будет быстро затухать, а будет убывать медленно и почти линейно. Следовательно, отсутствие тенденции к затуханию автокорреляционной функции может рассматриваться как свидетельство того, что существует корень, близкий к 1. Выборочная автокорреляционная функция похожа на теоретическую. Отсюда следует, что отсутствие затухания выборочной автокорреляционной функции логично истолковать в том смысле, что процесс y_t является нестационарным, хотя, возможно, его разность Δy_t или какая-либо разность более высокого порядка стационарна. Предполагается, что необходимая для получения стационарности степень разности d достигнута, если автокорреляционная функция ряда $\omega_t = \Delta^d y_t$ быстро затухает. На практике d обычно равно 0, 1 или 2. Для определения d достаточно просмотреть примерно 20 первых значений автокорреляции исходного ряда и рядов его первых и вторых разностей. Приняв предварительно, что разность имеет степень d , по общему виду выборочных автокорреляционной и частной автокорреляционной функций соответствующего разностного ряда получаем указания к выбору порядков p и q операторов авторегрессии и скользящего среднего, т. е. проводим идентификацию результирующего стационарного процесса АРСС.

В то время как автокорреляционная функция процесса авторегрессии порядка p убывает плавно, ее частная автокорреляционная функция имеет обрыв после p -й задержки. Однако автокорреляционная функция процесса скользящего среднего порядка q обрывается после задержки q , в то время как ее частная автокорреляция плавно убывает с ростом задержки.

Автокорреляционная функция смешанного процесса, содержащая компоненту авторегрессии порядка p и компоненту скользящего среднего порядка q , после первых $q - p$ задержек представляется в виде суммы экспонент и затухающих синусоид. В то же время частная автокорреляционная функция смешанного процесса приближенно представляется суммой экспонент и затухающих синусоид после $p - q$ задержек.

Выборочные автокорреляции могут иметь большие значения дисперсии и быть сильно коррелированы друг с другом, и нельзя ожидать детального сходства выборочной автокорреляционной функции с теоретической. Выборочные функции могут иметь

всплески и тренды, которые теоретические функции не имеют. Поэтому может понадобиться исследование на этапах оценивания и диагностической проверки двух или более возможных моделей процесса. Важно знать, какие отличия выборочной автокорреляционной функции допустимы. В частности, необходимо уметь оценить, становятся ли автокорреляции и частные автокорреляции практически нулевыми при задержках, больших некоторого k . Для больших задержек можно вычислить стандартные ошибки оцениваемых автокорреляций по формуле

$$\sigma[\rho_k] \approx \frac{1}{n^{1/2}} [1 + 2(\rho_1^2 + \rho_2^2 + \dots + \rho_q^2)]^{1/2}, \quad k > q,$$

в которой ρ_k — выборочная оценка k -й автокорреляции. Стандартная ошибка частной автокорреляции порядка не меньше $p + 1$ вычисляется по формуле

$$\sigma[\hat{\Phi}_{kk}] \approx \frac{1}{n^{1/2}}, \quad k > p,$$

где $\hat{\Phi}_{kk}$ — последний коэффициент процесса авторегрессии порядка p .

Даже для не слишком больших n выборочный коэффициент автокорреляции распределен примерно по нормальному закону со средним значением, равным нулю. При гипотезе, что автокорреляция равна нулю, ее оценка ρ_k , деленная на стандартную ошибку, будет распределена примерно по нормальному закону с единичной дисперсией. То же справедливо и для частных автокорреляций. Эти факты можно использовать для получения нестрогих правил проверки, являются ли теоретические автокорреляции и частные автокорреляции при задержках, больших некоторой, практически нулевыми.

Необходимо иметь в виду следующее:

- 1) по рядам ограниченной длины никогда не удастся доказать, что корень оператора авторегрессии, определяющий скорость стремления автокорреляций к нулю, точно равен единице;
- 2) не существует, конечно, резкого перехода от стационарного поведения ряда к нестационарному.

В ряде случаев некоторыми преимуществами может обладать использование нестационарных моделей в задачах прогнозирования. Когда коэффициент процесса авторегрессии Φ_1 близок к еди-

нице, мы не знаем, имеет ли среднее значение ряда какой-либо смысл или не имеет. Если использовать нестационарную модель, которая не содержит среднего значения μ , то прогнозы будущего поведения не будут зависеть от выборочного среднего, найденного по предыдущему периоду, которое может не иметь никакого отношения к будущему уровню ряда.

§ 10.7. Методы уточнения прогнозов по модели АРПСС

Разные методы прогнозирования должны приводить практически к одному результату — выбору одной и той же модели из возможных — и одинаковой оценке последствий выбранного варианта. Правильное ведение прогнозирования предполагает, в частности, применение соответствующих критериев при пересмотре прогнозов.

Для уточнения прогноза применяется метод адаптивного сглаживания, основы которого были заложены при разработке метода экспоненциального сглаживания и затем развиты путем включения в рассмотрение сезонных моделей. Этот метод целесообразно использовать в тех случаях, когда явления достаточно хорошо описываются с помощью постоянного уровня (константы), тренда и сезонных моделей, которые не претерпевают резких изменений для большинства рядов. Во многих случаях адаптивное сглаживание позволяет получать хорошие результаты в течение короткого промежутка времени, что иногда имеет более важное значение, чем получение более точного прогноза в течение длительного времени, но с применением значительных усилий (больших затрат).

Если прогнозы составляются для переменных интенсивности и результаты наблюдений получены в нерегулярные интервалы времени, то исходные данные преобразуются к величинам, характеризующим скорость их изменения на равномерных интервалах времени. Получаемые в результате прогнозы могут быть приведены в соответствие с величинами будущих интервалов времени. Заметим, что подобное использование сечения какой-либо входной переменной одинаково пригодно для любых рядов и не приводит к увеличению дисперсии, какое имело бы место в случае получения сечения из данных предыстории.

Рассмотрим получение прогноза для модели АРПСС. Для того чтобы в полной мере использовать достоинства метода, необходимо иметь достаточно длинные ряды (содержащие не менее 100 резуль-

татов наблюдений); это в значительной степени помогает понять изучаемый процесс с точки зрения получения предварительных данных относительно возможных изменений процесса в будущем. С помощью данного метода может быть проанализирован гораздо более широкий круг моделей, чем с помощью метода адаптивного сглаживания. Кроме того, он особенно полезен в тех случаях, когда требуется получить как можно более точный прогноз.

Анализ моделей начинается с формирования разностей из исходных данных y_t и вычисления частных автокорреляционных функций ϕ для разностных рядов $\omega_t = \Delta^d y_t$. Затем разностные ряды вводятся в модель с соответствующим числом коэффициентов авторегрессии и скользящего среднего:

$$\omega_t = \phi_1 \omega_{t-1} + \dots + \phi_p \omega_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}.$$

Для оценки коэффициентов модели такого типа и для вычисления дисперсии шума в результатах наблюдений используется метод максимума правдоподобия, т. е., как уже указывалось, операторы разности скользящего среднего могут быть разложены таким образом, чтобы выразить модель прогнозов через исходные данные наблюдений:

$$y_t = \Phi_1 y_{t-1} + \dots + \Phi_p y_{t-p} + \theta_0 + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}.$$

Сначала прогнозируется изменение преобразованных входных переменных, а затем осуществляется обратное преобразование.

Если необходимо проанализировать сезонные циклы, то вклад от каждого из них в прогноз вычисляется путем составления разностей для определенных отрезков цикла. Для годового цикла, например, в качестве таких отрезков можно использовать месяцы, и тогда прогноз для каких-то месяцев в будущем будет основываться на известных данных для этих месяцев в прошлые годы.

Приведем основные формулы для получения прогнозов, вероятностных пределов для прогнозов и для коррекции прогнозов. Пусть y_t — отклонение наблюдаемого ряда от любой известной детерминированной функции $f(t)$. В частности, для стационарного ряда функция $f(t)$ может быть равна среднему значению μ ряда или нулю. В последнем случае y_t образуют наблюдаемый ряд. Рассмотрим общую модель АРПСС

$$\Phi(B)\Delta^d y_t = \theta(B)a_t$$

или

$$\varphi(B)y_t = \theta(B)a_t.$$

Пусть известны значения наблюдаемого ряда до момента t . Тогда прогноз $\hat{y}_t(l)$, $l > 0$, с минимальной средней квадратической ошибкой — это условное математическое ожидание величины y_{t+l} при заданных значениях y_t, y_{t-1}, \dots :

$$\hat{y}_t(l) \equiv [y_{t+l}] = M[y_{t+l} | y_t, y_{t-1}, \dots].$$

Отсюда следует, что ошибки прогноза с упреждением, равным единице (на шаг вперед), — это некоррелированные между собой импульсы, генерируемые моделью.

На практике простейший способ вычисления прогнозов — непосредственное использование разностного уравнения, составленного из условных математических ожиданий:

$$\begin{aligned} \hat{y}_t(l) = & \varphi_1[y_{t+l-1}] + \dots + \varphi_{p+d}[y_{t+l-p-d}] + \\ & + [a_{t+l}] - \theta_1[a_{t+l-1}] - \dots - \theta_q[a_{t+l-q}]. \end{aligned} \quad (10.1)$$

Для вычисления условных математических ожиданий вместо значений y , известных к этому моменту, подставляются их реальные значения, вместо будущих значений y — их прогноз, вместо известных a — их реальные значения и вместо будущих a — нули. Процесс прогнозирования может быть начат аппроксимацией неизвестных a нулями.

Пусть требуется найти прогнозы с упреждением $1, 2, \dots, l$ и их доверительные пределы как взвешенную сумму текущего и предшествующих импульсов a_j [6]:

$$y_{t+l} = \sum_{j=-\infty}^{t+l} \psi_{t+l-j} a_j = \sum_{j=0}^{\infty} \psi_j a_{t+l-j},$$

где $\psi_0 = 1$ и остальные координаты (веса) ψ_j оператора ψ находят из уравнения

$$\varphi(B)\psi(B) = \theta(B),$$

приравнивая коэффициенты при одинаковых степенях B в выражении

$$\begin{aligned} (1 - \varphi_1 B - \dots - \varphi_{p+d} B^{p+d})(1 + \psi_1 B + \psi_2 B^2 + \dots + \psi_k B^k) = \\ = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q. \end{aligned}$$

Зная значения φ и θ , после оценки их по результатам наблюдений используем следующие выражения для $\psi = (\psi_1, \dots, \psi_j)$:

$$\psi_1 = \varphi_1 - \theta_1,$$

$$\psi_2 = \varphi_1 \psi_1 + \varphi_2 - \theta_2,$$

$$\dots\dots\dots$$

$$\psi_j = \varphi_1 \psi_{j-1} + \dots + \varphi_{p+d} \psi_{j-p-d} - \theta_j,$$

где $\psi_0 = 1$, $\psi_j = 0$ при $j < 0$ и $\theta_j = 0$ при $j > q$. Если k — наибольшее из целых чисел $p + d - 1$ и q , то при $j > k$ координаты вектора ψ удовлетворяют разностному уравнению

$$\psi_j = \varphi_1 \psi_{j-1} + \varphi_2 \psi_{j-2} + \dots + \varphi_{p+d} \psi_{j-p-d}.$$

Следовательно, координаты вектора ψ вычисляются рекуррентным способом. Подставляем вес ψ в y_{t+l} и получаем значение прогноза для каждого нужного уровня значимости ε и для каждого упреждения l :

$$y_{t+l} = \hat{y}_t(l) \pm u_{\varepsilon/2} \left(1 + \sum_{j=1}^{l-1} \psi_j^2 \right)^{1/2} \sigma(a).$$

Здесь среднее квадратическое отклонение $\sigma(a)$ можно заменить практически его оценкой $s(a)$ — выборочным стандартным отклонением белого шума a_t ; $u_{\varepsilon/2}$ — квантиль уровня $1 - \varepsilon/2$ стандартного нормального распределения.

Дисперсия ошибки прогноза на l шагов вперед для любого момента t , определяемая как математическое ожидание величины $e_t^2(l) = (y_{t+l} - \hat{y}_t(l))^2$, имеет вид

$$D(e) = \left(1 + \sum_{j=1}^{l-1} \psi_j^2 \right) \sigma^2(a).$$

Из предположения, что a подчиняется нормальному закону распределения, следует, что при известных значениях процесса до момента t условное распределение вероятности $P(y_{t+l} | y_t, y_{t-1}, \dots)$ будущего значения процесса y_{t+l} будет также нормальным со средним значением $\hat{y}_t(l)$ и стандартным отклонением

$$\sqrt{D(e)} = \left(1 + \sum_{j=1}^{l-1} \psi_j^2 \right)^{1/2} \sigma(a).$$

Можно выразить прогнозы $\hat{y}_{t+1}(l)$ и $\hat{y}_t(l+1)$ будущего наблюдения y_{t+l+1} , сделанные в моменты $t+1$ и t :

$$\hat{y}_{t+1}(l) = \psi_l a_{t+1} + \psi_{l+1} a_t + \psi_{l+2} a_{t-1} + \dots,$$

$$\hat{y}_t(l+1) = \psi_{l+1} a_t + \psi_{l+2} a_{t-1} + \dots$$

Вычитая второе уравнение из первого, получаем

$$\hat{y}_{t+1}(l) = \hat{y}_t(l+1) + \psi_l a_{t+1}. \quad (10.2)$$

Отсюда видно, что если прогноз величины y_{t+l+1} , сделанный в момент t , подправить добавлением ошибки прогноза на шаг вперед a_{t+1} с коэффициентом ψ_l , то получится прогноз той же величины y_{t+l+1} , но сделанный в момент $t+1$.

Положим, что делаются прогнозы в момент t с упреждениями $1, 2, \dots, l$. Затем, как только становится известным y_{t+1} , можно найти $a_{t+1} = y_{t+1} - \hat{y}_t(1)$ и пропорционально подправить прогноз $\hat{y}_{t+1}(l) = \hat{y}_t(l+1) + \psi_l a_{t+1}$ в момент $t+1$ для упреждений $1, 2, \dots, l-1$. Новый прогноз $\hat{y}_t(l+1)$ для упреждения l этим способом найти нельзя, но он легко вычисляется по прогнозам с меньшими упреждениями из разностного уравнения.

Когда становится известным новое отклонение y_{t+1} , прогноз для момента $t+1$ может быть скорректирован. Для этого необходимо вычислить новую ошибку прогноза $a_{t+1} = y_{t+1} - \hat{y}_t(1)$ и использовать разностное уравнение (10.1), где t заменено на $t+1$. Другой возможный метод корректировки — использовать прогнозы $\hat{y}_t(1), \hat{y}_t(2), \dots, \hat{y}_t(l)$ на момент t , получить первые $l-1$ прогнозов $\hat{y}_{t+1}(1), \hat{y}_{t+1}(2), \dots, \hat{y}_{t+1}(l-1)$ на момент $t+1$ из формулы (10.2) и затем найти последний прогноз $\hat{y}_{t+1}(l)$, используя разностное уравнение (10.1).

Приведем другие способы представления прогнозов.

1. Прогнозы в проинтегрированной форме. Для $l > q - p - d$ прогнозы задаются одной функцией

$$\hat{y}_t(l) = b_0^{(t)} y_0(l) + b_1^{(t)} y_1(l) + \dots + b_{p+d-1}^{(t)} y_{p+d-1}(l), \quad (10.3)$$

определяемой «опорными» значениями ряда $\hat{y}_t(q), \hat{y}_t(q-1), \dots, \hat{y}_t(q-p-d+1)$, где $\hat{y}_t(-j) = y_{t-j}$, $j = 0, 1, 2, \dots$. Если $q > p + d$, первые $q - p - d$ прогнозов не описываются этой функцией. В общем случае стационарный оператор авторегрессии приводит к

появлению затухающих экспоненциальных и синусоидальных членов, а нестационарный оператор Δ^d — к появлению полиномиальных членов.

Подстраивающиеся коэффициенты $b_j^{(t)}$ в (10.3) при переходе от момента t к моменту $t+1$ могут быть скорректированы на величину, зависящую от последней ошибки прогноза на шаг вперед a_{t+1} согласно общей формуле

$$b^{(t+1)} = L^T b^{(t)} + g a_{t+1},$$

где $L = F_l^{-1} F_{l+1}$, $g = F_l^{-1} \psi(l)$, $b = (b_0^{(t)}, b_1^{(t)}, \dots, b_{p+d-1}^{(t)})^T$, $\psi(l) = (\psi_l, \psi_{l+1}, \dots, \psi_{l+p+d})^T$.

2. Прогноз как взвешенная сумма прошлых наблюдений. С теоретической точки зрения полезно представить прогноз как взвешенную сумму прошлых наблюдений с весами π . Так, если модель представлена в обращенной форме:

$$a_t = \pi(B) y_t = (1 - \pi_1 B - \pi_2 B^2 - \dots) y_t,$$

где веса π_j можно получить из выражения

$$\varphi(B) = (1 - \pi_1 B - \pi_2 B^2 - \dots) \theta(B),$$

приравнявая коэффициенты при одинаковых степенях B , то прогноз с упреждением 1 имеет вид

$$\hat{y}_t(1) = \pi_1 y_t + \pi_2 y_{t-1} + \dots,$$

а прогнозы для бóльших упреждений можно получить из выражения

$$\hat{y}_t(l) = \pi_1 [y_{t+l-1}] + \pi_2 [y_{t+l-2}] + \dots,$$

где условные математические ожидания известных значений y необходимо заменить их реальными значениями, а будущие значения y — их прогнозом.

Прогноз для любого упреждения можно записать как линейную функцию имеющихся наблюдений, т. е.

$$\hat{y}_t(l) = \sum_{j=1}^{\infty} \pi_j^{(l)} y_{t+j-1},$$

где $\pi_j^{(l)}$ — функции весов π_j .

§ 10.8. Байесовские прогнозы

При байесовском подходе каждому наблюдению до его проведения ставится в соответствие ряд первичных вероятностных значений (априорных вероятностей) коэффициентов модели. Построение модели можно начинать из состояния полной неопределенности, так как даже в этом случае метод позволяет довольно точно устанавливать вероятностные значения коэффициентов.

После того как получены результаты наблюдений y , по правилу Байеса определяют апостериорные вероятности, основываясь на которых, вычисляют распределение вероятностей прогнозируемой величины y_l .

Апостериорная плотность вероятности события, состоящего в том, чтобы получить прогнозируемое значение y_l при заданной априорной плотности распределения вероятностей $p(y_l)$ и условной плотности распределения вероятностей $p(y|y_l)$, определяется по формуле

$$p(y_l | y) = \frac{p(y_l)p(y|y_l)}{\int p(y_l)p(y|y_l)dy_l}.$$

Такой подход позволяет игнорировать кратковременные изменения прогнозируемой переменной и в то же время четко регистрировать изменения, происходящие в основном процессе, которые на графике временной зависимости изображаются в виде ступеней и кривых, имеющих соответствующий наклон. Многократное повторение приведенной схемы вычислений дает возможность уменьшить число вероятностных распределений до числа состояний изучаемого процесса. В число состояний обычно входят нейтральное состояние, которое характеризуется неизменными значениями всех элементов процесса, и состояния, которые соответствуют различным возможным значениям каждого коэффициента и помех. В этих состояниях дисперсия распределения намного выше, чем в нейтральном состоянии, что указывает на отклонение того или иного коэффициента модели от нормы. Возможность уменьшения количества состояний существенна, так как в противном случае число распределений возросло бы пропорционально квадрату числа наблюдений.

Объем вычислений при использовании рассматриваемого метода гораздо больше, чем при адаптивном сглаживании, но здесь

успешное прогнозирование гарантированно и в тех случаях, когда могут возникнуть непредвиденные и существенные изменения в прогнозируемом процессе. Если мы имеем предварительные сведения об ожидаемых событиях, то можно начинать прогнозирование при небольшом массиве данных или вообще при их отсутствии. Вместе с тем, если имеется большой объем результатов наблюдений, прогнозирование можно начинать при отсутствии исходных предпосылок о поведении прогнозируемой величины. Алгоритм позволяет построить модель процесса по данным его предыстории.

Результаты текущих наблюдений представляются выражением $d_t = \mu_t + \varepsilon_t$, где μ_t — уровень полезной составляющей и ε_t — шум текущего наблюдения. Текущий уровень связан с трендом β_t выражением $\mu_t = \mu_{t-1} + \beta_t + \gamma_t$, где тренд изменяется по закону $\beta_t = \beta_{t-1} + \delta_t$. Будем учитывать возможность скачкообразного изменения основного уровня γ_t , изменение тангенса угла наклона δ_t кривой и изменение шума ε_t . Предполагается, что скачкообразное изменение γ_t основного уровня, изменение тангенса угла наклона кривой δ_t , а также изменение шума ε_t независимы и распределены по нормальным законам с нулевым средним и известными (но не обязательно постоянными) дисперсиями D_ε , D_γ и D_δ соответственно.

В модели, которая описывает j состояний, $j \gg 1$, обычно имеются два состояния для каждого возможного возмущения, причем каждое из них характеризуется своим значением дисперсии D_j . В устойчивом состоянии дисперсия мала, а в состоянии изменения велика. Имеются также априорные вероятности π_j того, что состояние j не изменяется. Ниже приведен пример четырех состояний и соответствующих им условных вероятностей и дисперсий, использованных авторами метода в качестве типичных начальных условий. Дисперсия шума D_0 зависит от соотношения между фактическими изменениями процесса и текущим прогнозом и описывается соотношением $R_j = D_j / D_0$ (табл. 10.4).

Пусть вектор ϕ содержит следующие элементы (параметры): ожидаемое значение уровня, ожидаемое значение тангенса угла наклона, дисперсию ошибок при определении уровня, ковариацию наклона и уровня, а также дисперсию ошибок при определении тангенса угла наклона. Тогда если в момент времени $t-1$ совместное распределение уровня и тангенса угла наклона является

двумерным нормальным и характеризуется параметрами Φ_{t-1} , то соответствующее распределение в момент времени t также будет двумерным нормальным, а его параметры будут определяться вектором Φ_t .

Таблица 10.4

Начальные условия для байесовского прогноза

Характеристика состояния	j	π_j	R_ε	R_γ	R_δ
Стабильное	1	0,900	1	0	0
Скачкообразное изменение уровня выходной переменной	2	0,003	1	100	0
Постепенное изменение тангенса угла наклона	3	0,003	1	0	1
Переходное	4	0,094	101	0	0

Существуют также выражения для коэффициентов сезонных сечений при анализе сезонных рядов методом Байеса [37].

§ 10.9. Анализ сезонных рядов

К сезонным относят такие явления, в развитии которых обнаруживаются закономерности, более или менее регулярно повторяющиеся в определенные промежутки времени. В анализе сезонных явлений ставятся следующие задачи: численно выразить проявление сезонных колебаний, а также установить факторы, вызывающие сезонные колебания, дать прогноз сезонных колебаний. Сезонные ряды можно разбить на составляющие: тенденции (тренд), сезонные волны (кратковременные колебания) и помехи (случайные колебания).

Тренд отражает общее изменение ряда за длительный промежуток времени (постоянное увеличение или постоянное уменьшение числового ряда). Сезонные волны — это более или менее регулярные изменения временного ряда, повторяющиеся через определенные промежутки времени. Сезонные колебания обычно имеют постоянный период. Случайные колебания вызываются внешними случайными причинами. Они искажают тренд, а также сезонные и циклические колебания. Выявив закономерные изменения динамики составляющих временного ряда, можно использовать их для экстраполяции значений ряда в будущем.

Методы нахождения основной тенденции развития явления (тренда) идентичны рассмотренным: методу наименьших квадратов, методу скользящих средних и т. д. В последнем случае интервал сглаживания определяется периодом сезонных колебаний.

Для описания сезонных моделей существуют два разных подхода, каждый из которых имеет множество вариантов.

Если дано N результатов наблюдений за какой-то период времени, то можно записать N поправочных членов (положительных и отрицательных) или N коэффициентов (больших или меньших единицы), которые затем либо суммируются с результатами прогноза, либо умножаются на него. Значения этих поправочных коэффициентов можно определить по результатам наблюдений в соответствующие моменты времени в прошлых циклах.

Весь ряд поправочных членов или коэффициентов может быть назван *сезонным сечением*. Во многих случаях оказывается, что существует небольшое семейство сечений, и тогда прогнозы для отдельных временных рядов могут быть получены путем выбора наиболее подходящего представления этого семейства.

Сечения играют важную роль в методике прогнозирования, так как их сущность легко объяснить, а эффективность их использования легко оценить на практике. Однако существенным недостатком применения сечений является связанная с ними нестабильность прогнозов. Если случайная величина x имеет дисперсию $\sigma^2(x)$, то дисперсия обратной величины ($y = 1/x$) будет порядка $\sigma^4(x)$. В большинстве методов, использующих сезонные или циклические сечения, в процессе проведения вычислений необходимо деление значений членов ряда на случайную переменную (или ее оценку, имеющую отличную от нуля дисперсию). Если при этом делитель имеет тот же порядок величины, что и величина помехи, то относительно малые колебания исходных данных могут привести к большим изменениям прогноза. Подобная нестабильность обычно проявляется как раз тогда, когда принимаемые решения могут иметь самые серьезные последствия (т. е. перед началом следующего цикла). Для уменьшения этой нестабильности используется предположение, согласно которому дисперсия помехи является функцией только среднего значения по всем наблюдениям за цикл и не зависит от сезонных изменений. При этом чувствительность метода к колебаниям данных в точках наивысших и наинизших значений прогнозируемых переменных соответственно повышается и понижается.

Другой подход заключается в представлении циклических изменений прогнозируемой переменной рядами Фурье:

$$\hat{x}_t = \sum_k A_k \cos(\omega kt) + \sum_k B_k \sin(\omega kt).$$

Здесь A_k и B_k — коэффициенты, полученные по исходным данным с помощью соответствующей регрессии; $\omega = 2\pi/p$ — основная частота; p — число наблюдений в одном сезонном цикле; t — время, за которое должен быть вычислен прогноз; $k = 1, 2, \dots, N$ и суммирование ведется по всем частотам вплоть до частоты Найквиста (наивысшей частоты гармонического разложения дискретного ряда, которая определяется половиной интервала между наблюдениями).

Достоинство такой модели состоит в том, что она обеспечивает стабильность прогноза даже в точках цикла с наименьшими значениями прогнозируемой переменной, так как коэффициенты вычисляются путем усреднения всего набора имеющихся данных, а не только результатов наблюдений в пределах одного цикла. Если для составления прогноза используются только первые члены разложения Фурье, то сезонное сечение представляет собой простую синусоиду с некоторыми амплитудой и фазой, зависящими от коэффициентов A_1 и B_1 . При увеличении числа членов разложения форма сечения может измениться. В соответствии с теоремой Фурье, если в дискретных рядах величина переменной повторяется через каждые m наблюдений, такие ряды могут быть представлены членами гармонического разложения.

Амплитуда j -й гармоники C_j , включенной в модель, равна $\sqrt{A_j^2 + B_j^2}$ (квадрат амплитуды иногда называют мощностью данной гармоники). Пусть σ^2 — дисперсия остаточных разностей между исходными данными и результатами расчетов по модели, включающей гармоники вплоть до j -й. Тогда модель будет соответствовать предыстории, охватывающей от трех до шести предыдущих циклов, если выполнено соотношение $C_j A_j > \sigma/2$. Проверка на соответствие предыстории обычно проводится по критерию наименьших квадратов. Отметим, что при вычислении дисперсии σ^2 необходимо учесть потерю степеней свободы некоторым числом переменных в модели. Это означает, что можно достичь минимальной дисперсии, используя модель с числом переменных, меньшим p . Однако такая модель будет неверной, если мощность

включенной в нее высшей гармоники не будет достаточно большой по сравнению с остаточной дисперсией.

Пусть сезонные волны описываются бесконечным рядом синусоидальных и косинусоидальных функций (рядом Фурье). Если в качестве переменной вместо ωkt взять $(2\pi/p)it$, $i=1, 2, \dots, p$, где p — полный исследуемый период, то значение y_t ряда в момент t вычисляется по формуле

$$y_t = a_0 + \sum_{i=1}^{p/2} \left[A_i \sin\left(\frac{2\pi}{p} it\right) + B_i \cos\left(\frac{2\pi}{p} it\right) \right],$$

где коэффициенты a_0 , A_i и B_i определяются формулами

$$a_0 = \frac{1}{p} \sum_{t=1}^p y_t, \quad A_i = \frac{2}{p} \sum_{t=1}^p y_t \sin\left(\frac{2\pi}{p} it\right), \quad B_i = \frac{2}{p} \sum_{t=1}^p y_t \cos\left(\frac{2\pi}{p} it\right).$$

Отсюда a_0 — среднее значение членов ряда за период. Для p дискретных наблюдений число гармоник не превышает $p/2$. Дисперсия σ_i^2 , учитываемая одной гармоникой, равна $C_i^2/2$, где $C_i = \sqrt{A_i^2 + B_i^2}$, $i=1, 2, \dots, p/2-1$. Для последней гармоники имеем $\sigma_{p/2}^2 = C_i^2$. Поскольку никакие две гармоники не коррелируют между собой, то они не могут учитывать одну и ту же часть общей дисперсии, т. е. дисперсии, учитываемые различными гармониками, суммируются.

В конце процесса прогнозирования следует объединить прогнозные значения с функцией тренда и функцией сезонных колебаний. При этом трудно оценить погрешность. Самое простое — «перенести» рассчитанную ошибку аппроксимации в будущее и получить доверительные интервалы.

Естественно, чем больший период берется для составления функции прогноза, тем больше надо определить гармоник, но скачки амплитуды колебаний уменьшатся и полученные амплитуды могут не соответствовать амплитудам прогнозируемого периода. Здесь предпочтительнее рассмотреть небольшой промежуток, предшествующий предсказываемому.

Для прогнозирования сезонных рядов применима также модель АРПСС. Прогнозы целесообразно вычислять, используя разностное уравнение. Для частного случая сезонных колебаний с периодом в один год мультипликативная модель имеет вид [6]

$$y_{t+l} = y_{t+l-1} + y_{t+l-2} - y_{t+l-3} + a_{t+l} - \theta a_{t+l-1} - \Theta a_{t+l-2} + \theta \Theta a_{t+l-3}.$$

Прогноз $[y_{t+l}] = M[y_{t+l} | \theta, \Theta, y_t, y_{t-1}, \dots]$ является условным математическим ожиданием значений y_{t+l} в момент $t+l$. В этом выражении параметры предполагают известными точно, и ряд чисел y_t, y_{t-1}, \dots — известными достаточно далеко в прошлое, θ и Θ — разные величины (параметры).

Практическое применение модели зависит от следующих фактов:

а) обратимые модели после подгонок к реальным рядам дают прогнозы, существенно зависящие только от сравнительно недавних значений (последних членов) ряда;

б) прогнозы не чувствительны к малым изменениям значений параметров, вносимым, например, ошибками оценивания.

Имеем

$$[y_{t+j}] = \begin{cases} y_{t+j} & \text{при } j \leq 0, \\ \hat{y}_t(j) & \text{при } j > 0, \end{cases}$$

$$[a_{t+j}] = \begin{cases} a_{t+j} & \text{при } j \leq 0, \\ 0 & \text{при } j > 0. \end{cases}$$

Теперь, чтобы получать прогнозы, заменяем неизвестные значения y прогнозами, а неизвестные a — нулями. Известные a_t — это уже вычисленные ошибки прогноза на шаг вперед, т. е. $a_t = y_t - \hat{y}_{t-1}(1)$.

Пример. Для получения прогноза на три месяца вперед при $\theta = 0,4$, $\Theta = 0,6$ имеем

$$y_{t+3} = y_{t+2} + y_{t-9} + y_{t-10} + a_{t+3} - 0,4a_{t+3} - 0,6a_{t-9} + 0,24a_{t-10}.$$

Взяв условные математические ожидания в момент t , получим

$$\hat{y}_t(3) = \hat{y}_t(2) - y_{t-9} - y_{t-10} - 0,6a_{t-9} + 0,24a_{t-10},$$

где $a_{t-9} = y_{t-9} - \hat{y}_{t-10}(1)$, $a_{t-10} = y_{t-10} - \hat{y}_{t-11}(1)$.

В результате имеем

$$\hat{y}_t(3) = \hat{y}_t(2) + 0,4y_{t-9} - 0,76y_{t-10} + 0,6\hat{y}_{t-10}(1) - 0,24\hat{y}_{t-11}(1).$$

Чтобы определить формулы коррекции и получить дисперсию ошибки прогноза $e_t(l)$, необходимо знать значения весов в выражении для модели вида [6]

$$y_t = \sum_{j=0}^{\infty} \psi_j a_{t-j}.$$

Общая формула для коррекции прогноза имеет вид

$$\hat{y}_{t+1}(l) = \hat{y}_{t+1}(l+1) + \psi_l a_{t+1}.$$

Зная веса ψ_j , можем вычислить дисперсию ошибок прогноза для любого упреждения l :

$$D(e) = (1 + \psi_1^2 + \dots + \psi_{l-1}^2) \sigma^2(a).$$

Для вычисления весов ψ_j можно использовать оператор скользящего среднего.

§ 10.10. Диагностическая проверка моделей и ошибка прогноза

Диагностическая проверка моделей может быть начата с оценки параметров моделей. Оценку параметров моделей по результатам наблюдений будем проводить по алгоритмам, описанным в гл. 7. В общем случае перед применением формальной процедуры получения оценок желательно исследовать поведение функции правдоподобия. В каждом случае функция правдоподобия содержит необходимую нам информацию. Так, существование двух максимумов функции правдоподобия приблизительно равной высоты указывает на то, что имеются две группы значений параметров, с помощью которых можно объяснить смысл исходных данных.

Необходимо соблюдать осторожность при интерпретации функции правдоподобия. В общем случае предположение, что логарифмическая функция правдоподобия вблизи ее максимума близка к квадратичной функции, не всегда применимо. Однако такие случаи редки.

Особая осторожность необходима, когда максимум функции правдоподобия лежит на границе измеренных значений или около нее. Первая производная функции правдоподобия в этом случае не равна нулю в максимуме, и квадратичная аппроксимация не является адекватным представлением функции правдоподобия.

Рассмотрим еще раз оценивание параметров при помощи теоремы Байеса, согласно которой если $p(\theta)$ — априорная плотность распределения вероятностей для θ , то апостериорную плотность распределения вероятностей $p(\theta|y)$ для θ после получения данных y можно выразить через априорную вероятность и функцию правдоподобия следующим образом:

$$p(\theta|y) = \frac{p(\theta)L(y|\theta)}{\int p(\theta)L(y|\theta)d\theta}.$$

Интеграл от $p(\theta|y)$ равен единице. Апостериорная плотность вероятности пропорциональна априорной вероятности, умноженной на функцию правдоподобия. В частности, если согласиться, что в интересующей нас области любые значения θ равновероятны, плотность распределения $p(\theta)$ можно считать в этой области равномерной, и, следовательно, $p(\theta|y)$ будет пропорционально функции правдоподобия $L(y|\theta)$. Говоря о равномерности распределения, подразумеваем, что распределение приближенно постоянно в области, где функция правдоподобия существенна и не принимает очень больших значений вне этой области. Для случая нескольких параметров совместное априорное распределение параметров считается пропорциональным следующему выражению:

$$|J_{\theta}|^{1/2} = \left| -M \left[\frac{\partial^2 L}{\partial \theta_i \partial \theta_j} \right] \right|^{1/2}.$$

Точное выражение для функции правдоподобия временного ряда y длиной $N = n + d$ для процесса АРПСС с параметрами (p, d, q) имеет вид

$$L(\Phi, \theta|y) = \sigma_a^{-n} f(\Phi, \theta) \exp \left\{ -\frac{S(\Phi, \theta)}{2\sigma_a^2} \right\},$$

где

$$S(\Phi, \theta) = \sum_{t=-\infty}^n a_t^2(y, \Phi, \theta).$$

Если априорная информация о величинах σ_a , Φ и θ отсутствует, то, поскольку информация о среднем квадратическом отклонении σ_a не дает нам информации о Φ и θ , рационально принять априорную плотность распределения величин Φ , θ и σ_a в виде

$$p(\Phi, \theta, \sigma_a) \approx |J(\Phi, \theta)|^{1/2} \sigma_a^{-1}.$$

Отсюда следует, что апостериорная плотность распределения вероятностей имеет вид

$$p(\Phi, \theta, \sigma_a|y) \approx \sigma_a^{-(n+1)} |J(\Phi, \theta)|^{1/2} f(\Phi, \theta) \exp \left\{ -\frac{S(\Phi, \theta)}{2\sigma_a^2} \right\}.$$

Интегрируя плотность распределения $p(\Phi, \theta, \sigma_a | y)$ от нуля до бесконечности по σ_a , получим точную совместную апостериорную плотность распределения параметров Φ и θ :

$$p(\Phi, \theta | y) \approx |J(\Phi, \theta)|^{1/2} f(\Phi, \theta) (S(\Phi, \theta))^{-n/2}.$$

Если y_t — процесс АРПСС с параметрами $(p, d, 0)$, то $\omega_t = \Delta^d y_t$ — процесс авторегрессии порядка p . Для такого процесса множители $|J(\Phi, \theta)|^{1/2}$ и $f(\Phi, \theta)$, которые во всех случаях являются существенно менее важными, чем множитель $S(\Phi, \theta)$, практически взаимно сокращаются. При данных предположениях параметры Φ процесса АР (p) для заданного ω имеют апостериорную плотность распределения

$$p(\Phi | y) \approx (S(\Phi, \theta))^{-n/2}.$$

В этом случае поверхность, уравнение которой определено функционалом для суммы квадратов отклонений, приближенно совпадающая с поверхностью, описанной функцией правдоподобия при отсутствии априорной информации, совпадает также с поверхностью, описанной функцией плотности апостериорной вероятности.

После того как модель идентифицирована и параметры оценены, подгоняемая модель подвергается диагностической проверке. Один из полезных методов проверки модели состоит в использовании избыточного числа параметров, т. е. в оценивании параметров для несколько более общей модели, чем предполагаемая. Этот метод основан на том, что мы можем заранее предсказать неадекватные свойства модели. Следовательно, полезно дополнить этот подход более общими способами проверки, основанными на остаточных ошибках после подгонки модели. Остаточные ошибки позволяют найти в данных указание, какие изменения модели необходимы. Проверку можно провести, используя:

- 1) автокорреляционную функцию остаточных ошибок;
- 2) кумулятивную периодограмму остаточных ошибок.

Если анализ не выявил, какие добавления новых параметров нужны, это, конечно, не означает, что используемая модель верна. Модель может лишь выдержать испытание. Если будут обнаружены свидетельства неадекватности модели, возникнет необходимость узнать, как изменить модель на следующем итеративном

цикле. Необходимо обнаружить, что именно неадекватно в модели, для того чтобы узнать, как ее изменить.

При недостаточном объеме данных статистическое испытание может отвергнуть модель, которая, тем не менее, вполне адекватна решаемой задаче. Напротив, испытания могут не выявить серьезных отклонений от сделанных предположений, потому что эти испытания нечувствительны к некоторым видам отклонений.

Методы введения избыточных параметров путем обобщения модели применимы в том случае, если мы знаем, какого рода отклонения следует ожидать. Методики, менее зависящие от таких представлений, основываются на анализе остаточных ошибок. Хотя некоторые представления о том, что искать, полезны и в данном случае, эти методы создают больше возможностей находить способ изменения моделей непосредственно из данных.

Рассмотрим диагностическую проверку, используя автокорреляционную функцию. Пусть исследуется модель вида

$$\Phi(B)\hat{\omega}_t = \theta(B)a_t,$$

где $\hat{\omega}_t = \Delta^d y_t$, и были получены оценки ММП параметров $(\hat{\Phi}, \hat{\theta})$, откуда имеем

$$\hat{a}_t = \hat{\theta}^{-1}(B)\hat{\Phi}(B)\hat{\omega}_t.$$

Величины \hat{a}_t будем называть *остаточными ошибками*. Можно доказать, что для адекватной модели выполнено соотношение

$$\hat{a}_t = a_t + O\left(\frac{1}{\sqrt{n}}\right).$$

По мере увеличения длины ряда остаточной ошибки величина \hat{a}_t становится все ближе к белому шуму a_t . Следовательно, можно ожидать, что изучение \hat{a}_t даст возможность выявить и указать природу неадекватности модели. В частности, некоторый характерный вид выборочной автокорреляционной функции для \hat{a}_t выявляют определенные изменения модели.

Предположим, что модель верна и известны точные значения параметров Φ и θ . Тогда выборочные автокорреляции $\rho_k(a)$ элементов ряда a должны быть некоррелированы и распределены приближенно нормально относительно нулевого среднего значения с дисперсией, пропорциональной n^{-1} , и, следовательно, со стандартной ошибкой, пропорциональной $n^{-1/2}$. Эти факты

можно использовать для оценки статистической значимости кажущихся отклонений выборочных автокорреляционных функций от нуля.

На практике не известны истинные значения параметров, мы располагаем только выборочными оценками $\hat{\Phi}$, $\hat{\theta}$, по которым нельзя вычислить элементы a , а только их оценки \hat{a} . Автокорреляции $\rho_k(\hat{a})$ ряда \hat{a} могут дать ценную информацию о недостаточности хорошей подгонке и возможной природе неадекватности модели. Однако может оказаться рискованно придавать статистическую значимость кажущимся отклонениям автокорреляций $\hat{\rho}_k(\hat{a})$ от их теоретических нулевых значений исходя из стандартной ошибки $\approx n^{-1/2}$, соответствующей $\hat{\rho}_k(\hat{a})$. Показано [6], например, что для процесса $AR(10.1)$ с параметром Φ дисперсия оценки $\hat{\rho}_k(\hat{a})$ равна $\Phi^2 n^{-1}$, что может оказаться существенно меньше, чем n^{-1} . Хотя во всех случаях для малых задержек возможны уменьшение дисперсии и сильная корреляция $\hat{\rho}_k(\hat{a})$, при больших задержках эти отличия быстро исчезают. Поэтому использование величины $n^{-1/2}$ в качестве стандартной ошибки для $\hat{\rho}_k(\hat{a})$ будет приводить к недооценке статистической значимости кажущегося отклонения от нуля для автокорреляций при малых задержках, но обычно вполне оправдано для средних и больших задержек.

При небольших задержках можно рассматривать величину $n^{-1/2}$ как верхнюю границу стандартных ошибок $\hat{\rho}_k(\hat{a})$, а не как сами стандартные ошибки. Если пользоваться $n^{-1/2}$ как стандартной ошибкой $\hat{\rho}_k(\hat{a})$ при малых задержках, то можно недооценить значимость кажущихся расхождений. Во многих случаях предпочтительно не исследовать отдельные оценки $\hat{\rho}_k(\hat{a})$, а оценить неадекватность модели по нескольким автокорреляциям, рассматриваемым как единое целое.

В других случаях, особенно при корректировке сезонных рядов, кумулятивная периодограмма является эффективным средством обнаружения периодического отклонения от случайного процесса [6, 41].

Нормированная кумулятивная периодограмма $C(\rho_j)$ элементов ряда a_i определяется по формуле

$$C(\rho_j) = \frac{\sum_{i=1}^j I(\rho_i)}{ns^2},$$

где

$$I(\rho_i) = \frac{2}{n} \left[\left(\sum_{t=1}^n a_t \cos(2n\rho_i t) \right)^2 + \left(\sum_{t=1}^n a_t \sin(2n\rho_i t) \right)^2 \right],$$

$\rho_i = i/n$ — частота, s^2 — оценка дисперсии $\sigma^2(a)$.

В любом из методов прогнозирования рядов собственно прогноз представляет собой, по существу, оценку ожидаемого распределения результатов наблюдений в будущем. Для того чтобы на основе полученного прогноза можно было принимать решение, в большинстве случаев необходимо знать функцию исходного распределения. Если это распределение описывается стандартной функцией, то, определив один или два его параметра, оценим вероятности возможных значений будущих наблюдений.

При формальном анализе часто удобно принять предположение о том, что помеха (ошибка) в исходных данных подчиняется нормальному закону распределения. Однако это предположение нельзя считать обоснованным для всех наблюдений.

Прогнозируемая выходная переменная характеризуется распределением, дисперсия которого зависит от дисперсии шума в исходных данных. Однако, поскольку прогнозы вычисляются на основе результатов многих наблюдений, часто предполагают, что входная переменная описывается нормальным законом распределения независимо от того, каков вид распределения, описывающего шум в исходных данных.

Распределение ошибок при прогнозировании зависит как от распределения самой выходной переменной, так и от распределения соответствующего ей шума, который обычно считается не зависящим от шума в исходных данных. При прогнозировании суммы значений переменной интенсивности в течение времени упреждения, которое намного больше интервала между наблюдениями, распределение ошибок представляет собой распределение суммы выборок из нескольких таких распределений и с возрастом времени упреждения все более и более стремится к нормальному.

В большинстве случаев, представляющих практический интерес, исходное распределение может быть восстановлено из полученного по прогнозу среднего значения и какого-либо другого параметра, характеризующего распределение выходной переменной. В качестве такого параметра, как правило, выбирается стандартное отклонение. Если имеется какой-то набор наблюдаемых величин для построения прогноза, то обычно оказывается, что для каждого

ряда существует определенная зависимость между разбросом ошибок и уровнем прогноза.

Среднее абсолютное отклонение представляет собой оценку отклонений, которая имела бы место, если бы прогнозы разрабатывались на основе минимизации суммы абсолютных величин остаточных разностей между результатами наблюдений и медианой распределения вместо обычно проводимой минимизации суммы квадратов отклонений от среднего. Среднее абсолютное отклонение использовалось вместо стандартного отклонения только потому, что его вычисление требовало меньше времени и меньшего объема памяти. Особенно часто этот показатель использовался в программах ЭВМ для прогнозирования спроса. При использовании современных компьютеров применение среднего абсолютного отклонения нецелесообразно.

Если распределение описывается нормальным законом, то имеем среднее абсолютное отклонение $m = 0,8\sigma$ (точное значение коэффициента при σ равно $\sqrt{2/\pi} = 0,7979$). Поскольку средняя ошибка прогноза должна быть равна нулю, то дисперсия определяется средней квадратической ошибкой. Выбрав подходящую модель процесса, пересмотрим значения коэффициентов этой модели по значениям средней квадратической ошибки каждый раз, когда поступает новая информация и заново измеряется ошибка прогноза.

Если шум результатов наблюдений автокоррелирован незначительно, то дисперсия ошибок прогноза переменной интенсивности в течение времени упреждения будет равна сумме дисперсий ошибок прогнозов для отдельных интервалов, на которые разбивается время упреждения l с целью периодического уточнения прогнозов. Следовательно, стандартное отклонение, используемое для принятия решений, приблизительно равно $\sigma\sqrt{l}$. В тех случаях, когда имеет место автокорреляция шума, выражение для стандартного отклонения может быть записано в виде $\sigma_l = l^\delta \sigma_1$, где σ_1 — стандартное отклонение ошибок одного знака, l — время упреждения, измеренное в интервалах между проверками прогноза, и δ — постоянная, которая является характеристикой всего семейства прогнозируемых рядов.

Как уже отмечалось, точность прогноза зависит от шума входных переменных. При этом разумно считать, что прогнозы не содержат систематической ошибки, так как средняя ошибка прогноза должна быть равна нулю. Пока модель соответствует изучаемому

процессу, ошибки прогноза должны колебаться около нуля. Если же модель прогноза ошибочна или станет ошибочной из-за того, что резко изменится сам процесс, то появится последовательность «положительных» или «отрицательных» ошибок и, следовательно, средняя ошибка прогноза больше не будет равна нулю. Имеет смысл систематически осуществлять проверку близости средней ошибки прогноза к нулю. Если средняя ошибка превысит некоторый заранее установленный предел, пользователю необходимо принять меры для своевременной корректировки прогноза. Для осуществления такой корректировки прежде всего требуется установить конкретную причину смещения прогноза, после этого внести в модель достаточно обоснованные изменения, принять новые значения для одного или большего количества коэффициентов, изменить некоторые или все весовые множители либо, вообще, изменить сам вид модели.

§ 10.11. Пример прогнозирования газопотребления

Рассмотрим пример построения прогноза потребления газа некоторым объектом в зависимости от температуры окружающей среды. Значения газопотребления p в базовом периоде задавались каждые сутки, а также задавались соответствующие им значения температуры t воздуха. Особенностью данной задачи является тот факт, что значения температуры t воздуха имеют одно распределение, а значения газопотребления $p(t)$ при фиксированной температуре в силу различных причин имеет другое распределение. Распределения значений газопотребления и температуры легко установить по результатам наблюдений и с помощью ММП составить функционал для оценки параметров предлагаемых зависимостей. В приведенных ниже расчетах прогнозов принималось, что все исходные данные не коррелированы и подчиняются нормальному закону распределения с известными дисперсиями $\sigma^2(p_i)$ и $\sigma^2(t_i)$, $i = 1, 2, \dots, m$, значения которых находятся по результатам наблюдений. Предполагалось также, что продолжительность периода прогнозирования составляет семь дней с уточнением прогноза каждый день по мере поступления новой информации.

В процессе исследований имеющихся данных было установлено, что в поставленной задаче для получения прогнозных значений достаточно воспользоваться линейной конъюгентной моделью, учитывающей неопределенности и в значениях температуры t , и в

значениях газопотребления $p(t)$, т. е. оценки параметров k и b линейной функции $p = kt + b$, описывающей газопотребление в зависимости от температуры, находятся как координаты точки минимума функционала

$$F(p, t) = \sum_{i=1}^n \frac{(p_i - kt_i - b)^2}{\sigma^2(p_i) + k^2 \sigma^2(t_i)}. \quad (10.4)$$

Ковариационная (дисперсионная) матрица оценок k и b является обратной матрице

$$M = \begin{pmatrix} -\frac{\partial^2 F(p, t)}{\partial k^2} & -\frac{\partial^2 F(p, t)}{\partial k \partial b} \\ -\frac{\partial^2 F(p, t)}{\partial k \partial b} & -\frac{\partial^2 F(p, t)}{\partial b^2} \end{pmatrix}.$$

Элементы матрицы вычисляются при найденных оценках параметров k и b .

При учете дополнительных природных факторов, таких как влажность воздуха v , скорость ветра w и других, достаточно воспользоваться линейной зависимостью газопотребления p от этих факторов, т. е.

$$p = b + k_1 t + k_2 v + k_3 w.$$

Оценки параметров b , k_1 , k_2 , k_3 находятся в процессе минимизации функционала (см. § 7.12)

$$F(p, t, v, w) = \sum_{i=1}^n \frac{(p_i - b - k_1 t - k_2 v - k_3 w)^2}{\sigma^2(p_i) + k_1^2 \sigma^2(t_i) + k_2^2 \sigma^2(v_i) + k_3^2 \sigma^2(w_i)}. \quad (10.5)$$

Ковариационная матрица оценок \hat{b} , \hat{k}_1 , \hat{k}_2 , \hat{k}_3 находится также с помощью матрицы четвертого порядка, составленной из вторых производных этого функционала по искомым параметрам b , k_1 , k_2 , k_3 .

Верхний предел суммирования в (10.4) и (10.5) определяется продолжительностью базового периода. Продолжительность базового периода, с одной стороны, должна содержать число точек, превышающее число параметров как минимум в два раза, а с другой стороны, должна быть не столь велика, чтобы не изменились условия протекающего процесса, хотя увеличение числа точек увеличивает точность получаемых оценок параметров. Проведенные расчеты показали, что продолжительность базового периода может быть от 10 до 20 дней.

Часто для анализа газопотребления используют нелинейные зависимости. Например, потребление газа при переходе от зимнего периода через весенний до летнего периода или от летнего периода через осенний до зимнего периода хорошо описывается логистической функцией. При коротких временных периодах для описания газопотребления можно использовать полиномиальную зависимость; как правило, порядок полинома не превышает двух.

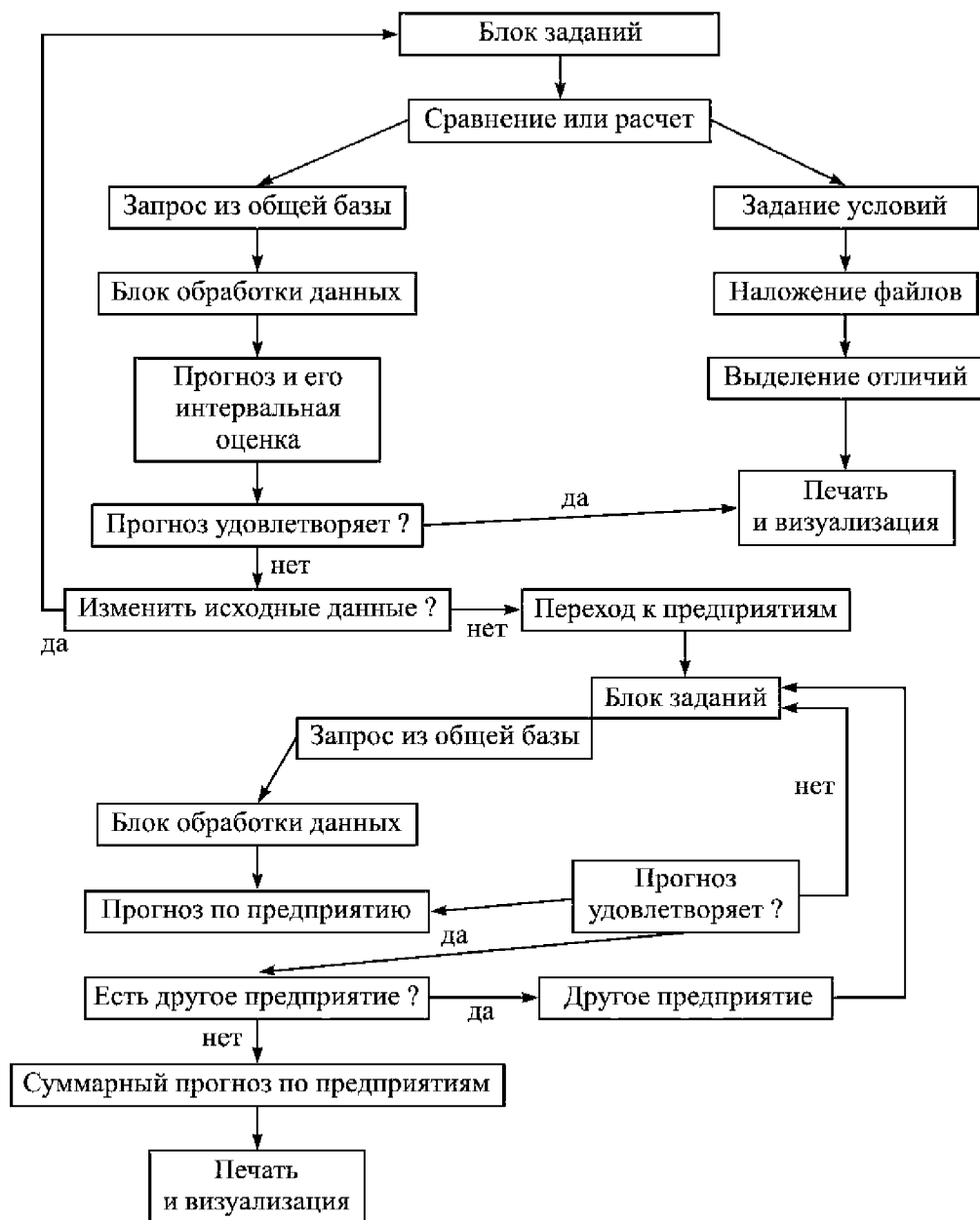


Рис. 10.1. Общая блок-схема расчета прогнозов

Тогда можно применить формулы из § 7.13 или воспользоваться приближенной формулой: если потребление как функция температуры имеет вид $p = b + k_1 t + k_2 t^2$, то оценки \hat{b} , \hat{k}_1 и \hat{k}_2 находятся как координаты точки минимума функционала

$$F_1(p, t) = \sum_{i=1}^n \frac{(p_i - b - k_1 t_i - k_2 t_i^2)^2}{\sigma^2(p_i) + \sigma^2(t_i)(k_1^2 + 4k_2^2 t_i^2)}.$$

При этом в обоих случаях получаются достаточно близкие оценки параметров b , k_1 и k_2 .

Общая схема получения прогноза газопотребления приведена на рис. 10.1. В нее входят блоки заданий и расчетный блок. Их блок-схемы представлены соответственно на рис. 10.2 и 10.3.

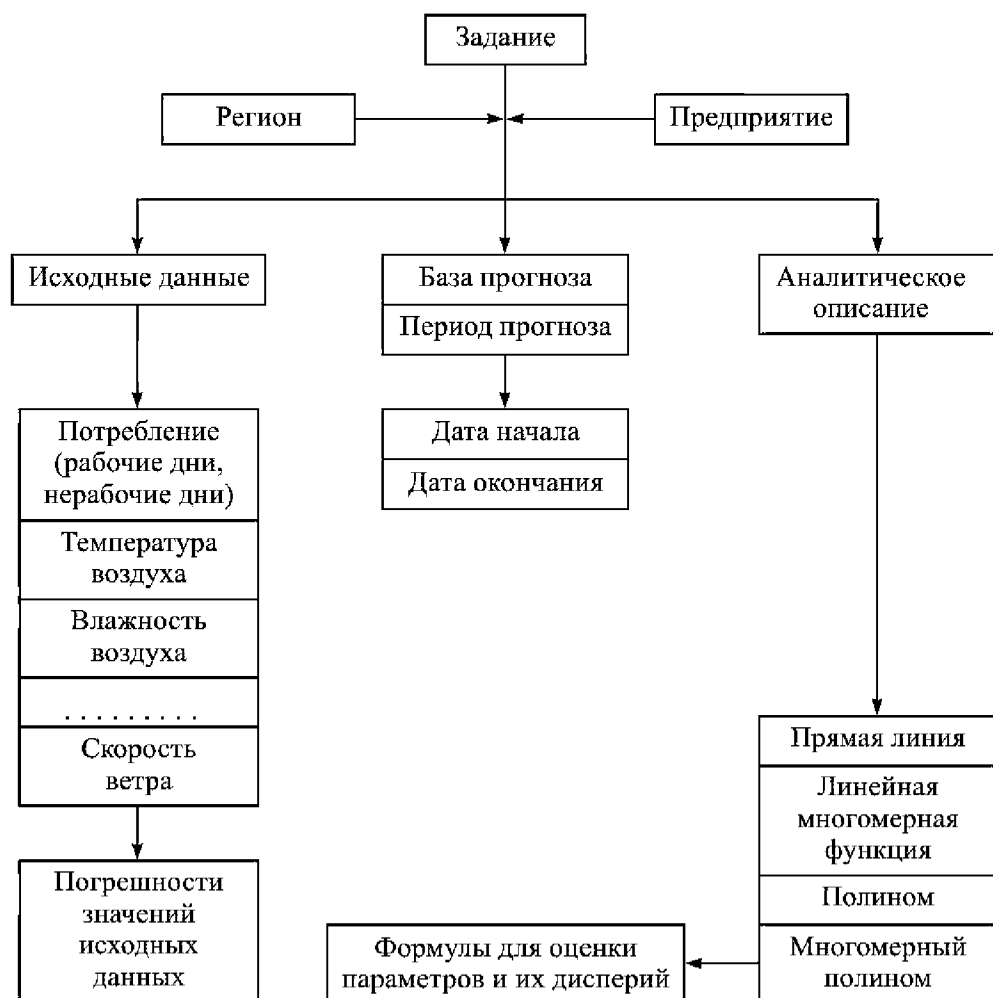


Рис. 10.2. Блок заданий



Рис. 10.3. Блок обработки данных

Результаты расчетов прогнозов газопотребления приведены на рис. 10.4 и 10.5. Расчеты проводились для рабочих и нерабочих дней. Весь массив исходных данных за год (356 дней) разбивался на две части: первая часть — база для прогноза, вторая часть использовалась для сравнения прогноза с реальными данными. База для прогноза составляла 30 дней (см. рис. 10.4).

Предварительно для получения оценок параметров прогнозирующих функций строилась зависимость газопотребления от температуры. Чтобы продемонстрировать различие оценок для разных моделей, методом наименьших квадратов строились для каждого варианта три прямые регрессии: одна не учитывала неопределенность по температуре, т. е. *невязка* бралась только по газопотреблению, другая не учитывала неопределенность по газопотреблению, а также строилась прямая с учетом неопределенностей температуры и газопотребления (условия для конфлюэнтного анализа).

Соответственно параметры k и b для регрессионных моделей были обозначены: k_{ipw} и b_{ipw} (если учитывалась только неопреде-

ленность по температуре) или k_{ptw} и b_{ptw} (если учитывалась только неопределенность по газопотреблению) для рабочих дней, k_{ptr} и b_{ptr} или k_{ptr} и b_{ptr} для нерабочих дней.

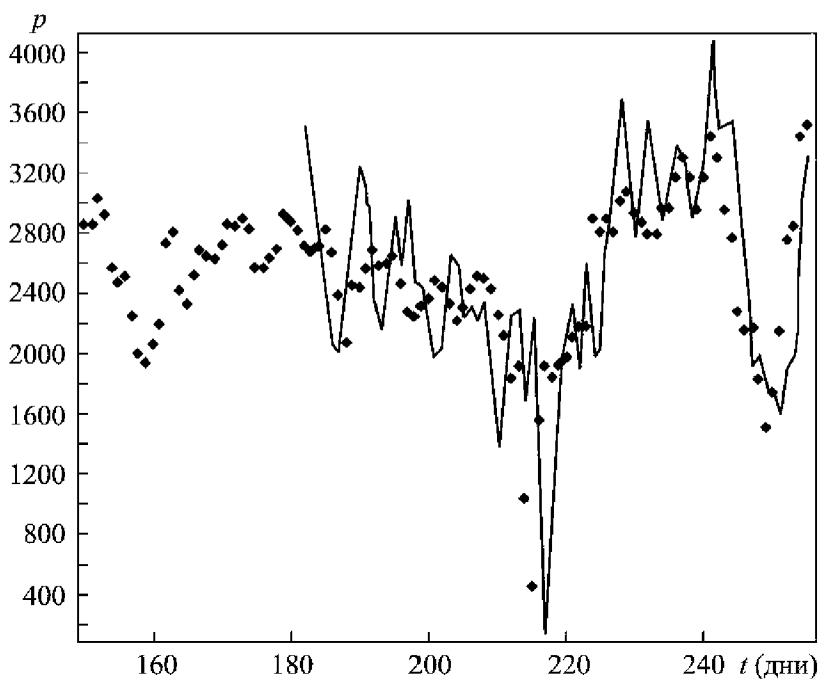


Рис. 10.4. Исходный расчет прогноза (база 30 дней)

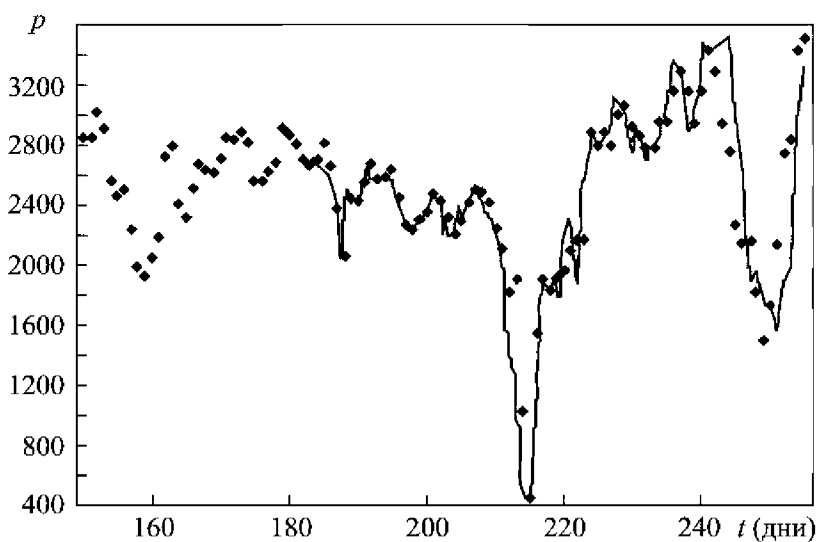


Рис. 10.5. Подправленный расчет прогноза (база 30 дней)

Для конфлюэнтной модели использовались обозначения $k_{\text{konf } w}$ и $b_{\text{konf } w}$, $k_{\text{konf } r}$ и $b_{\text{konf } r}$.

Для примера, приведенного на рис. 10.4, получены следующие оценки значений параметров k и b при $\sigma(t_i) = 7^\circ\text{C}$ и дисперсии $\sigma^2(p_i)$ для заданных значений p_i и фиксированной температуры t_i .

Дисперсия значений газопотребления вычислялась по представленным результатам, предварительно отнормированным по значениям температуры.

В итоге получено:

1) для рабочих дней

$$\begin{aligned}\hat{k}_{tpw} &= -152,6, & \hat{b}_{tpw} &= 4496,8, \\ \hat{k}_{ptw} &= -223,5, & \hat{b}_{ptw} &= 3174,5, \\ \hat{k}_{\text{konf } w} &= -160,3, & \hat{b}_{\text{konf } w} &= 4355,2;\end{aligned}$$

2) для нерабочих дней

$$\begin{aligned}\hat{k}_{tpr} &= -137,2, & \hat{b}_{tpr} &= 5173,1, \\ \hat{k}_{ptr} &= -225,4, & \hat{b}_{ptr} &= 3621,5, \\ \hat{k}_{\text{konf } r} &= -146,9, & \hat{b}_{\text{konf } r} &= 4994,9.\end{aligned}$$

Эти данные убедительно демонстрируют различие результатов прогнозирования по указанным уравнениям прямой.

После оценки параметров k и b строился прогноз с учетом параметров конфлюэнтной модели. К последнему значению базового периода добавлялось значение $\hat{k}_{\text{konf } w}$, умноженное на разность «истинных» значений температур Δt между последней точкой базового периода и первой точкой прогноза. Аналогично, к первой точке прогноза добавлялось соответствующее значение $\hat{k}_{\text{konf } w}\Delta t$, в результате получалась вторая точка прогноза и т. д. По рассчитанным значениям дисперсий $\hat{k}_{\text{konf } w}$ и Δt с помощью формул переноса ошибок определялась средняя квадратическая ошибка прогноза в каждой точке. Если прогнозируемые значения считались согласно уравнению прямой $p = \hat{b}_{\text{konf}} + \hat{k}_{\text{konf}}t$, то есте-

ственно, что последний прогноз был менее эффективен, так как здесь на прогноз дополнительно влияет ошибка в оценке \hat{b}_{conf} .

Чтобы получить для нелинейных зависимостей газопотребления $p(t)$ первую точку прогноза при температуре $t_{1п}$, необходимо к последней точке базового периода с температурой $t_{пб}$ добавить величину $\Delta p_1 = p(t_{1п}) - p(t_{пб})$, вычисленную согласно выбранной модели. Для получения второй точки прогноза необходимо к первому прогнозному значению прибавить величину $\Delta p_2 = p(t_{2п}) - p(t_{1п})$ и т. д.

Некоторые результаты такого подхода приведены на рис. 10.4 и 10.5. Прогноз, представленный на рис. 10.4, выявил, что данные по температуре были ненадежны: трудно предположить, что первая точка прогноза будет так отклоняться от прогнозного значения по какой-то другой причине.

Далее была проведена корректировка значений температуры в прогнозируемых точках. Результаты уточненного прогноза представлены на рис. 10.5. Заметим, что использование скорректированных значений температуры несущественно меняет оценки параметров модели, применяемой для прогнозирования газопотребления.

ЛИТЕРАТУРА

1. Акулич И.Л. Математическое программирование в примерах и задачах. — М.: Высшая школа, 1986. — 317 с.
2. Бард Й. Нелинейное оценивание параметров. — М.: Статистика, 1979. — 349 с.
3. Бартон Д., Вард Г. Справочник по радиолокационным измерениям. — М.: Сов. радио, 1976. — 392 с.
4. Беллман Р. Динамическое программирование / Пер. с англ. под ред. Н.Н. Воробьева. — М.: ИЛ, 1960. — 400 с.
5. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования / Пер. с англ. под ред. А.А. Первозванского. — М.: Наука, 1965. — 458 с.
6. Бокс Дж., Дженкинс Г. Анализ временных рядов. Прогноз и управление / Под ред. В.Ф. Писаренко. — М.: Мир, 1974. — 406 с.
7. Борисов А.И., Алексеев А. В. Обработка нечеткой информации в системах принятия решений. — М.: Радио и связь, 1989. — 124 с.
8. Борисов А.И., Крумберг О. А., Федоров И.П. Принятие решений на основе нечетких моделей. — Рига: Зинатне, 1990. — 184 с.
9. Бородюк В.И., Воццинин А.П. Ошибки регистрации независимых переменных в задачах множественной регрессии // Заводская лаборатория. — 1973. — Т. 39, № 7. — С. 217–222.
10. Браунли К. А. Статистическая теория и методология в науке и технике / Под ред. Л.Н. Большева. — М.: Наука, 1977. — 407 с.
11. Вентцель Е.С. Теория вероятностей. — Изд. 4-е. — М.: Наука, 1969. — 576 с.
12. Вентцель Е.С. Исследование операций. — М.: Сов. радио, 1972. — 552 с.
13. Вильямс Н.Н. Параметрическое программирование в экономике (методы оптимальных решений). — М.: Статистика, 1976. — 96 с.
14. Вопросы статистической теории распознавания / Под ред. Б.В. Варского. — М.: Сов. радио, 1967. — 400 с.
15. Гольштейн Е. Г., Юдин Д. Б. Новые направления в линейном программировании. — М.: Сов. радио, 1966. — 524 с.
16. Гончарский А. В., Леонов А.С., Ягола А. Г. Некоторое обобщение принципа невязки для случая оператора, заданного с ошибкой // Докл. АН СССР. — 1972. — Т. 203, № 6. — С. 1238–1239.
17. Горелик А.Л., Скрипкин В. А. Методы распознавания. — М.: Высшая школа, 1977. — 222 с.
18. Горелик А.Л., Скрипкин В. А. Методы распознавания. 2-е изд. — М.: Высшая школа, 1989. — 232 с.

19. Горелик А.Л., Барабаш Ю.Л., Кривошеев О.В., Эпштейн С.С. Селекция и распознавание на основе локационной информации / Под ред. А.Л. Горелика. — М.: Радио и связь, 1990. — 240 с.
20. Грешилов А.А. Некорректные задачи цифровой обработки информации и сигналов. — М.: Радио и связь, 1984. — 161 с.
21. Грешилов А. А. Анализ и синтез стохастических систем. Параметрические модели и конфлюэнтный анализ. — М.: Радио и связь, 1990. — 320 с.
22. Грешилов А.А. Метод наименьших квадратов и элементы конфлюэнтного анализа — М.: Изд-во МГТУ им. Н. Э. Баумана, 1990. — 67 с.
23. Грешилов А.А., Стакун В.А., Стакун А.А. Математические методы построения прогнозов. — М.: Радио и связь, 1997. — 112 с.
24. Грешилов А.А., Стакун В.А., Стакун А.А. Статистические методы принятия решений с элементами конфлюэнтного анализа. — М.: Радио и связь, 1998. — 112 с.
25. Де Гроот М. Оптимальные статистические решения: Пер. с англ. — М.: Мир, 1974. — 491 с.
26. Демиденко Е.З. Линейная и нелинейная регрессии. — М.: Финансы и статистика, 1981. — 302 с.
27. Демиденко Е.З. Оптимизация и регрессия. — М.: Наука, 1989. — 292 с.
28. Дуда Р., Харт П. Распознавание образов и анализ сцен: Пер. с англ. — М.: Мир, 1976. — 511 с.
29. Дюбуа Д., Прад А. Теория возможностей. Приложения к представлению знаний в информатике / Пер. с фр. В. Б. Тарасова под ред. С.А. Орловского. — М.: Радио и связь, 1990. — 286 с.
30. Жилинская Е.И., Товмаченко Н.Н., Федоров В.В. Методы регрессионного анализа при наличии ошибки в предикторных переменных. — М.: Изд-во АН СССР, 1978. — 34 с.
31. Журавлев Ю.И., Никифоров В.В. Проблемы реализации алгоритма обобщенной невязки // Кибернетика. — 1971. — № 3. — С. 19–21.
32. Зангвилл У.И. Нелинейное программирование / Пер. с англ. под ред. Е.Г. Гольштейна. — М.: Сов. радио, 1973. — 312 с.
33. Зельнер А. Байесовские методы в эконометрии. — М.: Статистика, 1980. — 438 с.
34. Зуховицкий С.И., Авдеева Л.И. Линейное и выпуклое программирование. — М.: Наука, 1964. — 348 с.
35. Искусственный интеллект. Справочник. В 2 т. / Под ред. Д.А. Поспелова. — М.: Радио и связь, 1990. — Т. 2, 248 с.
36. Исследование операций. Методологические основы и математические методы. Т. 1 / Пер. с англ. под ред. И.М. Макарова, И.М. Бескровного. — М.: Мир, 1981. — 712 с.
37. Исследование операций. Модели и применение. Т. 2 / Пер. с англ. под ред. И.М. Макарова, И.М. Бескровного. — М.: Мир, 1981. — 677 с.

38. *Калихман И.Л.* Сборник задач по математическому программированию. — М.: Высшая школа, 1975. — 270 с.
39. *Карпелевич Ф.И., Садовский Л.Е.* Элементы линейной алгебры и линейного программирования. — М.: Наука, 1965. — 275 с.
40. *Кендалл М., Стьюарт А.* Статистические выводы и связи / Пер. с англ. под ред. А.Н. Колмогорова. — М.: Наука, 1973. — 590 с.
41. *Кильдишев Г.С., Френкель А.А.* Анализ временных рядов и прогнозирование. — М.: Статистика, 1973. — 103 с.
42. *Корн Г., Корн Т.* Справочник по математике для научных работников и инженеров: Пер. с англ. — М.: Наука, 1970. — 720 с.
43. *Крамер Г.* Математические методы статистики / Под ред. А.Н. Колмогорова. — М.: Мир, 1975. — 648 с.
44. *Круг Г.К., Сосулин Ю.А., Фатуев В.А.* Планирование эксперимента в задачах идентификации и экстраполяции. — М.: Наука, 1977. — 208 с.
45. *Круг Г.К., Кабанов В.А., Фомин Г.А., Фомина Г.С.* Планирование эксперимента в задачах нелинейного оценивания и распознавания образов. — М.: Наука, 1981. — 172 с.
46. *Кудрявцев Л.Д.* Курс математического анализа. Т. 2 / Изд. 2-е. — М.: Высшая школа, 1988. — 575 с.
47. *Лазарев В.Г., Лазарев Ю.В.* Динамическое управление потоками информации в сетях связи. — М.: Радио и связь, 1983. — 216 с.
48. *Леонов А.С.* Некоторые аспекты реализации регуляризующего алгоритма обобщенной невязки // Обработка и интерпретация физических экспериментов. Вып. 4. — М.: Изд-во МГУ, 1976. — С. 69–81.
49. *Линник Ю.В.* Метод наименьших квадратов и основы математико-статистической теории обработки информации. — М.: Физматгиз, 1958. — 333 с.
50. *Мартин Дж.* Системный анализ передачи данных. Т. 2 / Пер. с англ. под ред. В. С. Лапина. — М.: Мир, 1975. — 431 с.
51. *Митрофанов Д.Г., Ермоленко В.П.* Распознавание воздушных целей за счет измерения их пространственной протяженности // Зарубежная радиоэлектроника. — 1996. — № 1. — С. 53–56.
52. Модели. Алгоритмы. Принятие решений: Сб. статей. — М.: Наука, 1989. — 250 с.
53. *Монаков В.М., Беляева Э.С., Краснер Н.Я.* Методы оптимизации: Пособие для учителя. — М.: Просвещение, 1978. — 175 с.
54. *Морозов В.А.* Линейные и нелинейные некорректные задачи // Итоги науки и техники. Математический анализ. — Л. — М.: ВИНТИ, 1973. — 400 с.
55. *Муртаф Б.* Современное линейное программирование: Теория и практика / Пер. с англ. под ред. И.А. Станевичуса. — М.: Мир, 1984. — 224 с.
56. *Небабин В.Г., Сергеев В.В.* Методы и техника радиолокационного распознавания. — М.: Радио и связь, 1984. — 152 с.

57. Небабин В.Г., Белоус О.И. Методы и техника противодействия радиолокационному распознаванию // Зарубежная радиоэлектроника. — 1987. — № 2. — С. 142–144.

58. Небабин В.Г., Гришин В.К. Методы и техника радиолокационного распознавания: Современное состояние, тенденции развития, перспективы // Зарубежная радиоэлектроника. — 1992. — № 10. — С. 46–49.

59. Патрик Э. Основы теории распознавания образов: Пер. с англ. — М.: Сов. радио, 1980. — 408 с.

60. Персептрон — система распознавания образов / Под ред. А.Г. Ивахненко. — Киев: Наукова думка, 1975. — 278 с.

61. Петросян Л.А., Зенкевич Н.А., Семин Е.А. Теория игр: Учеб. пособие. — М.: Высшая школа, 1998. — 304 с.

62. Распознавание образов. Состояние и перспективы / Пер. с англ. под ред. И. Б. Гуревича. — М.: Радио и связь, 1985. — 103 с.

63. Репин В. Г., Тартаковский Г.П. Статистический синтез при априорной неопределенности и адаптация информационных систем. — М.: Сов. радио, 1977. — 432 с.

64. Рокафеллор Р. Выпуклый анализ / Пер. с англ. под ред. А.Д. Иоффе, В.М. Тихомирова. — М.: Мир, 1973. — 469 с.

65. Селекция и распознавание на основе локационной информации / Под ред. А.Л. Горелика. — М.: Радио и связь, 1990. — 239 с.

66. Сосулин Ю.Г., Фишман М.М. Теория последовательных решений и ее применения. — М.: Радио и связь, 1985. — 272 с.

67. Стайнберг Б.Д. Формирование радиолокационного изображения самолета в диапазоне СВЧ // ТИИЭР. — 1988. — Т. 76, № 12. — С. 26–46.

68. Статистические методы в экспериментальной физике / Пер. с англ. под ред. А.П. Тяпкина. — М.: Атомиздат, 1976. — 335 с.

69. Стронгин Р.Г. Численные методы в многоэкстремальных задачах. — М.: Наука, 1978. — 239 с.

70. Сухарев А.Г., Тимохов А.В., Федоров В.В. Курс методов оптимизации. — М.: Физматлит, 1986. — 326 с.

71. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. — М.: Наука, 1979. — 288 с.

72. Трубицын Е.Г., Ермоленко В.П. Алгоритм распознавания сложных воздушных целей // Зарубежная радиоэлектроника. — 1992. — № 10. — С. 82–84.

73. Уилкс С. Математическая статистика: Пер. с англ. — М.: Наука, 1967. — 632 с.

74. Уотермен Д. Руководство по экспертным системам: Пер. с англ. — М.: Мир, 1989. — 388 с.

75. Успенский А.Б., Федоров В.В. Вычислительные аспекты МНК при анализе и планировании регрессионных экспериментов. — М.: Изд-во Моск. ун-та, 1975. — 168 с.

76. Федоров В. В. Теория оптимального эксперимента. — М.: Наука, 1971. — 312 с.

77. Фиакко А., Мак-Кормик Г. Нелинейное программирование. Методы последовательной безусловной минимизации / Пер. с англ. под ред. Е.Г. Гольштейна. — М.: Мир, 1972. — 240 с.
78. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей / Пер. с англ. под ред. Б. Г. Сушкова. — М.: Мир, 1984. — 496 с.
79. Финкельштейн М.И., Мендельсон В.А., Кутев В.Л. Радиолокация слоистых земных покровов. — М.: Сов. радио, 1977. — 174 с.
80. Фомин Я.А., Тарловский Г.Р. Статистическая теория распознавания образов. — М.: Радио и связь, 1986. — 263 с.
81. Фор А. Восприятие и распознавание образов: Пер. с фр. — М.: Машиностроение, 1989. — 271 с.
82. Франклин Д., Кармоди У. Проблемы обработки нечеткой информации // ТИИЭР. — 1988. — Т. 78, № 10. — С. 32–36.
83. Фу К. Последовательные методы в распознавании образов и обучении машин: Пер. с англ. — М.: Наука, 1971. — 255 с.
84. Фу К. Структурные методы в распознавании образов: Пер. с англ. — М.: Мир, 1977. — 319 с.
85. Фукунага К. Введение в статистическую теорию распознавания: Пер. с англ. — М.: Наука, 1979. — 367 с.
86. Хармут Х.Ф. Несинусоидальные волны в радиолокации и радиосвязи. — М.: Радио и связь, 1985. — 376 с.
87. Химмельблау Д. Анализ процессов статистическими методами: Пер. с англ. — М.: Мир, 1973. — 957 с.
88. Ху Т. Целочисленное программирование и потоки в сетях / Пер. с англ. под ред. А.А. Фридмана. — М.: Мир, 1974. — 419 с.
89. Худсон Д. Статистика для физиков / Пер. с англ. под ред. Е.М. Лейкина. — М.: Мир, 1970. — 296 с.
90. Шишов С. А. Классификаторы на основе нейронных структур // Зарубежная радиоэлектроника. — 1992. — № 8. — С. 78–83.
91. Штойер Р. Многокритериальная оптимизация. Теория, вычисления и приложения / Пер. с англ. под ред. А.В. Лотова. — М.: Радио и связь, 1992. — 504 с.
92. Юдин Д.Б., Гольштейн Е.Г. Линейное программирование. Теория и конечные методы. — М.: Физматгиз, 1963. — 775 с.
93. Яноши Л. Теория и практика обработки результатов измерений / Пер. с англ. под ред. Н.П. Клепикова. — М.: Мир, 1968. — 462 с.
94. Frisch R. Statistical confluence analysis by means of complete regression systems. — Oslo, 1934. — 192 p.
95. Fuller W.A. Measurement error models. — N. Y.: Wiley, 1987. — 440 p.
96. Gleser L.J. Improvements of the naive approach to estimation in nonlinear errors-in-variables regression models // Statist. Anal. Meas. Error Models and Appl.: Proc. AMS-IMS-SIAM. It Summer Res. Conf., Arcata, Calif., June 10–16, 1989. — Providence, 1990. — P. 99–114.

97. *Haikin L.M., Kushnir A.F., Dainty A.* Combined automated and off-line computer processing system for seismic monitoring with small aperture arrays // *Seismological Research Letters*. — 1998. — V. 69, №. 3. — P. 235–247.

98. *Haikin L.M., Kushnir A.F., Dainty A., Troitskiy E.V.* Statistical classification approach to discrimination between weak earthquakes and quarry blasts recorded by Israel Seismic Network // *Physics of Earth and Planetary Interiors*. — 1999. — V. 113. — P. 161–182.

99. *Jefferys W.H.* Robust estimation when more than one variable per equation of condition has error // *Biometrika*. — 1990. — V. 77. — P. 597–607.

100. *Lyons L.* On estimating systematic errors from repeated measurements // *J. Phys. A*. — 1992. — V. 25. — P. 1967–1979.

101. *Parallel Distributed Processing. V. I* / Eds. D. Rumelhart, J. McClelland. — Cambridge: MIT Press, 1988.

102. *Schater D.W.* Measurement error model estimation using iteratively weighted least squares // *Statist. Anal. Meas. Error Models and Appl.: Proc. AMS-IMS-SIAM. It Summer Res. Conf., Arcata, Calif., June 10–16, 1989*. — Providence, 1990. — P. 129–138.

103. *Schneeweiß H., Mittag H-J.* Lineare Modelle mit fehler behafteten Daten. — Heidelberg — Wien: Physica-Verlag, 1986. — 504 p.

104. *Spiegelman C.H.* Plotting techniques for error-in-variables problems // *Statist. Anal. Meas. Error Models and Appl.: Proc. AMS-IMS-SIAM. It Summer Res. Conf., Arcata, Calif., June 10–16, 1989*. — Providence, 1990. — P. 167–168.

105. *Werbos P.J.* Back propagation through time: What it does and how to do it // *Proc. IEEE*. — 1990. — V. 78, №. 10. P. 1550–1560.

106. *Whittemore A.S.* Error-variables regression problems in epidemiology // *Statist. Anal. Meas. Error Models and Appl.: Proc. AMS-IMS-SIAM. It Summer Res. Conf., Arcata, Calif., June 10–16, 1989*. — Providence, 1990. — P. 17–31.

107. *Widrow B., Lehr M.A.* 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation // *Proc. IEEE*. — 1990. — V. 78, №. 9. P. 1415–1442

108. *Williamson J.H.* Least-squares fitting of straight line // *Canad. J. Phys.* — 1968. — V. 46, №. 16. — P. 1845.

109. *Yohai V.J., Zamar R.H.* Bounded influence estimation in the error-in-variables model // *Statist. Anal. Meas. Error Models and Appl.: Proc. AMS-IMS-SIAM. It Summer Res. Conf., Arcata, Calif., June 10–16, 1989*. — Providence, 1990. — P. 243–248.

110. *Cetin M., Karl W.C.* Future-enhanced synthetic aperture radar image formation based on nonquadratic regularization // *IEEE Trans. Image Processing*. — 2001. — V. 10, №. 4. — P. 623–631.

ОПИСАНИЕ ПРОГРАММЫ «РЕГРЕССИЯ» (ИНСТРУКЦИЯ ДЛЯ ПОЛЬЗОВАТЕЛЯ)

Программа «Регрессия» предназначена для расчета параметров линейной регрессии Y на X , X на Y и ортогональной регрессии, а также параметров прямой линии, параболы и полинома 3-й степени с помощью МНК. Главное окно программы представлено на рис. П1.1.

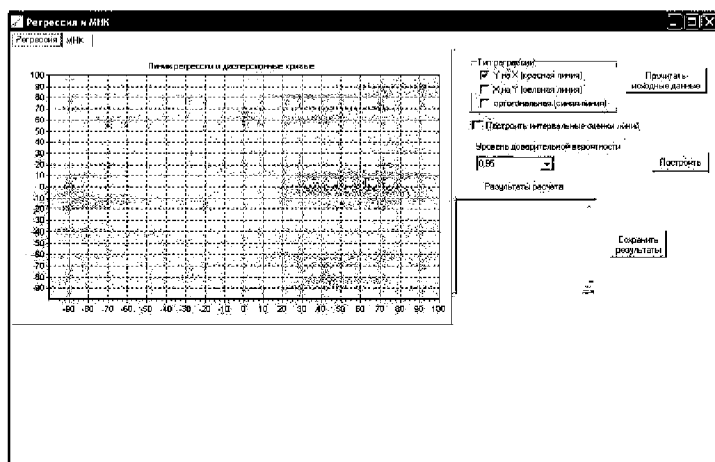


Рис. П1.1. Окно программы, вкладка «Регрессия»

В верхней части окна программы находятся две вкладки, которые позволяют переключаться между «режимами» регрессии и МНК.

Рассмотрим элементы, находящиеся на вкладке «Регрессия».

Для проведения расчетов необходимо загрузить в программу данные, для чего служит кнопка «Прочитать исходные данные». При ее нажатии появляется диалоговое окно выбора текстового файла с данными (рис. П1.2).

Текстовый файл можно создать в любом простом текстовом редакторе, например в Блокноте. Внимательно изучите формат файла исходных данных! В этом файле пять строк. В первой строке содержится число пар точек, по которым рассчитывается регрессия. Во второй строке находятся значения для X , в третьей — средние

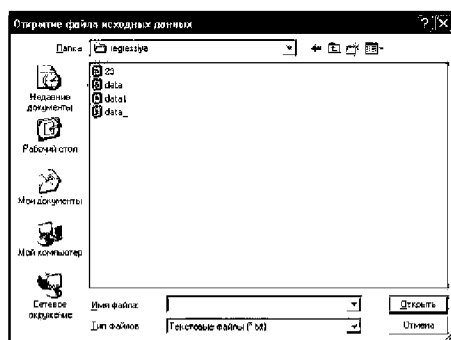


Рис. П1.2. Диалоговое окно выбора файла исходных данных

квадратические отклонения каждого измерения для X , в четвертой и пятой — значения и средние квадратические отклонения для Y . В строке между числами может быть любое число пробелов и символов табуляции, между строками с данными не должно быть пустых строк. Пример текстового файла представлен ниже:

```
5
0 2 3 4 8
1.1 2.2 1.4 3.1 1.1
8 2 6 4 13
1 2 1 2 3
```

По данным одного файла строятся все три линии регрессии.

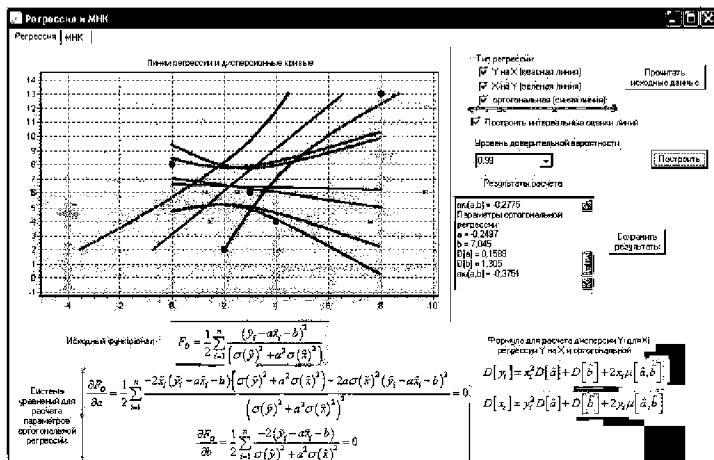
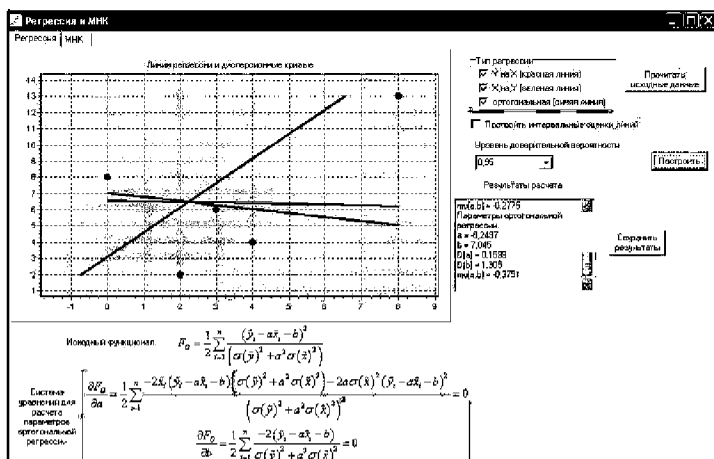
Следует обратить внимание, что при записи в файл дробных чисел разделителем целой и дробной частей является точка, а не запятая (см. третью строку приведенного примера текстового файла).

Данные для нового варианта регрессии должны записываться в отдельный файл исходных данных! В файле с данными должно быть пять строк (как указано выше).

После того как данный файл прочитан, можно построить линии регрессии. Для этого нужно нажать кнопку «Построить». Тип регрессий выбирается установкой соответствующих флажков на поле «Тип регрессии».

При нажатии кнопки «Построить» будут построены выбранные регрессии и кривые, а справа в белом поле будут выведены параметры регрессий (рис. П1.3).

При этом дополнительно в нижней части окна программы отображаются формулы для дисперсий значений x_i и y_i и система уравнений, из которой находятся оценки параметров ортогональной регрессии (рис. П1.4).



Если отметить флажком параметр «Построить интервальные оценки линий», то при выводе линий регрессии будут отображаться интервальные оценки линий регрессии. Предварительно можно в выпадающем списке задать уровень доверительной вероятности, который будет определять «ширину» доверительного интервала («расстояние» между интервальными кривыми).

Если необходимо увидеть весь процесс решения задачи, результаты расчетов можно сохранить. Для этого нужно нажать кнопку «Сохранить результаты», после чего появится диалоговое окно сохранения файла (рис. П1.5). В этом окне нужно задать имя файла, в который запишутся результаты, выбрать директорию, где

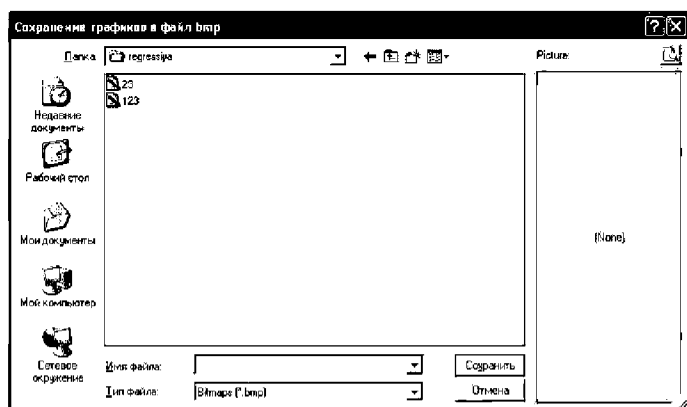


Рис. П1.5. Диалоговое окно сохранения файла

будет сохранен файл, и нажать кнопку «Сохранить». При этом будут созданы два файла: один с расширением `bmp` — в него запишутся графики; другой с расширением `txt` — в него запишется весь процесс поэтапного решения задачи с ответами на каждом этапе, что позволит проконтролировать расчеты, и рассчитанные параметры регрессий. Для построения линий регрессии Y на X и X на Y применяется векторно-матричная форма записи решений систем уравнений.

Для приведенных на рис. П1.3–П1.4 данных текстовый файл результатов будет иметь следующий вид:

Параметры регрессии

Y на X :

Матрица X системы:

0 1

2 1

3 1

4 1

8 1

Матрица Y системы:

8

2

6

4

13

D – матрица дисперсий измерений Y :

1 0 0 0 0

0 4 0 0 0

0 0 1 0 0

0 0 0 4 0

0 0 0 0 9

D^{-1} - обратная матрица дисперсий измерений Y:

```
1 0 0 0 0
0 0,25 0 0 0
0 0 1 0 0
0 0 0 0,25 0
0 0 0 0 0,1111
```

Матрица $X^t * D^{-1} * X$:

```
21,11 5,389
5,389 2,611
```

Матрица $(X^t * D^{-1} * X)^{-1}$ - обратная (она же является ковариационной):

```
0,1001 -0,2066
-0,2066 0,8094
```

Матрица $X^t * D^{-1} * Y$:

```
34,56
16,94
```

Примечание: X^t - транспонированная матрица X

$a = -0,04153$

$b = 6,575$

$D[a] = 0,1001$

$D[b] = 0,8094$

$\mu[a,b] = -0,2066$; μ - корреляционный момент параметров регрессии a и b

Параметры регрессии

X на Y:

Матрица Y системы:

```
8 1
2 1
6 1
4 1
13 1
```

Матрица X системы:

```
0
2
3
4
8
```

D - матрица дисперсий измерений X:

```
1,21 0 0 0 0
0 4,84 0 0 0
0 0 1,96 0 0
0 0 0 9,61 0
0 0 0 0 1,21
```

$D^{(-1)}$ - обратная матрица дисперсий измерений X :

0,8264 0 0 0 0

0 0,2066 0 0 0

0 0 0,5102 0 0

0 0 0 0,1041 0

0 0 0 0 0,8264

Матрица $Y_t * D^{(-1)} * Y$:

213,4 21,25

21,25 2,474

Матрица $(Y_t * D^{(-1)} * Y)^{(-1)}$ - обратная (она же является ковариационной):

0,03231 -0,2775

-0,2775 2,788

Матрица $Y_t * D^{(-1)} * X$:

97,63

8,972

Примечание: Y_t - транспонированная матрица Y

$a = 0,6647$

$b = -2,082$

$D[a] = 0,03231$

$D[b] = 2,788$

$mu[a,b] = -0,2775$; mu - корреляционный момент параметров регрессии a и b

Параметры ортогональной
регрессии:

$a = -0,2497$

$b = 7,045$

$D[a] = 0,1589$

$D[b] = 1,305$

$mu[a,b] = -0,3751$; mu - корреляционный момент параметров регрессии a и b

Вкладка «МНК» предназначена для расчетов параметров прямой линии, параболы и полинома 3-й степени методом наименьших квадратов. Также здесь предусмотрена возможность построения доверительных интервалов для указанных типов линий.

Внешний вид программы при активной вкладке «МНК» приведен на рис. П1.6.

Для загрузки исходных данных в программу служит кнопка «Прочитать исходные данные», при нажатии которой открывается стандартное диалоговое окно выбора файла, аналогичное приведенному на рис. П1.2. Внимательно изучите формат файла исходных данных!

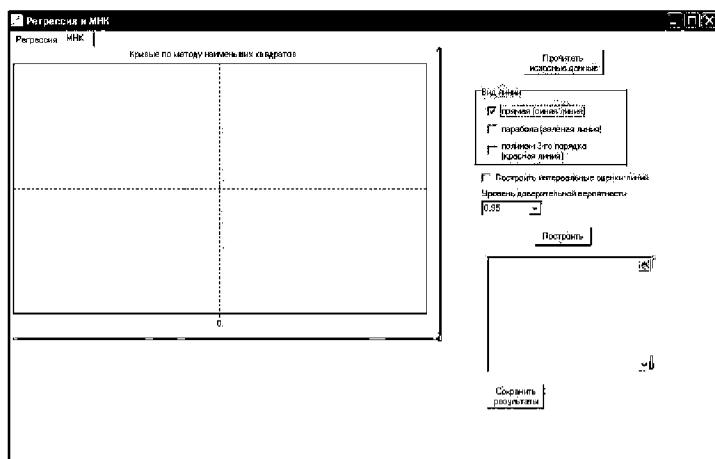


Рис. П1.6. Окно программы, вкладка «МНК»

Файл исходных данных должен состоять из трех строк: в первой строке указывается число точек, во второй перечислены значения X , в третьей — Y . Пример содержимого файла приведен ниже:

```
7
8.1 6.2 9.9 12.1 14.2 14.9 18.2
1.9 6.9 3.9 8.1 21.2 30.1 13.8
```

После выбора файла следует в поле «Вид линии» отметить «галочками» те типы линий, параметры которых будут рассчитываться, и нажать кнопку «Построить». При этом будут отображены исходные точки и соответствующие линии, а в текстовом поле справа — параметры выбранных типов линий (рис. П1.7).

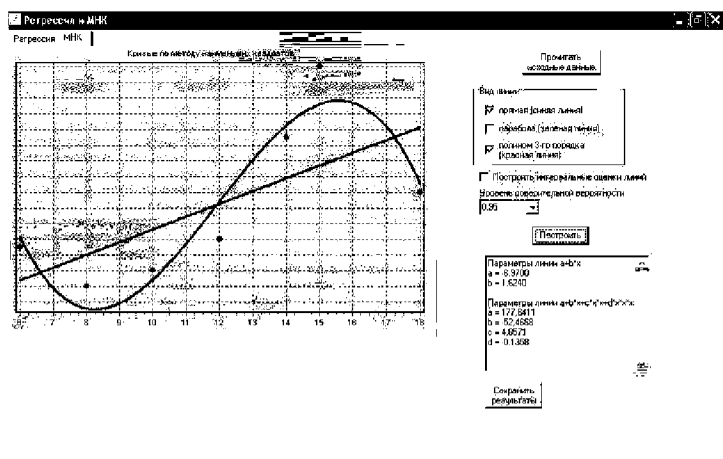


Рис. П1.7. Расчет параметров линий методом наименьших квадратов

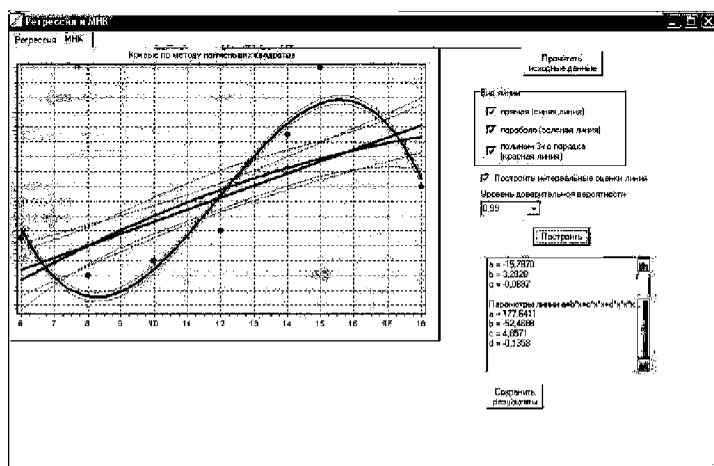


Рис. П1.8. Расчет параметров линий методом наименьших квадратов

Если отметить флажком параметр «Построить интервальные оценки линий», то при выводе будут отображаться интервальные оценки прямой линии, параболы или полинома 3-й степени (рис. П1.8). Предварительно можно в выпадающем списке задать уровень доверительной вероятности, который будет определять «ширину» доверительного интервала («расстояние» между интервальными кривыми).

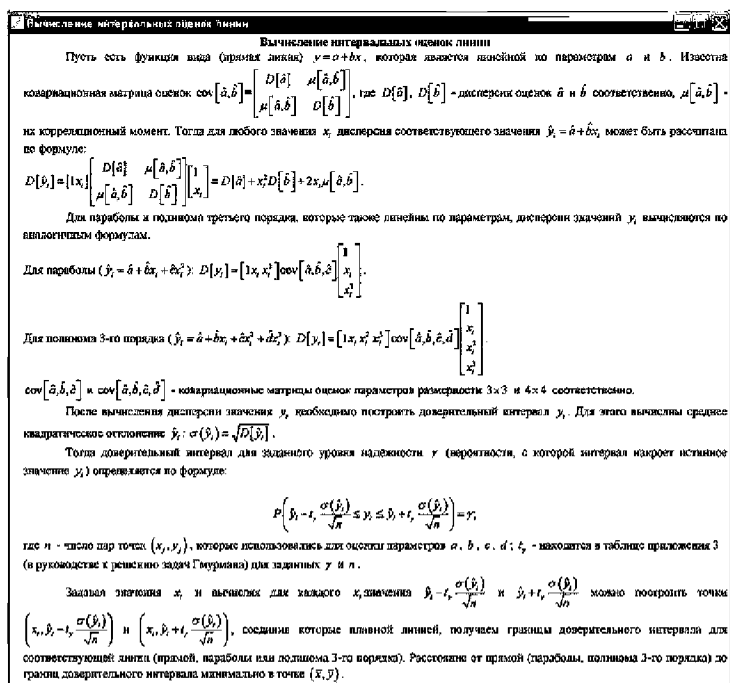


Рис. П1.9. Окно «Вычисление интервальных оценок линий»

Если отметить флажком поле «Построить интервальные оценки линий» на экране появляется окно с основными формулами для расчета ковариационных матриц параметров линий и интервальных оценок линий (рис. П1.9).

Для сохранения результатов расчета в файле служит кнопка «Сохранить результаты», которая работает аналогично кнопке на вкладке «Регрессия», т. е. в файл записываются не только параметры линий и ковариационные матрицы, но и все этапы расчетов. Пример текстового файла с сохраненными результатами приведен ниже:

```
-----
Прямая линия
матрица X:
1,0000 8,1000
1,0000 6,2000
1,0000 9,9000
1,0000 12,1000
1,0000 14,2000
1,0000 14,9000
1,0000 18,2000
матрица Xt (транспонированная) :
1,0000 1,0000 1,0000 1,0000 1,0000 1,0000 1,0000
8,1000 6,2000 9,9000 12,1000 14,2000 14,9000 18,2000
матрица Xt*X:
7,0000 83,6000
83,6000 1103,3600
матрица Xt*Y:
85,9000
1195,4800
обратная матрица Xt*X^(-1):
1,5021 -0,1138
-0,1138 0,0095
Оценки параметров:
a=-7,0294
b=1,6161
Оценка дисперсии Y:
sigma(y)^2=14,7356
Ковариационная матрица оценок параметров
sigma(y)^2*Xt*X^(-1):
22,1338 -1,6770
-1,6770 0,1404
-----
```

```
Парабола
матрица X:
1,0000 8,1000 65,6100
1,0000 6,2000 38,4400
1,0000 9,9000 98,0100
```

```

1,0000 12,1000 146,4100
1,0000 14,2000 201,6400
1,0000 14,9000 222,0100
1,0000 18,2000 331,2400
матрица  $X^t$  (транспонированная):
1,0000 1,0000 1,0000 1,0000 1,0000 1,0000 1,0000
8,1000 6,2000 9,9000 12,1000 14,2000 14,9000 18,2000
65,6100 38,4400 98,0100 146,4100 201,6400 222,0100 331,2400
матрица  $X^t \cdot X$ :
7,0000 83,6000 1103,3600
83,6000 1103,3600 15711,4340
1103,3600 15711,4340 236491,2200
матрица  $X^t \cdot Y$ :
85,9000
1195,4800
17486,4360
обратная матрица  $X^t \cdot X^{-1}$ :
13,9148 -2,4057 0,0949
-2,4057 0,4327 -0,0175
0,0949 -0,0175 0,0007
Оценки параметров:
a=-21,1534
b=4,2240
c=-0,1080
Оценка дисперсии Y:
 $\sigma(y)^2=9,4636$ 
Ковариационная матрица оценок параметров
 $\sigma(y)^2 \cdot X^t \cdot X^{-1}$ :
131,6836 -22,7668 0,8982
-22,7668 4,0951 -0,1658
0,8982 -0,1658 0,0069
-----
Полином 3-й степени
матрица X:
1,0000 8,1000 65,6100 531,4410
1,0000 6,2000 38,4400 238,3280
1,0000 9,9000 98,0100 970,2990
1,0000 12,1000 146,4100 1771,5610
1,0000 14,2000 201,6400 2863,2880
1,0000 14,9000 222,0100 3307,9490
1,0000 18,2000 331,2400 6028,5680
матрица  $X^t$  (транспонированная):
1,0000 1,0000 1,0000 1,0000 1,0000 1,0000 1,0000
8,1000 6,2000 9,9000 12,1000 14,2000 14,9000 18,2000
65,6100 38,4400 98,0100 146,4100 201,6400 222,0100 331,2400
531,4410 238,3280 970,2990 1771,5610 2863,2880 3307,9490
6028,5680

```


матрица $X^t \cdot X$:

7,0000 83,6000 1103,3600 15711,4340
 83,6000 1103,3600 15711,4340 236491,2200
 1103,3600 15711,4340 236491,2200 3707156,4000
 15711,4340 236491,2200 3707156,4000 59903715,0000

матрица $X^t \cdot Y$:

85,9000
 1195,4800
 17486,4360
 264253,2200

обратная матрица $X^t \cdot X^{-1}$:

170,9042 -46,5471 3,9153 -0,1034
 -46,5471 12,8441 -1,0917 0,0291
 3,9153 -1,0917 0,0937 -0,0025
 -0,1034 0,0291 -0,0025 0,0001

Оценки параметров:

a=185,2742
 b=-53,8181
 c=4,9155
 d=-0,1359

Оценка дисперсии Y:

$\sigma(y)^2 = 0,2332$

Ковариационная матрица оценок параметров

$\sigma(y)^2 \cdot X^t \cdot X^{-1}$:

39,8475 -10,8528 0,9129 -0,0241
 -10,8528 2,9947 -0,2545 0,0068
 0,9129 -0,2545 0,0218 -0,0006
 -0,0241 0,0068 -0,0006 0,0000

Замечание. При расчете дисперсии (ковариационной матрицы) оценок параметров линий в МНК предполагается, что дисперсия значений y_i неизвестна и ее оценка находится по формуле

$$\hat{\sigma}(y_i)^2 = \hat{\sigma}(y)^2 = \frac{F_{YX}}{n - m}.$$

Эта оценка учитывается в операции

$\sigma(y)^2 (X^T X)^{-1}$ при определении дисперсии параметров линии.

ПРОГРАММЫ ДЛЯ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Для решения задачи линейного программирования можно использовать две программы, представленные на диске: LinProg и MatProg. Обе они обеспечивают решение задачи линейного программирования в пошаговом режиме, обеспечивая на каждом шаге вывод результатов.

П2.1. Описание программы LinProg

Рассмотрим работу программы на конкретном числовом примере.

Требуется найти минимальное значение функции

$$f(x) = x_1 - x_2$$

при ограничениях

$$\begin{cases} 2x_1 + x_2 \leq 7; \\ -x_1 + 4x_2 \geq 10; \end{cases}$$

$$x_1, x_2 \geq 0.$$

Приведем задачу к каноническому виду. Для этого в первое неравенство добавим слабую переменную x_3 , а из второго неравенства вычтем x_4 :

$$\begin{cases} 2x_1 + x_2 + x_3 = 7; \\ -x_1 + 4x_2 - x_4 = 10. \end{cases}$$

Запишем задачу в матричном виде:

$$f = \mathbf{C} \cdot \mathbf{X} \rightarrow \min,$$

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{b},$$

где

$$\mathbf{C} = [1 \quad -1 \quad 0 \quad 0], \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ -1 & 4 & 0 & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 7 \\ 10 \end{bmatrix}.$$

Сформируем файл с исходными данными задачи, который будет использовать программа LinProg. Он может быть создан в любом текстовом редакторе, например, в Блокноте и имеет вид:

```

flag   m     n
      0     c1  c2  ...  cn
      b1   a11 a12  ...  a1n
      b2   a21 a22  ...  a2n
      .....
      bm   am1 am2  ...  amn

```

Параметр `flag` может принимать только два значения: 0 — если целевая функция минимизируется, 1 — если целевая функция максимизируется; m — число ограничений, n — число переменных; c_i , b_j , a_{lk} — коэффициенты целевой функции, правой и левой части ограничений соответственно; $i, k = 1, \dots, n$; $j, l = 1, \dots, m$.

Если требуется записать в файл дробные числа, то они должны быть записаны в виде десятичных дробей, причем разделителем дробной и целой частей является точка! В строке между числами может быть любое число пробелов и знаков табуляции (это справедливо для всех описанных ниже файлов исходных данных).

Для исходных данных примера файл запишется следующим образом:

```

0 2 4
0 1 -1 0 0
7 2 1 1 0
10 -1 4 0 -1

```

Сохраним его под именем `primer.txt`.

Замечание. Если правая часть b_j j -го ограничения отрицательна, то перед записью этого ограничения в файл его надо предварительно умножить на -1 .

Запускаем программу LinProg, главное окно которой представлено на рис. П2.1.

Открываем файл `primer.txt` с исходными данными задачи. Для этого нажимаем меню «Файл» и далее выбираем пункт «Загрузить». В появившемся диалоговом окне выбора файла указываем `primer.txt` (рис. П2.2) и нажимаем кнопку «Открыть».

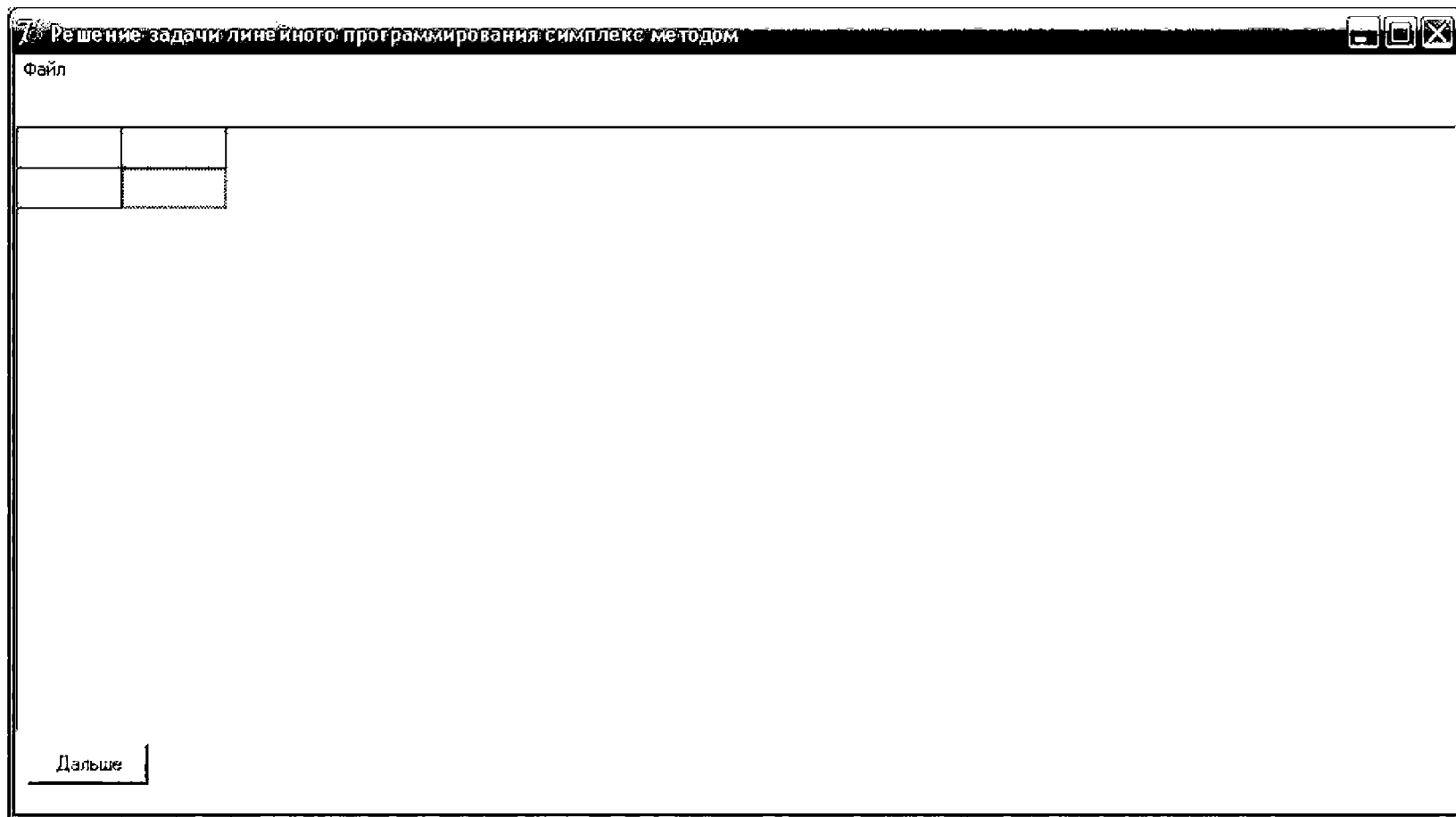


Рис. П2.1. Главное окно программы LinProg

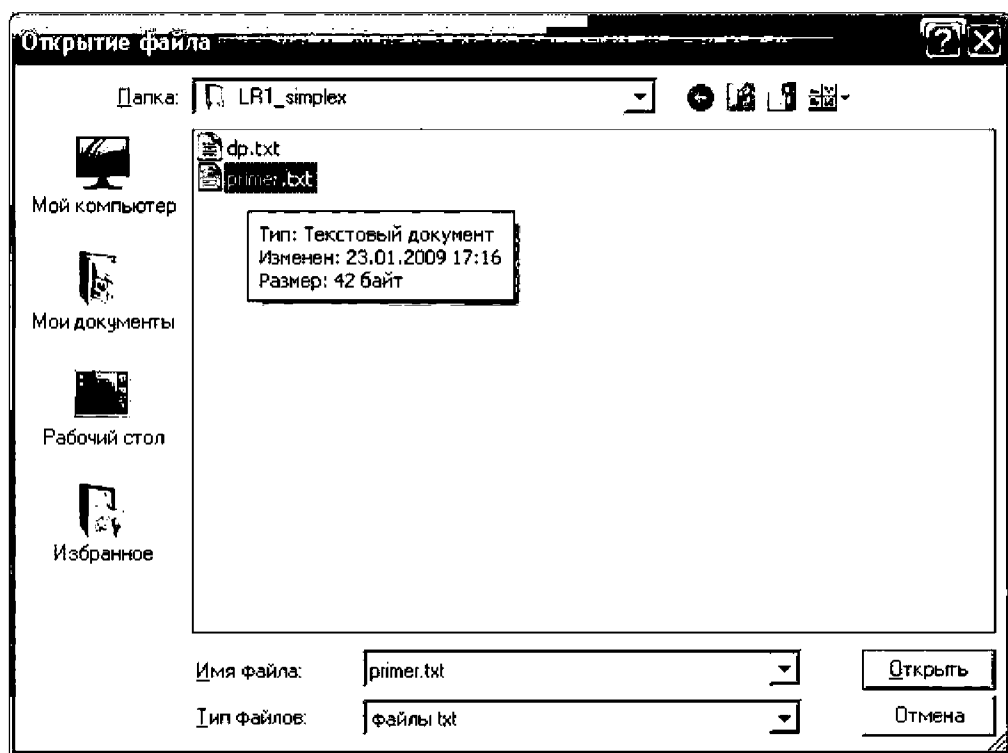


Рис. П2.2. Диалоговое окно выбора файла с данными

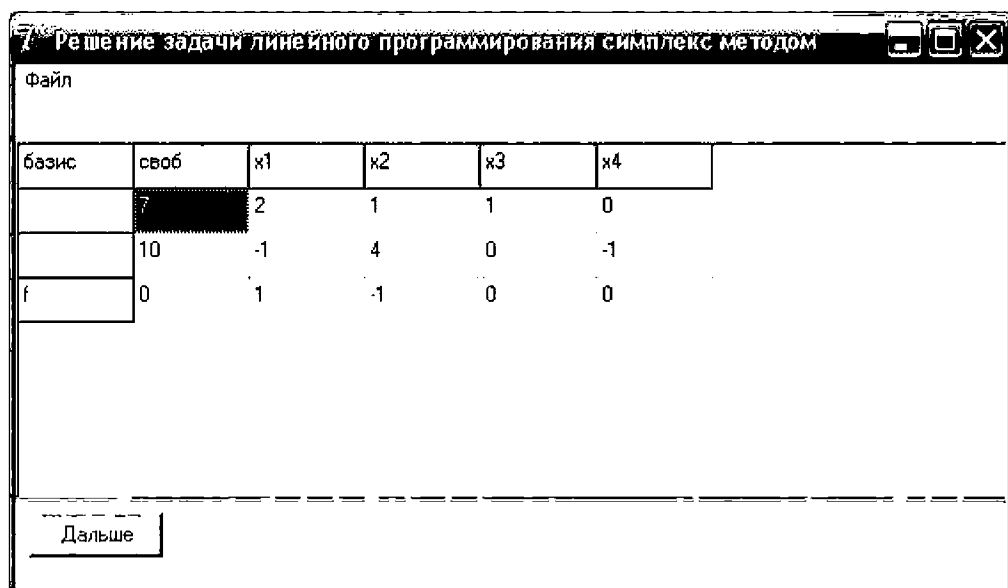


Рис. П2.3. Главное окно программы с загруженными данными

Данные из файла отобразятся в главном окне программы (рис. П2.3).

Замечание. Программа осуществляет минимизацию целевой функции. Если в файле исходных данных $\text{flag} = 1$ (задача максимизации), то предварительно при считывании данных из файла в программу коэффициенты целевой функции умножаются на -1 .

Выбираем пункт меню «Файл» и далее «Поиск опорного решения». Если в симплекс-таблице в столбце свободных членов нет отрицательных элементов, то появляется сообщение «Добавляем искусственные переменные».

После нажатия кнопки «ОК» в диалоговом окне с этим сообщением в исходную симплекс-таблицу вводятся искусственные переменные.

Замечание. Для отыскания допустимого базисного решения используется метод искусственного базиса. В каждое ограничение-равенство вводится по одной дополнительной новой переменной x_s , $s = n + 1, \dots, n + m$. После этого формируется фиктивная целевая функция phi , элементы которой представляют собой сумму всех элементов ограничений задачи в соответствующем столбце, взятую со знаком минус. Сначала решается задача минимизации фиктивной целевой функции phi . Элементы строки с целевой функцией пересчитываются на каждом шаге так же, как и другие элементы симплекс-таблицы. Если система ограничений не противоречива, то на k -й итерации строка phi обнуляется, она вычеркивается и далее процесс продолжается со строкой целевой функции f .

Если в столбце свободных членов есть отрицательные элементы, то появляется сообщение «В столбце свободных членов есть отрицательные элементы!» и внизу главного окна программы появляется кнопка «Умножить строки с отрицательными свободными членами на -1 ». Если нажать эту кнопку, то все строки в симплекс-таблице с отрицательными свободными членами умножаются на -1 .

После того как введены искусственные базисные переменные, выводится сообщение, где предлагается указать разрешающий элемент в симплекс-таблице. При нажатии кнопки «ОК» кнопка «Дальше» (расположена в нижнем левом углу главного окна программы) становится активной.

Чтобы выделить разрешающий элемент, надо щелкнуть один раз левой кнопкой мышки по соответствующей клетке симплекс-таблицы и нажать кнопку «Дальше» (рис. П2.4).

базис	своб	x1	x2	x3	x4	x5	x6
x5	7	2	1	1	0	1	0
x6	10	-1	4	0	-1	0	1
f	0	1	-1	0	0	0	0
phi	-17	-1	-5	-1	1	-1	-1

Дальше

Рис. П2.4. Выбор генерального элемента

Если разрешающий элемент выбран верно, то проводится пересчет элементов симплекс-таблицы. Чтобы продолжить далее пошагово алгоритм симплекс-метода, надо нажать кнопку «Дальше». При этом выделенная синим цветом ячейка будет указывать на каждой итерации на ведущий элемент.

Как только искусственные базисные элементы будут выведены из базиса, появится соответствующее сообщение «Искусственные переменные выведены из базиса».

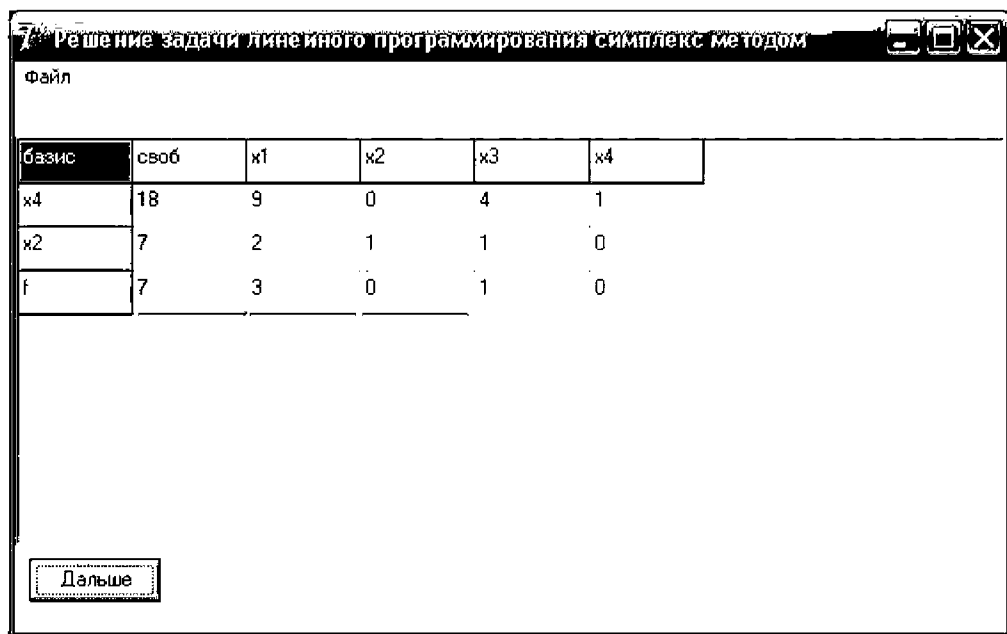
Если на данном шаге получена оптимальная точка, то отображается сообщение «Оптимальное решение найдено».

По анализу оптимальной симплекс-таблицы можно сделать вывод о наличии других альтернативных решений. Если в оптимальной таблице среди коэффициентов целевой функции, соответствующих свободным переменным, есть нулевые, то, возможно, существуют и другие оптимальные точки. Чтобы найти альтернативные решения, надо выполнить следующие действия: 1) каждый столбец оптимальной симплекс-таблицы, соответствующий свободной переменной, в котором коэффициент целевой функции, равен нулю, рассмотреть как ведущий; 2) для каждого такого ведущего столбца попытаться найти его разрешающий элемент; 3) если разрешающий элемент найден, то провести новую итерацию симплекс-метода. Полученное решение и будет другой опти-

мальной точкой. Если разрешающий элемент не найден, то перейти к следующему ведущему столбцу.

В случае наличия нескольких оптимальных решений выводится сообщение «Есть альтернативные минимумы! Укажите их!» При этом сами дополнительные экстремумы в программе не ищутся, это нужно сделать самостоятельно по полученной оптимальной симплекс-таблице.

Если разрешающий элемент определен неправильно, то выводится сообщение об ошибке «Ведущий элемент выбран неверно» и кнопка «Дальше» становится неактивной. Чтобы указать разрешающий элемент снова, надо выбрать в меню «Файл» пункт «Поиск опорного решения» и повторить описанные выше действия.



Решение задачи линейного программирования симплекс методом

Файл

базис	своб	x1	x2	x3	x4
x4	18	9	0	4	1
x2	7	2	1	1	0
f	7	3	0	1	0

Дальше

Рис. П2.5. Оптимальная симплекс-таблица

После четырех итераций получаем оптимальное решение (рис. П2.5).

П2.2. Описание программы MatProg

В программной системе MatProg реализовано решение задач линейного программирования (ЛП): симплекс метод, транспортная задача, поиск максимального потока и минимального разреза сети, регрессионный анализ и поиск в семантических сетях, используемый при разработке экспертных систем и решателей.

В рассматриваемой системе все файлы данных должны быть созданы только программно средствами системы. Для каждой задачи устанавливается свой фильтр: вводится свое расширение для файла данных. Файлы, созданные в блокноте, не открываются.



Рис. П2.6. Главное окно программы MatProg

Методика решения каждой задачи приводится в справках в меню системы. Главное окно программы представлено на рис. П2.6.

Решение задачи ЛП с помощью программы MatProg

Для решения в системе необходимо предварительно привести математическую модель задачи ЗЛП к канонической форме. По математическому описанию задачи создать файл описания ЗЛП (или открыть ранее созданный, для этого перейти к пункту, соответствующему рис. П2.12). Далее вводятся данные с последующим сохранением их в файле. Рассмотрим решение задачи на изложенном выше числовом примере.

После запуска системы выбрать в основном меню пункт: «Симплекс метод». Для создания нового файла данных выбрать пункт меню создания файла. Для этого служит подменю «Создать новый файл» пункта меню «Симплекс-метод» (рис. П2.7).

После выбора этого пункта появляются диалоговые окна для задания числа ограничений и числа переменных задачи (рис. П2.8).

Далее требуется указать, ищется минимум или максимум целевой функции (рис. П2.9). Ввести 0, если задача на минимум, ввести 1, если задача на максимум.

Определяется размерность симплекс таблицы и далее выводится следующая форма (рис. П2.10): в первой строке вводятся коэффициенты целевой функции, первый столбец – столбец свободных

членов и строки матрицы ограничений. Далее в форму вводятся значения коэффициентов целевой функции и ограничений.



Рис. П2.7. Подменю в разделе «Симплекс-метод»

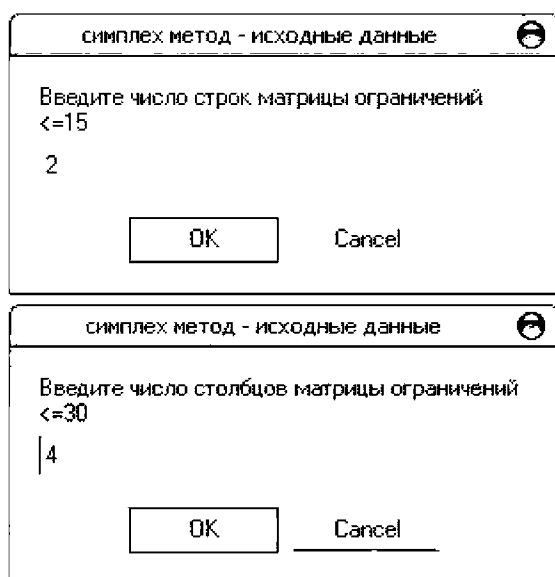


Рис. П2.8. Ввод числа ограничений и переменных задачи

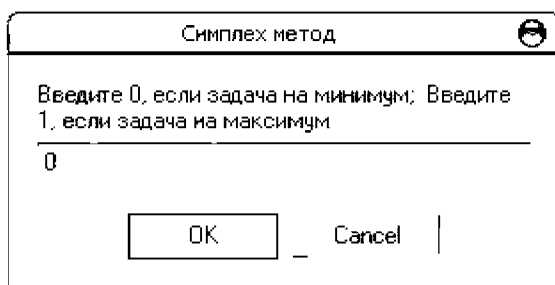


Рис. П2.9. Выбор минимизации или максимизации целевой функции

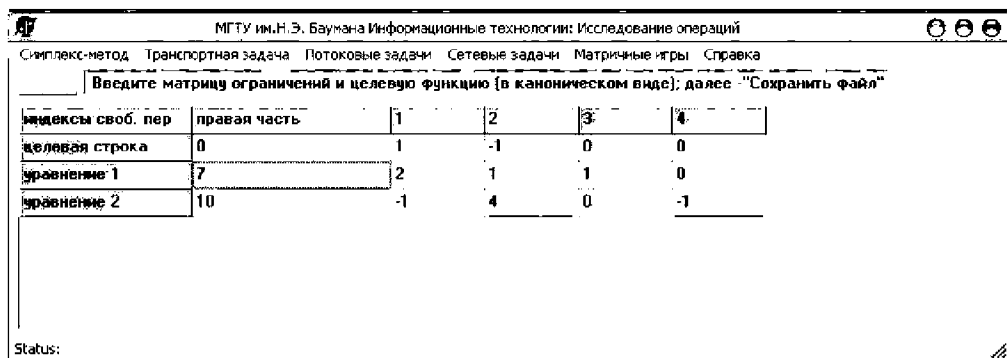


Рис. П2.10. Окно основной программы с введенными исходными данными

В предложенную форму вводятся данные: в строке «целевая функция» ввести коэффициенты целевой функции, в матрицу ограничений ввести коэффициенты системы ограничений, начиная со столбца свободных членов, при вводе в качестве разделителя дробной части использовать «,».

Теперь всю информацию можно сохранить в текстовом файле с расширением `.flp`, для чего нужно выбрать пункт подменю «Сохранить файл» — появится диалоговое окно сохранения файла (рис. П2.11).

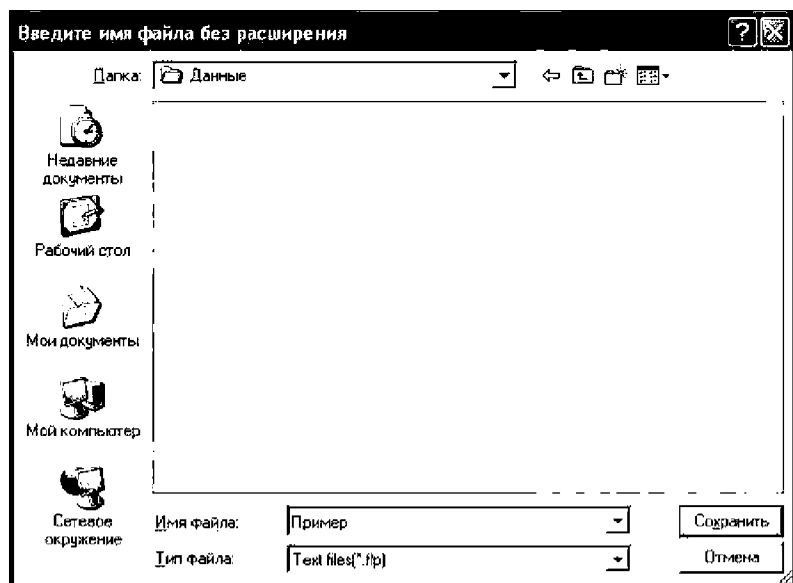


Рис. П2.11. Диалоговое окно сохранения файла данных

Откроем сохраненный файл с помощью подпункта «Открыть файл», предварительно выбрав пункт меню «Симплекс-метод» (рис. П2.12).

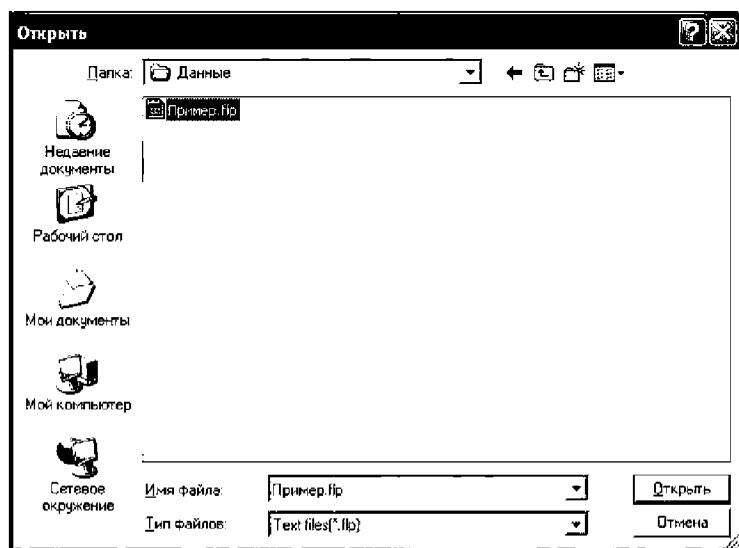


Рис. П2.12. Диалоговое окно открытия файла данных

Теперь можно приступить к решению задачи.

Замечание. Задачу можно решать и без предварительного сохранения данных в текстовом файле, после того как они интерактивно введены в программу.

Решение задачи в программе MatProg разбито на два этапа: поиск опорного (допустимого) решения и поиск оптимального решения (после того, как найдено опорное решение).

Для поиска опорного решения служит пункт подменю «Поиск опорного решения», при выборе которого для описанных в примере данных симплекс-таблица принимает вид, представленный на рис. П2.13.

МГТУ им.Н.Э. Баумана Информационные технологии: Исследование операций				
Симплекс-метод Транспортная задача Поточковые задачи Сетевые задачи Матричные игры Справка				
Опорное решение получено исходная система в каноническом виде				
индексы	0	3	4	
0	1	-0.333333333	-0.333333333	
1	2	0.444444444	0.111111111	
2	3	0.111111111	-0.222222222	
Status:				

Рис. П2.13. Симплекс-таблица, соответствующая опорному решению

Следует отметить, что в процессе решения выводятся только те столбцы симплекс-таблицы, которые соответствуют свободным переменным.

Дальнейшее решение задачи можно провести двумя способами: либо сразу получить оптимальное решение, выбрав пункт подменю «Поиск оптимального решения», после чего окно программы будет иметь вид, представленный на рис. П2.14; либо выполнить поиск оптимального решения с пошаговым выводом в основное окно программы промежуточных результатов.

индексы	0	1	3
0	0	-3	-1
4	18	9	4
2	7	2	1

Рис. П2.14. Оптимальная симплекс-таблица

индексы	0	1	4
0	-2.5	-0.75	0.25
3	4.5	2.25	0.25
2	2.5	-0.25	-0.25

Продолжить - на следующую итерацию

Рис. П2.15. Первая итерация симплекс-метода

индексы	0	1	3
0	-7	-3	-1
4	18	9	4
2	7	2	1

Рис. П2.16. Вторая итерация симплекс-метода (оптимальная симплекс-таблица)

Промежуточные результаты решения задачи при пошаговом решении приведены на рис. П2.15, П2.16. Выделенная синим цветом ячейка соответствует генеральному элементу.

ТРАНСПОРТНАЯ ЗАДАЧА

Для решения транспортной задачи в программе MatProg служит пункт меню «Транспортная задача» (рис. ПЗ.1). Исходная задача предварительно приводится к сбалансированной. Методика решения задачи приводится в справке в подменю системы.

В системе все файлы данных должны быть созданы только программно средствами системы. Для транспортной задачи устанавливается фильтр: вводится свое расширение для файла данных .ftz. Файлы, созданные в блокноте, не открываются.

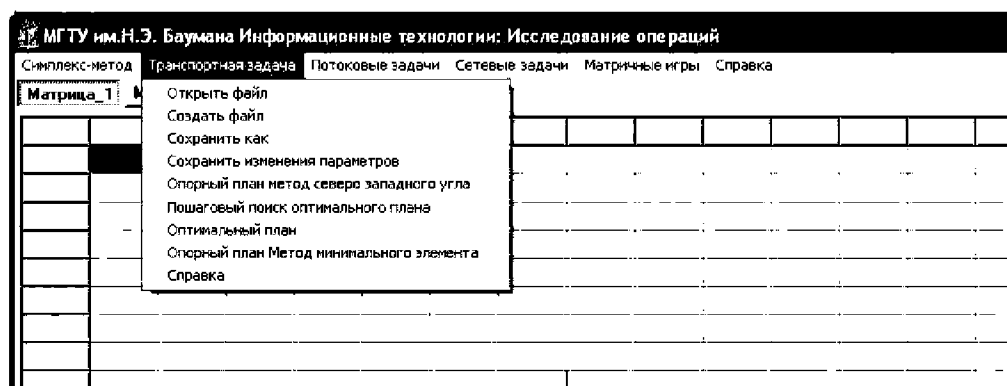


Рис. ПЗ.1. Подменю пункта основного меню «Транспортная задача»

Для создания нового файла выбрать пункт подменю «Создать файл». После этого в соответствующих диалоговых окнах необходимо ввести число поставщиков и потребителей. Для дальнейшей демонстрации работы программы при решении транспортной задачи рассмотрим конкретный числовой пример. Зададим число поставщиков, равное 2, и равное 3 число потребителей (рис. ПЗ.2).

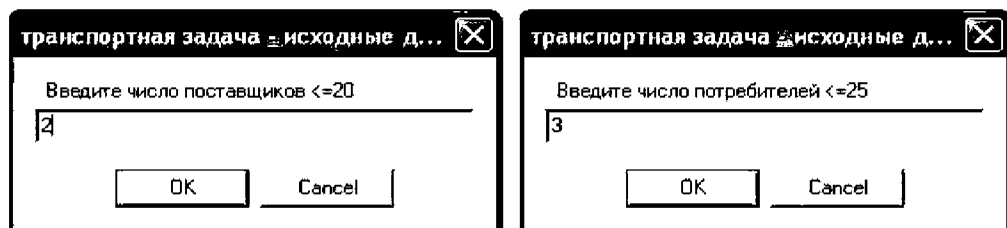


Рис. ПЗ.2. Ввод числа поставщиков и потребителей

После этого в главном окне программы нужно ввести матрицу стоимости перевозок, а также запасы и потребности. Окно программы с уже введенными исходными данными представлено на рис. ПЗ.3.

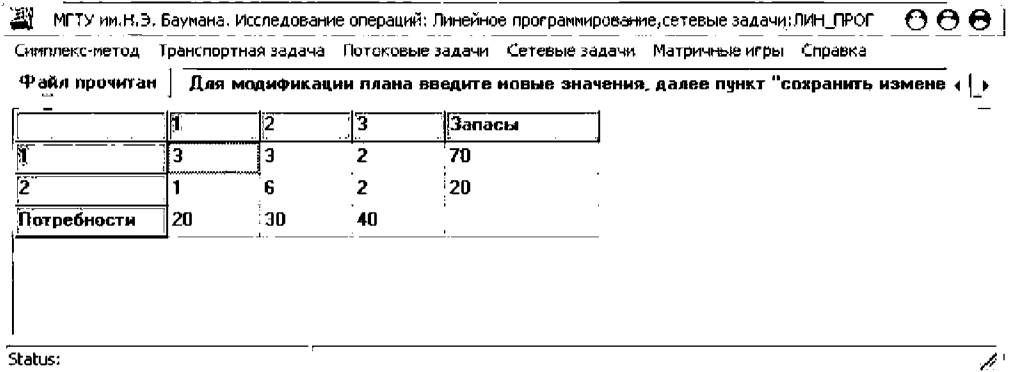


Рис. ПЗ.3. Главное окно программы с введенными исходными данными

Замечание. Данные вводятся для сбалансированной задачи! Если исходная задача не сбалансирована, ее надо предварительно привести к сбалансированной!

Введенные данные необходимо сохранить в файле с расширением .ftz, расширение задается в программе по умолчанию. Этот файл в дальнейшем можно будет снова использовать для проведения вычислений. Для этого служит подпункт меню «Сохранить как», в появившемся диалоговом окне необходимо задать имя файла и выбрать директорию, в которой он будет сохранен (рис. ПЗ.4).

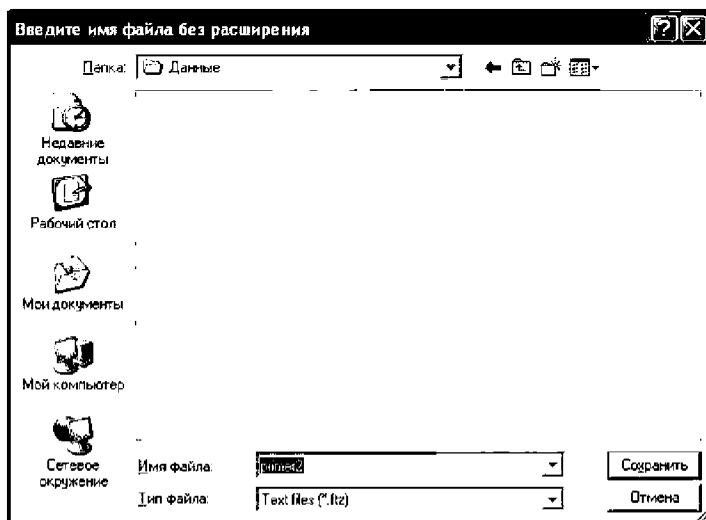


Рис. ПЗ.4. Диалоговое окно сохранения файла

Приступим непосредственно к решению задачи. Следующим шагом является поиск опорного плана. В программе для этого используются два метода: метод северо-западного угла и метод минимального элемента. Найдем опорный план методом северо-западного угла, выбрав соответствующий подпункт меню «Транспортная задача» (см. рис. ПЗ.1). Главное окно программы с найденным опорным планом показано на рис. ПЗ.5.

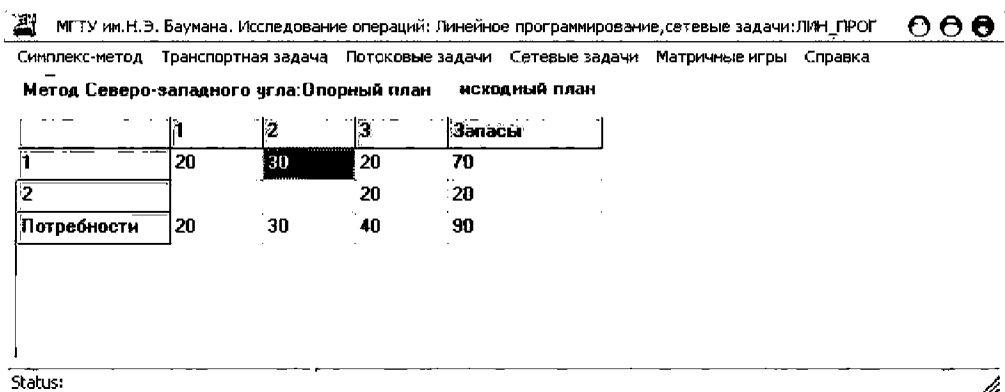


Рис. ПЗ.5. Опорный план, полученный методом северо-западного угла

Дальнейшее решение задачи можно провести двумя способами: 1) сразу получить оптимальное решение (оптимальный план перевозок), выбрав подпункт «Оптимальный план»; 2) пошагово провести вычисления с контролем на каждом шаге промежуточных данных — подпункт «Пошаговый поиск оптимального плана». Решим задачу вторым способом, промежуточные результаты приведены на рис. ПЗ.6 и ПЗ.7, оптимальный план — на рис. ПЗ.8.

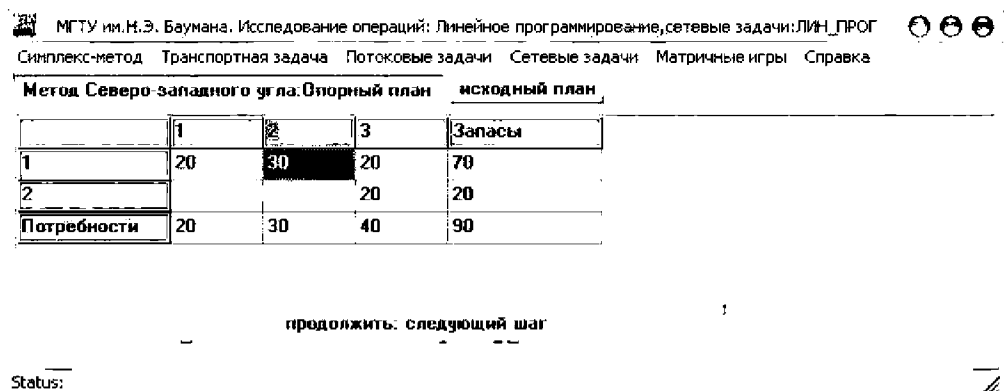



Рис. ПЗ.6. Исходные данные, программа переведена в режим пошагового решения

Для вывода в пошаговом режиме используются две закладки: на одной странице выводятся текущая матрица плана и значения потенциалов с отображением целевой клетки, на второй странице (при щелчке по второй закладке) выводится цикл пересчета.

 МГТУ им.Н.Э. Баумана. Исследование операций: Линейное программиро...


Симплекс-метод Транспортная задача Потокосые задачи Сетевые задачи
Матричные игры Справка

Текущий план: стоимость 190 итерация: 1 Матрица пересчета

	1>f=3	2>f=3	3>f=2	Запасы
1>u=0	20	30	20	70
2>u=0	X		20	20
Потребности	20	30	40	90

продолжить: следующий шаг

Status:

 МГТУ им.Н.Э. Баумана. Исследование операций: Линейное программиро...

Симплекс-метод Транспортная задача Потокосые задачи Сетевые задачи
Матричные игры Справка

Текущий план: стоимость 190 итерация: 1 Матрица пересчета

	1	3
1	20	20
2	X	20

Максимальная разность потенциалов 2

Status:

Рис. ПЗ.7. Первая итерация

МГТУ им.Н.Э. Баумана. Исследование операций: Линейное программирование, сетевые задачи: ЛИН_П...

Симплекс-метод Транспортная задача Потокосые задачи Сетевые задачи Матричные игры Справка

Оптимальный план: стоимость 190 итерация:2 Матрица разности стоимости и псевдостоимости 0

	1	2	3	Запасы
1		30	40	70
2	20		0	20
Потребности	20	30	40	90

Оптимальный план: стоимость 190 итерация:2

Status: _____

МГТУ им.Н.Э. Баумана. Исследование операций: Линейное программирование, сетевые задачи: ЛИН_П...

Симплекс-метод Транспортная задача Потокосые задачи Сетевые задачи Матричные игры Справка

Оптимальный план: стоимость 190 итерация:2 Матрица разности стоимости и псевдостоимости 0

	1	2	3
1	-2	0	0
2	0	-3	0

Максимальная разность потенциалов 0

Status: _____

Рис. П3.8. Вторая итерация, получен оптимальный план

Максимальная разность потенциалов, равная 0, является признаком того, что получен оптимальный план.

ЗАДАЧА О МАКСИМАЛЬНОМ ПОТОКЕ

Для решения задачи в программе MatProg служит пункт меню «Потоковые задачи» (рис. П4.1). Методика решения задачи приводится в справке в подменю системы.

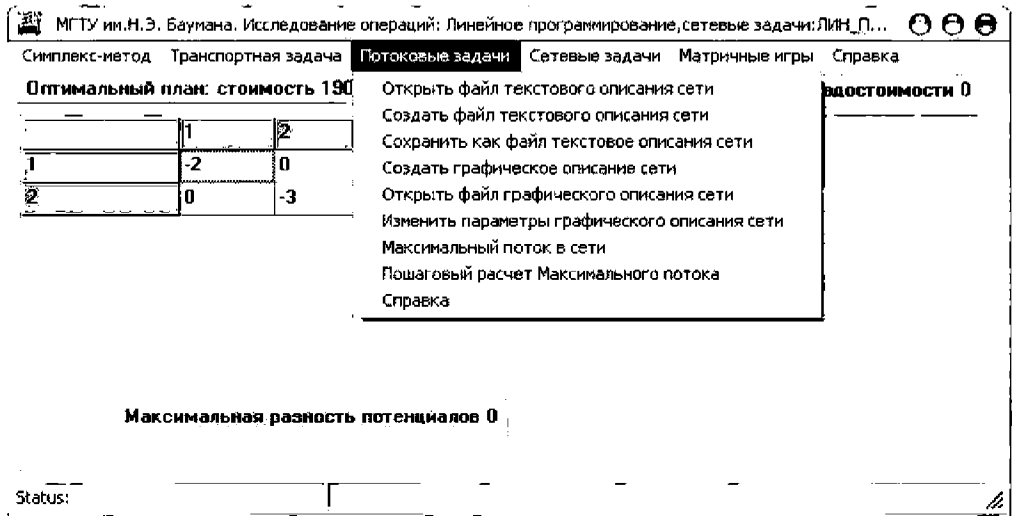


Рис. П4.1. Подменю пункта основного меню «Потоковые задачи»

Файлы данных должны быть созданы только программно средствами системы. Для рассматриваемой задачи устанавливается фильтр: вводится свое расширение для файла данных .fmp. Файлы, созданные в блокноте, не открываются.

Для этого необходимо предварительно по математическому описанию задачи создать описание графа сети. Для описания графа сети используется представление в виде списка ребер графа: каждая ветвь задается номерами начального и конечного узлов, пропускной способности ветви. Описание графа сети: задать число n вершин и число m ветвей графа; пронумеровать вершины графа, где номер вершины — целое число, вершины задаются последовательностью целых чисел: $0, 1, \dots, n$; задать описание ветви: начальный узел, конечный узел, пропускная способность ветви; задать источник — сток.

Создать файл можно в двух режимах: в текстовом и в графическом. Во втором случае текстовое описание создается по умолчанию. В связи с наглядностью описания и возможностями графического отображения сети результатов моделирования (минимальный разрез и два множества вершин) рассмотрим графический режим.

Выбираем пункт подменю «Создать графическое описание сети», после чего открывается окно графического редактора (рис. П4.2).

В этом редакторе по умолчанию выбран режим создания и нумерации вершин графа. Вершины отображаются и нумеруются по щелчку левой кнопки мыши. При нажатии кнопки «вершина» этот режим переключается (сбрасывается или устанавливается), и нажатии кнопки «линия» осуществляется ввод ветвей графа и их параметров и отображение в редакторе (рисование).

Редактор позволяет сохранять файл или его модифицировать.

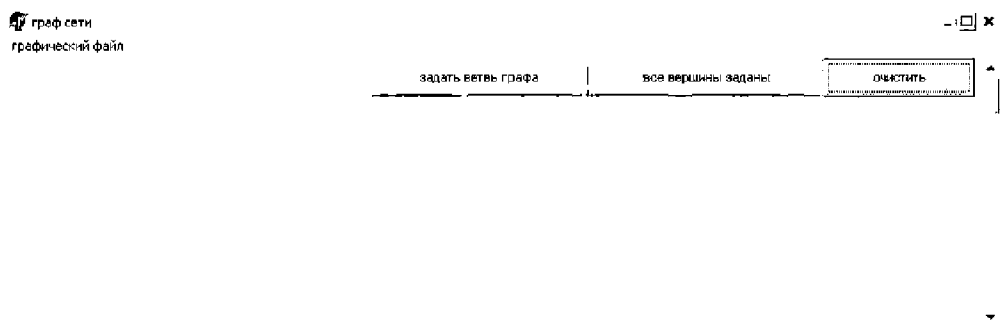


Рис. П4.2. Окно для графического «формирования» графа

Создание графа начинается вводом всех вершин графа (нажатием левой кнопки мыши), при этом осуществляются их нумерация и подсчет — пример с пятью вершинами приведен на рис. П4.3.

После ввода всех вершин необходимо перейти в режим рисования ветвей графа. Для этого сбрасывается режим рисования вершин (кнопка «вершина»), после чего выводится окно сообщения «число вершин графа».

После завершения ввода всех вершин графа при нажатии кнопки «линия» осуществляется переход к рисованию ветвей графа. Каждое ребро вводится нажатием кнопки «линия», после чего появляется диалоговое окно задания параметров ветви (рис. П4.4).

После ввода всех ветвей в редакторе создается картинка графа (рис. П4.5).

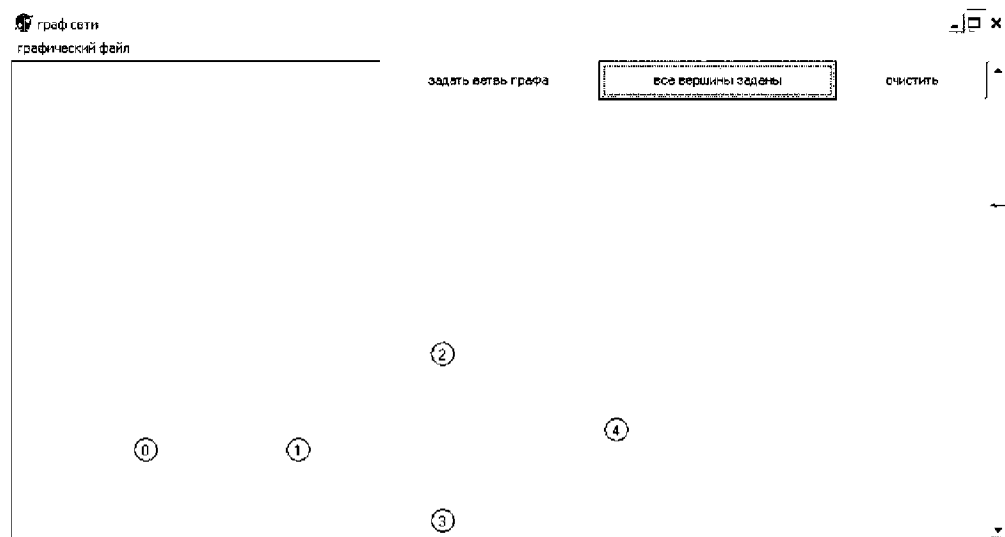


Рис. П4.3. Окно редактора с введенными вершинами

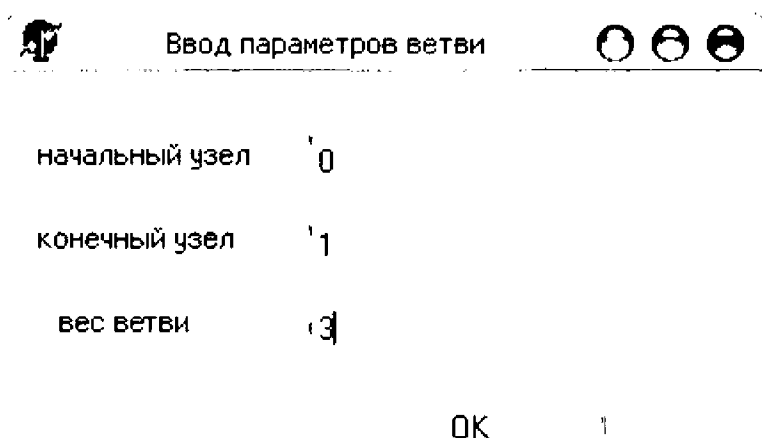


Рис. П4.4. Окно ввода параметров ветви

В программе вершина, соответствующая истоку, выделяется синим цветом, стоку — красным. Эти вершины вводятся в диалоге после вызова пункта меню редактора «сохранить файл» (рис. П4.6), после чего появляется окно для сохранения файла (рис. П4.7).

По графическому описанию в системе создается также текстовое описание сети в виде (рис. П4.8).

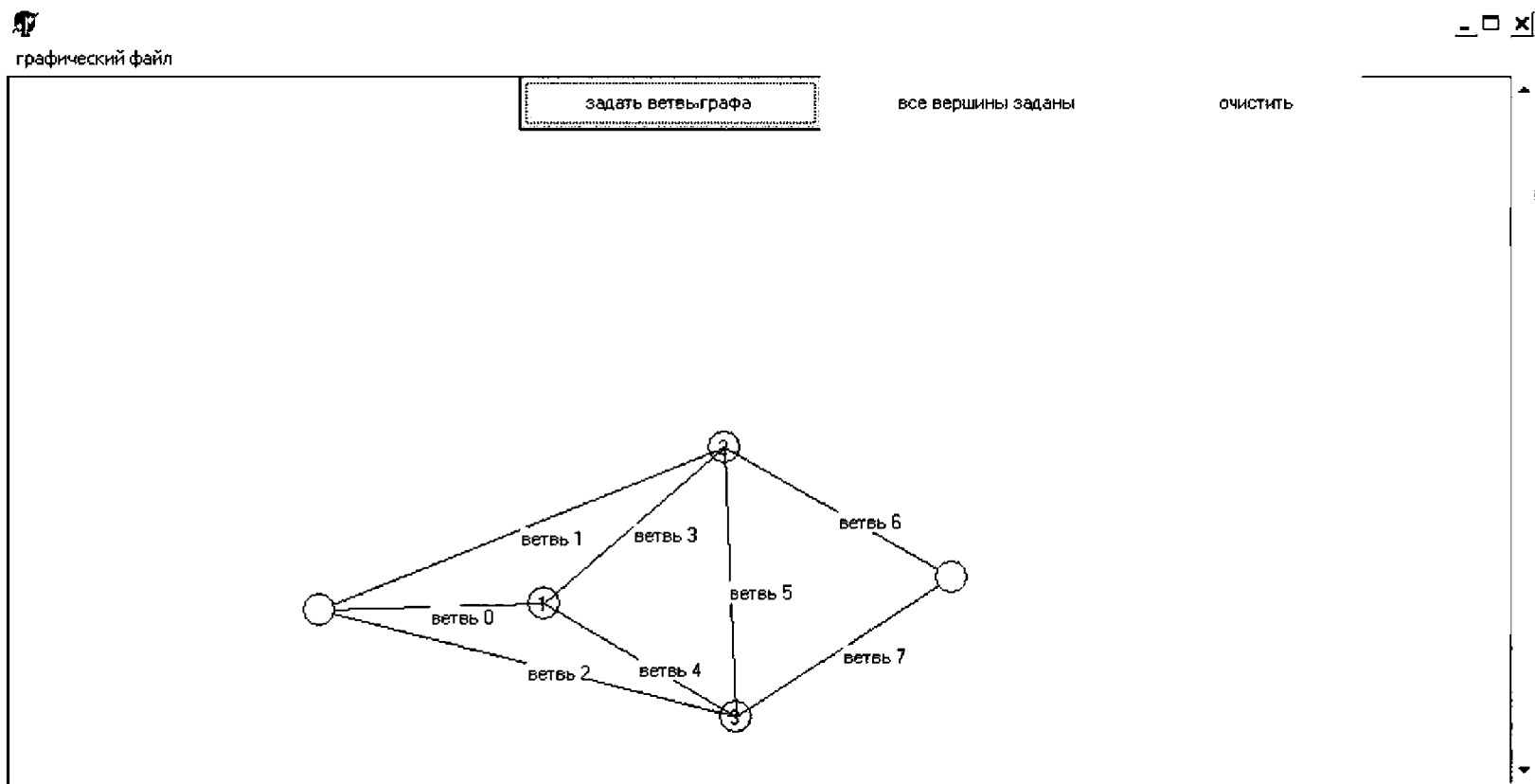


Рис. П4.5. Окно редактора с изображением графа

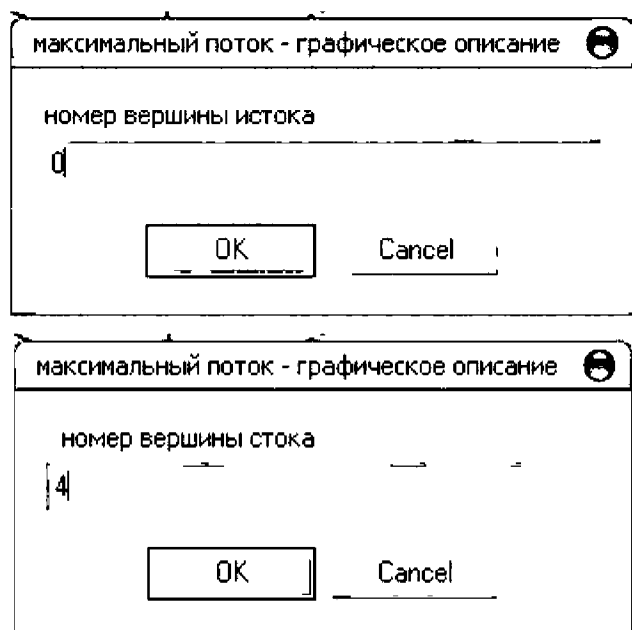


Рис. П4.6

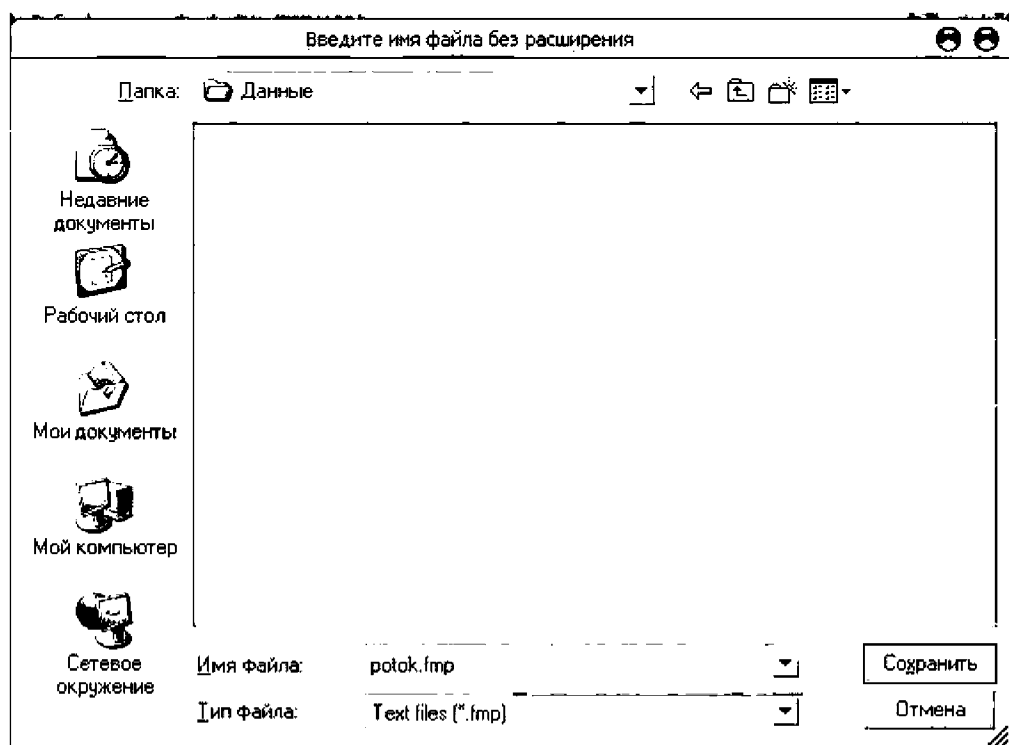


Рис. П4.7. Диалоговое окно сохранения файла

Источники	Начальный запас	Конечный запас	Пропускная способность ветви A	Поток F	C _i
0	0	1	3	0	3
1	0	2	2	0	2
2	0	3	2	0	2
3	1	2	4	0	4
4	1	3	5	0	5
5	2	3	3	0	3
6	2	4	6	0	6
7	3	4	4	0	4
Максимум	0	4			

N вершин графа	Вершины графа	N доп. путей	Дополняющие пути	Дополняющий поток
0				
1	1			
2	2			
3	3			
4	4			

Рис. П4.8. Окно программы с текстовым описанием задачи

После сохранения нового файла необходимо перейти к моделированию (расчету максимального потока и минимального разреза).

Вычисление максимального потока и минимального разреза можно выполнить в двух режимах. В первом режиме (пункт подменю «Максимальный поток в сети») сразу выводится результат решения задачи, отображаются окончательный результат и «протокол» моделирования потока всех итераций расчета; во втором режиме (пункт подменю «Пошаговый расчет максимального потока») выполняется моделирование задачи в пошаговом режиме.

Результаты моделирования потока в пошаговом режиме представлены на рис. П4.9—П4.11: на каждом шаге поиска дополняющего пути и потока по нему выводится последовательность вершин пути и поток в нем, для каждой ветви выводится величина потока и остаточная пропускная способность ветви.

Конечные результаты расчета приведены на рис. П4.12, где показаны ветви, составляющие разрез, множества вершин A и B, лежащих по разные стороны от разреза, максимальный поток по каждой ветви, причем ветви, входящие в разрез будут насыщенными.

На рис. П4.13 приведен граф (в программе ветви разреза выделены красным цветом).

Замечание. Программа MatProg позволяет решать также сетевые задачи, игры в чистых стратегиях, системы линейных урав-

нений, вычислять коэффициенты линейной регрессии для чего служат соответствующие пункты меню. Они здесь не рассматриваются.

Синплекс-метод Транспортная задача Потокотые задачи Сетевые задачи Матричные игры Справка

Расчет максимального потока: итерация1 описание каждой ветви графа, вершины истока , стока

N ветвей	Потокотый узел	Конецный узел	Пропускная способность ребра R	Поток F	Знач
0	0	1	3	0	3
1	0	2	2	2	0
2	0	3	2	0	2
3	1	2	4	0	4
4	1	3	5	0	5
5	2	3	3	0	3
6	2	4	6	2	4
7	3	4	4	0	4
исток-сток	0	4			
Вершины графа					
0	0	1	Дополняющие пути		Дополняющие
1	1		4 > 2 > 0 >		2
2	2				
3	3				
4	4				

Status:

Рис. П4.9. Первая итерация

Синплекс-метод Транспортная задача Потокотые задачи Сетевые задачи Матричные игры Справка

Расчет максимального потока: итерация2 описание каждой ветви графа, вершины истока , стока

N ветвей	Потокотый узел	Конецный узел	Пропускная способность ребра R	Поток F	Знач
0	0	1	3	0	3
1	0	2	2	2	0
2	0	3	2	2	0
3	1	2	4	0	4
4	1	3	5	0	5
5	2	3	3	0	3
6	2	4	6	2	4
7	3	4	4	2	2
исток-сток	0	4			
Вершины графа					
0	0	1	Дополняющие пути		Дополняющие
1	1	2	4 > 2 > 0 >		2
2	2		4 > 3 > 0 >		2
3	3				
4	4				

Status:

Рис. П4.10. Вторая итерация

Симплекс-метод Транспортная задача Потокотовые задачи Сетевые задачи Матричные игры Справка					
Расчет максимального потока, итерация 3 описание каждой ветви графа, вершины истока, стока					
N ветвей	Начальный запас	Конечный запас	Пропускная способность ребра R	Поток F	Баланс
0	0	1	3	3	0
1	0	2	2	2	0
2	0	3	2	2	0
3	1	2	4	3	1
4	1	3	5	0	5
5	2	3	3	0	3
6	2	4	6	5	1
7	3	4	4	2	2
Мак.сток	0	4			
N вершин графа	Вершины графа	N доп. путей	Дополняющие пути	Дополняющий поток	
0	0	1	4 > 2 > 0 >	2	
1	1	2	4 > 3 > 0 >	2	
2	2	3	4 > 2 > 1 > 0 >	3	
3	3				
4	4				
Status:					

Рис. П4.11. Третья итерация

Симплекс-метод Транспортная задача Потокотовые задачи Сетевые задачи Матричные игры Справка					
Максимальный поток: 7 описание каждой ветви графа, вершины истока, стока					
N ветвей	Начальный запас	Конечный запас	Пропускная способность ребра R	Поток F	Баланс
0	0	1	3	3	0
1	0	2	2	2	0
2	0	3	2	2	0
3	1	2	4	3	1
4	1	3	5	0	5
5	2	3	3	0	3
6	2	4	6	5	1
7	3	4	4	2	2
Мак.сток	0	4			
N вершин графа	Вершины графа	N доп. путей	Дополняющие пути	Дополняющий поток	
0	0	1	4 > 2 > 0 >	2	
1	1	2	4 > 3 > 0 >	2	
2	2	3	4 > 2 > 1 > 0 >	3	
3	3		Максимальный поток:	7	
4	4		Множество A(исток): 0		
			Множество B(сток): 1 2 3 4		
			ветви разреза: 0 1 2		
Status:					

Рис. П4.12. Оптимальное решение

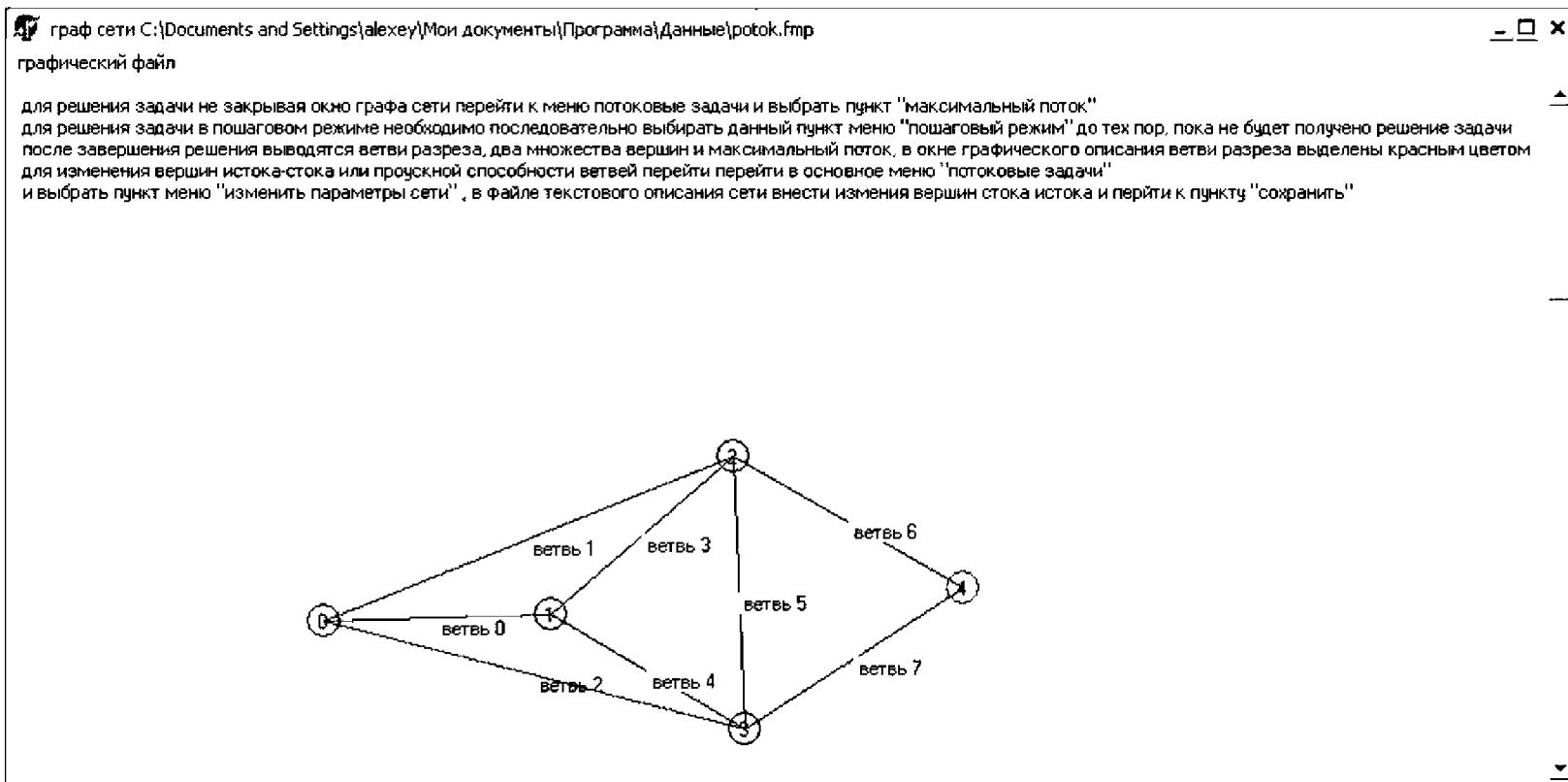


Рис. П4.13. Граф сети, ветви разреза

Поиск в графах И-ИЛИ — решатель

Решатель — инструментальное средство для создания баз знаний и системы поиска решений: оболочка для создания проблемно ориентированных систем, реализующие решение задач планирования вычислений при логическом синтезе программ или вывод в экспертных системах в различных предметных областях путем настройки на соответствующую базу знаний.

Математическая модель предметной области представляет собой совокупность отношений, связанных между собой общими атрибутами. Схеме модели предметной области соответствует семантическая сеть, описываемая графом И-ИЛИ, представляющая композицию схем отношений, составляющих модель.

База сети формируется в виде списка в динамической памяти, который представлен списком элементарных модулей. Для описания модуля сети используется спецификационный интерфейс, информационная часть которого является структурой следующего вида: первый аргумент — выходной атрибут, второй — список входных атрибутов, третий — имя модуля. Число звеньев в базе равно числу модулей. Включение нового модуля соответствует включению нового звена в базу. Построение дерева решения задачи рассматривается как решение задачи поиска пути на графе И-ИЛИ, ведущего из начального состояния в конечное — целевое состояние.

Как альтернатива — база знаний экспертной системы, описываемая правилами продукций: если \langle условия, связанные связкой И, выполняются \rangle , то \langle заключение \rangle .

Пример. Если $v_1, v_2, v_3, v_4, \dots, v_n$ — простые высказывания, составляющие словарь системы, то возможен следующий набор правил:

Если $v_1 \& v_2$, то v_3 ;
Если $v_4 \& v_2$, то v_3 ;
Если $v_5 \& v_6 \& v_7$, то v_2 ;
Если $v_4 \& v_7$, то v_1 ;

.....

и так далее.

Этот набор правил продукций описывается графом И-ИЛИ, на котором выполняется вывод.

Пример. Принцип описания базы.

Рассматриваются два режима: создание базы описания сети (описание графа И-ИЛИ) и поиск в сети.

1. Создание файла базы описания сети.

Все файлы данных должны быть созданы только программно средствами системы. Для решателя устанавливается фильтр: вводится свое расширение для файла данных «.fps». Файлы, созданные в блокноте, не открываются.

Для решения задачи необходимо по математическому описанию создать описание графа сети — список модулей сети. Для каждого модуля задается имя модуля, имя выходной вершины, количество входных вершин и последовательно вводится список этих вершин.

Имена модулей — уникальны (или имена правил), по умолчанию имена модулей начинаются с символа m, следующие символы задают номер модуля: m1, m2, m3, ..., m35 и т. д. Число модулей в этой версии ограничено значением 50. Имена вершин — строка длиной до 3 символов: a, a0, v1, ..., v32 и т. д.

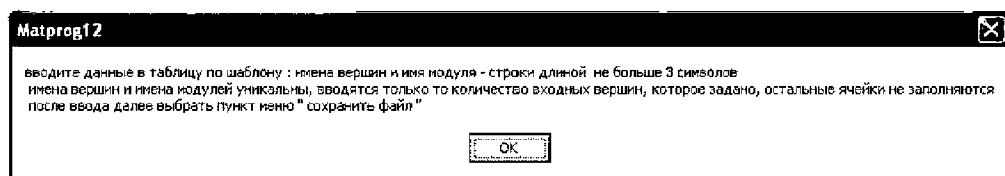
Файл описания сети создается программно: имя файла программно задается с расширением — «.fps». Полное имя файла сети: < имя > .fps.

Инструкция создания файла базы описания сети.

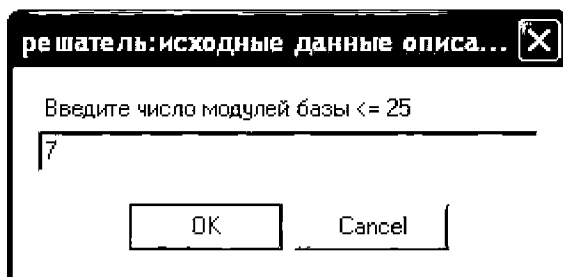
Выбрать пункт меню «решатель»;

Далее пункт подменю — «создать файл»;

Далее отображается окно «подсказки» — help.



После закрытия этого окна выводится диалоговое окно для запроса, в которое вводится число модулей сети:



В данной версии число модулей ограничено 50.

После закрытия окна запроса отображается форма для ввода данных, число строк в которой равно числу модулей сети. Строка формы предназначена для описания модуля сети: первый стол-

бец — имя модуля, второй — имя выходной вершины модуля, третий — количество входных вершин, в следующие столбцы — имена входных вершин модуля (число вершин в данной версии ограничено 5 значениями). Ниже пример формы для ввода 7 модулей.

поиск в графах и-или

Симплекс-метод Транспортная задача Потокосые задачи Решатель Сетевые задачи Матричные игры Справка

для задания сети (графа и-или) вводится информация описания модулей сети: имя модуля, выходная вершина, список

номер	имя модуля	выходная вершина	количество входов	1 вершина	2 вершина	3 вершина	4 вершина	5 вершина
1								
2								
3								
4								
5								
6								
7								

Status:

Вводится последовательно информация о каждом модуле сети: каждая строка формы заполняется по шаблону описания модуля сети: имя модуля, имя выходной вершины, количество входных вершин, имена вершин; имя модуля и имена вершин — строка длиной до 3 символов.

После завершения ввода необходимо сохранить созданное описание сети: выбрать пункт меню «сохранить файл описания сети». Далее ввести имя файла без расширения. (Файл по умолчанию сохраняется с расширением «.fps».) Например, для данного случая: dat0.fps.

Пример заполненной формы для описания сети с 14 модулями:

поиск в графах и-или

Симплекс-метод Транспортная задача Потокосые задачи Решатель Сетевые задачи Матричные игры Справка

описание сети (графа и-или): имя модуля, выходная вершина, список входных вершин:

номер	имя модуля	вершина вых	количество в	имя вершин	имя вершин	имя вершин	имя вершин	имя вершин
1	m1	v	2	a	b			
2	m2	a	1	c				
3	m3	v0	2	b	f			
4	m4	b	2	c	d			
5	m5	b	2	d	x			
6	m6	a	1	y				
7	m7	f	2	d	z			
8	m8	d	2	a1	b1			
9	m9	b1	2	y1	y2			
10	m10	b1	2	y2	z			
11	m11	c	3	y3	y4	a1		
12	m12	v0	2	x	y5			
13	m13	d	2	y6	z			
14	m14	y5	2	z	y6			

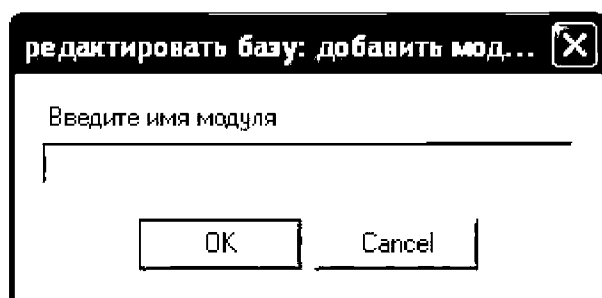
Status:

Просмотр (чтение) базы файла описания сети:

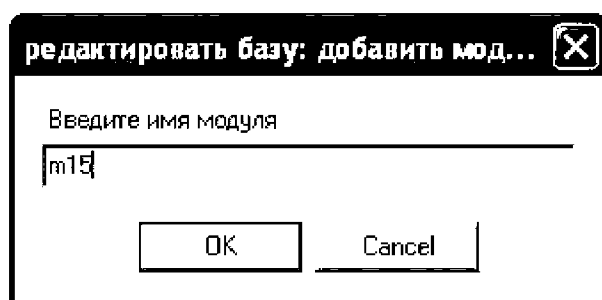
выбрать пункт подменю «открыть файл» и открыть файл с расширением «.fps». После чего выводится вышерассмотренное окно описания сети.

Редактирование базы сети. Описание сети можно дополнить: добавить новые модули сети.

Выбрать пункт меню «решатель». Далее пункт подменю «открыть файл». После отображения формы описания базы выбрать пункт подменю — «редактировать базу». После этого отображается последовательность форм для ввода по шаблону параметров модуля сети. В первую форму требуется ввести имя добавляемого модуля:



Заполненное окно:

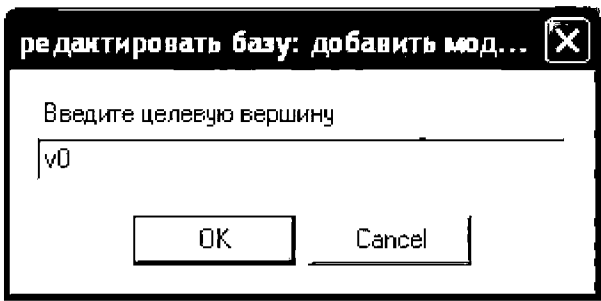


Далее вводятся имена вершин в соответствующих последовательных окнах.

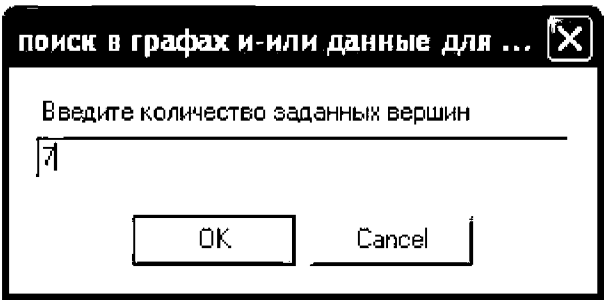
После добавления одного или нескольких модулей необходимо сохранить модифицированный файл: выбрать пункт меню «сохранить файл» и далее сохранить файл под тем же или другим именем.

2. Поиск решения: вывод плана решения: определение последовательности вызовов модулей сети, формирующих дерево решения.

Выбрать пункт подменю « поиск».
Ввести имя целевой вершины:



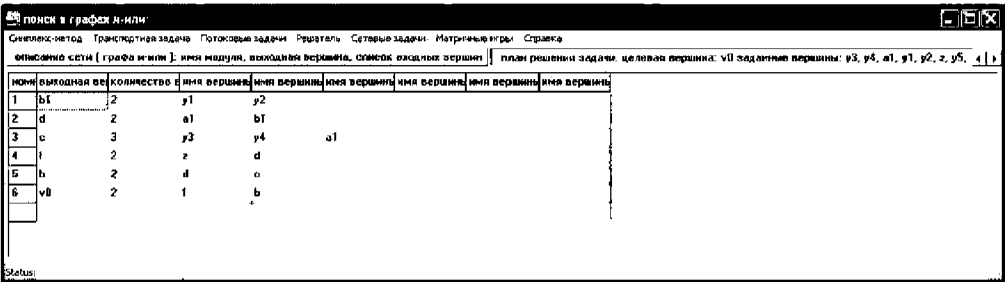
В следующем окне ввести количество заданных (известных) вершин:



В аналогичных диалоговых окнах последовательно вводятся имена заданных вершин (имена, совпадающие до символа с именами вершин графа).

Далее выполняется поиск (логический вывод в базе) и выводиться план решения задачи — цепочка последовательности выполнения модулей, в верхней строке выводится список известных вершин и целевая вершина.

Решение: v0 — целевая вершина, [y3, y4, a1, y1, y2, z, y5] — список заданных вершин:



Решение: v_0 — целевая вершина, $[y_1, y_2, z, a_1, c, y_4, y_3]$ — список заданных вершин:

пакет в графах и-или

Синтаксис: метод [графа и-или] имя модуля, имя входной вершины, список входных вершин, [план доверия задачи, целевая вершина: v_0 заданные вершины: $y_1, y_2, z, a_1, c, y_4, y_3$, и-и]

номер	имеющийся в	количество	имя вершины	имя вершины	имя вершины	имя вершины	имя вершины	имя вершины
1	b1	2	y1	y2				
2	d	2	a1	b1				
3	f	2	z	d				
4	b	2	c	d				
5	v0	2	f	b				

Результат:

ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ, ЗАДАЧА О РЮКЗАКЕ

Для решения задачи о рюкзаке служит программа `dynprog.exe`, которая не имеет графического интерфейса и запускается двойным щелчком левой кнопкой мыши (можно также ее запустить из командного окна MS DOS) — при этом на экране «вспыхнет» командное окно MS DOS и в директории, где находится программа, будет создан файл с результатами ее работы. Для корректной работы программы необходимо, чтобы в директории вместе с программой лежал текстовый файл `input.txt` (созданный в Блокноте), в котором записаны исходные данные задачи. Вид файла следующий:

```
9 4
1 3 2 2
2 3 2 5
0 0 0 0
```

Здесь в первой строке первое число (9) — объем рюкзака, второе число (4) — число переменных (число типов предметов), вторая строка содержит объемы предметов, третья — их веса, четвертая — ограничения на максимальное количество предметов каждого вида. Если в четвертой строке в позиции для соответствующего вида предмета записан 0, это означает, что ограничений на число предметов данного вида нет. В приведенном примере для всех предметов нет ограничений на их количество.

Замечание. Все числа в файле должны быть целыми! Если записаны дробные числа, то программа будет использовать только целые их части (произойдет отсечение дробных частей).

Результат работы программы записывается (в директории, где находится программа) в файл с именем `dinprog_results.txt`. Пример формата файла для указанных данных приведен ниже:

Исходные данные задачи:

Объем рюкзака - 9

Матрица-строка объемов предметов - [1 3 2 2]

Матрица-строка весов предметов - [2 3 2 5]

Матрица-строка ограничений на максимальное число предметов -

[9 3 4 4]

Таблицы, получаемые при решении.

Таблица 1 -----

Объем, V	Вес, p	Значения, x1
0	0	0
1	2	1
2	5	0
3	7	1
4	10	0
5	12	1
6	15	0
7	17	1
8	20	0
9	22	1

Таблица 2 -----

Объем, V	Вес, p	Значения, x2
0	0	0
1	0	0
2	5	0
3	5	0
4	10	0
5	10	0
6	15	0
7	15	0
8	20	0
9	20	0

Таблица 3 -----

Объем, V	Вес, p	Значения, x3
0	0	0
1	0	0
2	5	0
3	5	0
4	10	0
5	10	0
6	15	0
7	15	0
8	20	0
9	20	0

Таблица 4

Объем, V	Вес, p	Значения, x_4
0	0	0
1	0	0
2	5	1
3	5	1
4	10	2
5	10	2
6	15	3
7	15	3
8	20	4
9	20	4

Оптимальные точки:

$x = [1\ 0\ 0\ 4]$

Оптимальное значение целевой функции:

$f_{\text{опт}} = 22$

Замечание. Хотя изначально ограничения на максимальное число предметов каждого вида не заданы, в программе автоматически вычисляется, сколько по максимуму предметов каждого вида можно положить в заданный объем, и эти данные записываются в четвертой строке выходного файла («Матрица-строка ограничений на максимальное число предметов — [9 3 4 4]»).

Для решения различного рода математических задач, задач моделирования существует несколько математических пакетов. Одним из широко распространенных пакетов является пакет MATLAB. Для решения задачи о рюкзаке в пакете MATLAB написана специальная функция `dynprog.m`, которая также находится на лазерном диске.

Краткие сведения о математическом пакете MATLAB. Математический пакет MATLAB — мощный программный продукт, который позволяет проводить самые разнообразные вычисления: от простого умножения и сложения до моделирования сложных технических систем. Он имеет свой собственный язык — язык написания *m*-файлов (чем-то напоминающий язык Си). На этом языке пользователь может писать свои функции, определять и сохранять последовательности различных вычислений и т. д.

Следует сказать, что математический пакет MATLAB — дорогостоящий продукт (лицензия стоит несколько тысяч долларов), но

ознакомительную версию можно бесплатно скачать с сайта компании-разработчика: <http://www.mathworks.com>.

Внешний вид главного окна пакета MATLAB версии R14 изображен на рис. П5.1.

Окно разделено на несколько частей. Самую большую занимает область Command Window (однако вид отображения окна MATLAB можно настроить по своему усмотрению). Command Window (командное окно) предназначено для ввода команд, вызова функций, просмотра результатов вычислений и т. п. В левой верхней части расположена область Current Directory (текущая директория). В ней отображается содержимое текущей активной директории. Полный путь к ней можно посмотреть под главным меню программы. Под Current Directory находится область Command History (история команд), где ведется запись всех вводимых в Command Window команд и функций. Более подробную справочную информацию о системе MATLAB можно найти на сайте www.exponenta.ru.

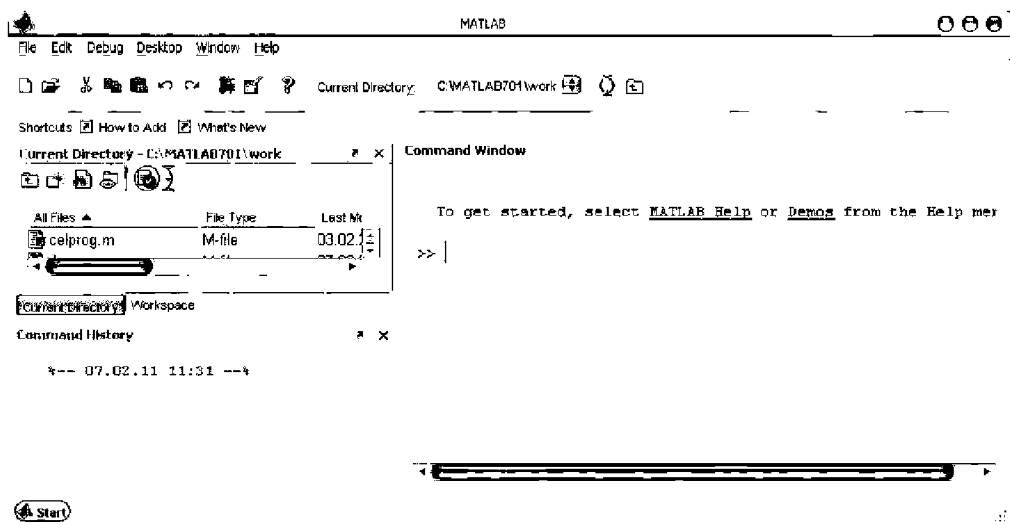


Рис. П5.1. Главное окно пакета MATLAB

Для вызова функций нам необходимо знать, как в MATLAB задаются матрицы. Чтобы задать, например, матрицу-строку $a = [1 \ 2 \ 3]$, нужно перейти в Command Window и набрать точно такую же запись (рис. П5.2). Если после выражения поставить знак «;», то результат его выполнения не будет выводиться на экран. Если знак «;» не ставить, то элементы матрицы отобразятся в Command Window (см. рис. П5.2).

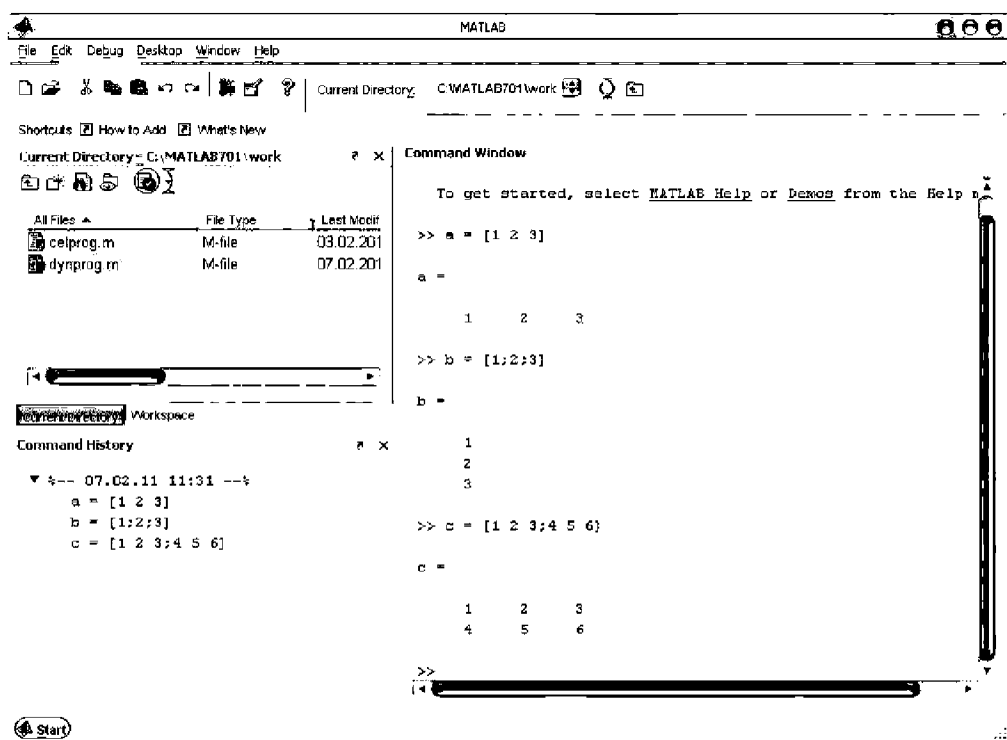


Рис. П5.2. Задание матрицы-строки, матрицы-столбца и прямоугольной матрицы

Матрица-столбец задается аналогично, только элементы разделяются между собой не пробелами, а знаком «;» (см. рис. П5.2).

Как уже было сказано, в командном окне можно не только инициализировать переменные, но и совершать над ними всевозможные операции, а также вызывать функции этих аргументов.

Чтобы получить некоторую справочную информацию о выполняемых функцией действиях, об ее аргументах и выходных параметрах, надо набрать `help function_name`, где `function_name` — имя конкретной функции. Например, для решения задачи динамического программирования мы будем использовать имеющуюся на диске функцию `dynprog`. На рис. П5.3 показано, как получить о ней справочную информацию.

Задаем исходные данные: объем рюкзака $V = 9$, объемы предметов $v = [1 \ 3 \ 2 \ 2]$ (всего четыре предмета), веса предметов $p = [2 \ 3 \ 2 \ 5]$, ограничения на максимальное количество предметов каждого вида $m = [0 \ 0 \ 0 \ 0]$ (нет ограничений) и вызываем функцию `dynprog` (рис. П5.4).

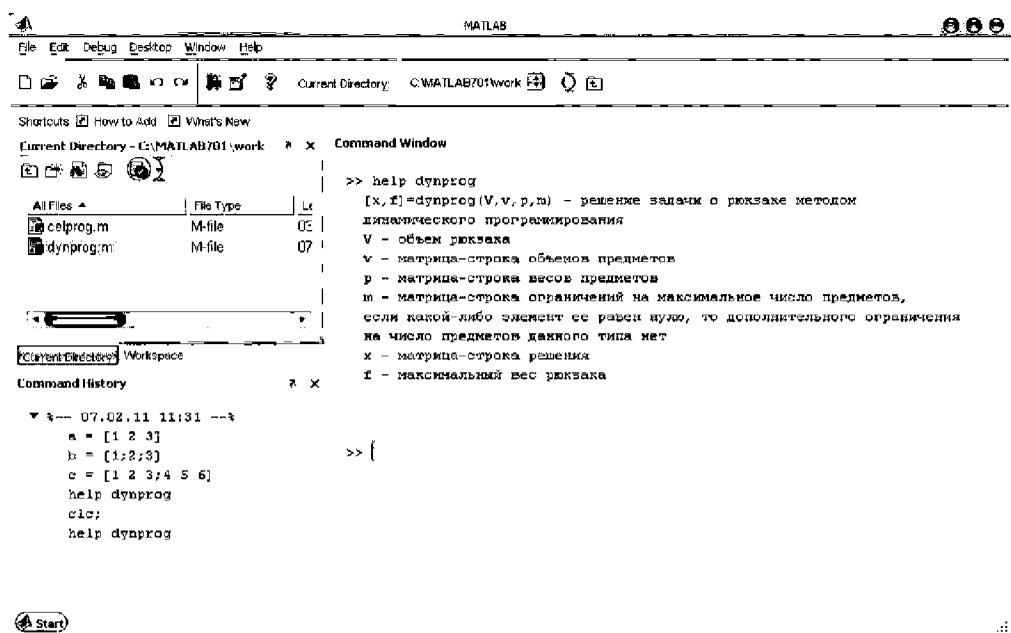


Рис. П5.3. Получение справочной информации о функции dynprog

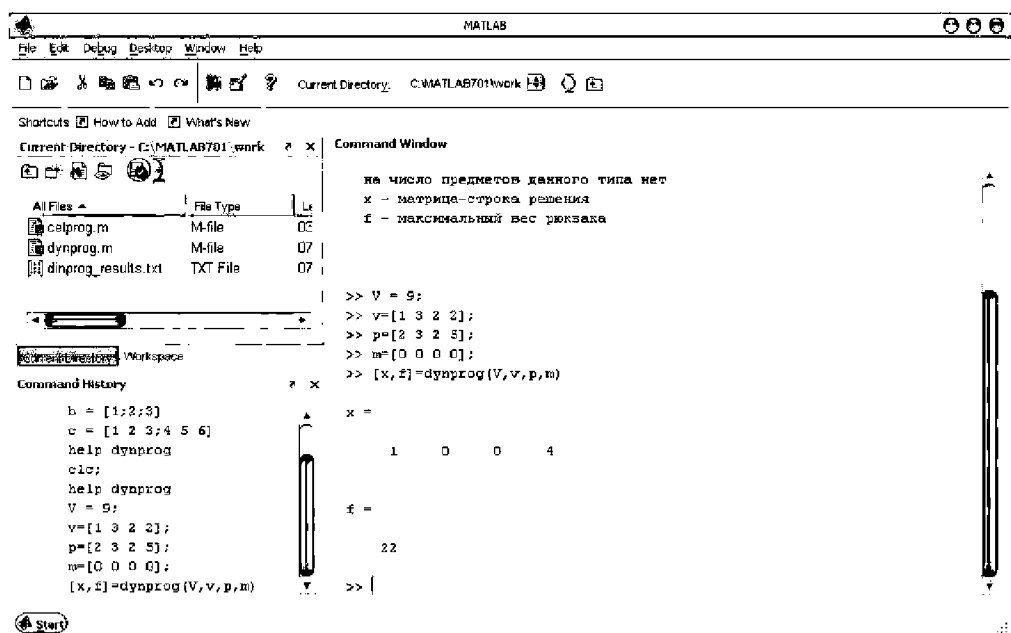


Рис. П5.4. Решение задачи динамического программирования с помощью функции dynprog

Замечание. Файл dynprog.m должен находиться в текущей директории.

На рис. П5.4 видно, что имеется одна оптимальная точка, оптимальное значение целевой функции равно 22.

Более подробный результат работы функции `dynprog` записывается в файл `dinprog_results.txt`, который сохраняется в текущей директории и может быть просмотрен с помощью текстового редактора WordPad. Содержимое этого файла аналогично содержимому приведенного ранее выходного файла программы `dynprog`.

ЦЕЛОЧИСЛЕННОЕ ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

Для решения задач целочисленного линейного программирования используется метод отсечения Гомори. Весь алгоритм также реализован в m-файле `celprog.m`, который нужно запускать в математическом пакете MATLAB. Подробный результат работы функции `celprog` записывается в файл `celprog_results.txt`, который сохраняется в текущей директории и может быть просмотрен с помощью текстового редактора WordPad.

Рассмотрим пример вызова функции `celprog` для решения задачи о рюкзаке по тем данным, которые изложены в предыдущем пункте: зададим объем рюкзака $b = 9$, объемы предметов $A = [1 \ 3 \ 2 \ 2]$ и их веса $c = [2 \ 3 \ 2 \ 5]$ (рис. П6.1).

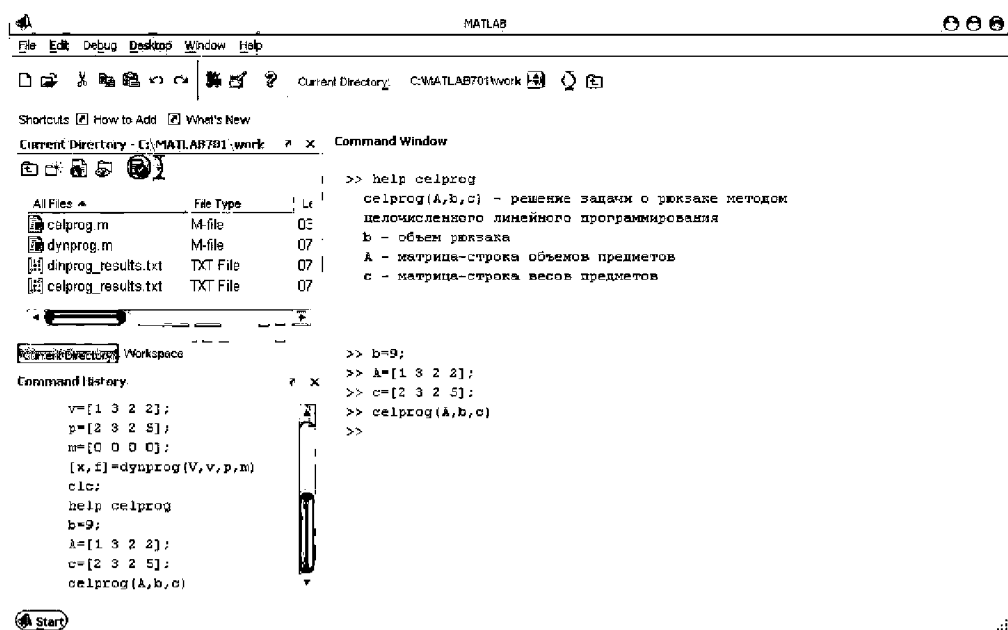


Рис. П6.1. Вызов функции `celprog`

Содержимое файла `celprog_results.txt` представлено ниже:

	b	x1	x2	x3	x4	x5
x4	4.5000	0.5000	1.5000	1.0000	1.0000	0.5000
f	22.5000	0.5000	4.5000	3.0000	0.0000	2.5000

Производящая строка:

$$0.5000 \cdot x_1 + 1.5000 \cdot x_2 + 1.0000 \cdot x_3 + 1.0000 \cdot x_4 + 0.5000 \cdot x_5 = 4.5000$$

Уравнение отсечения, построенное по производящей строке:

$$0.5000 \cdot x_1 + 0.5000 \cdot x_2 + 0.0000 \cdot x_3 + 0.0000 \cdot x_4 + 0.5000 \cdot x_5 \geq 0.5000$$

Симплекс-таблица с добавленным ограничением:

	b	x1	x2	x3	x4	x5	x6
x4	4.5000	0.5000	1.5000	1.0000	1.0000	0.5000	0.0000
x6	-0.5000	-0.5000	-0.5000	-0.0000	-0.0000	-0.5000	1.0000
f	22.5000	0.5000	4.5000	3.0000	0.0000	2.5000	0.0000

Симплекс-таблица с фиктивной целевой функцией, построенной по добавленному ограничению:

	b	x1	x2	x3	x4	x5	x6
x4	4.5000	0.5000	1.5000	1.0000	1.0000	0.5000	0.0000
x6	-0.5000	-0.5000	-0.5000	-0.0000	-0.0000	-0.5000	1.0000
f	22.5000	0.5000	4.5000	3.0000	0.0000	2.5000	0.0000
fi	-0.5000	-0.5000	-0.5000	0.0000	0.0000	-0.5000	1.0000

Оптимальная симплекс-таблица:

	b	x1	x2	x3	x4	x5	x6
x4	4.0000	0.0000	1.0000	1.0000	1.0000	0.0000	1.0000
x1	1.0000	1.0000	1.0000	0.0000	0.0000	1.0000	-2.0000
f	22.0000	0.0000	4.0000	3.0000	0.0000	2.0000	1.0000

Целочисленная оптимальная точка найдена: $x = [1 \ 0 \ 0 \ 4]$. Функция `celprog` не ищет другие альтернативные оптимумы (если они имеются).

Замечание. Функция `celprog` решает только задачи полностью целочисленного программирования — когда все переменные могут принимать только целочисленные значения. После того как в симплекс-таблицу добавляется новое ограничение-отсечения, базисное решение перестает быть допустимым. Дальше можно либо решать задачу двойственным симплекс-методом, либо ввести фиктивную целевую функцию (как делается в `celprog`) — это описано в соответствующих разделах линейного программирования методического пособия.

Примечание. Если работа какой-либо функции в командном окне MATLAB занимает очень длительное время и ее необходимо прервать, то это можно сделать, нажав комбинацию клавиш «Ctrl+C».

ПРИМЕР РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ДВОЙСТВЕННЫМ СИМПЛЕКС-МЕТОДОМ

Минимизировать функцию $z = 3x_1 + x_2$ при условиях:

$$\begin{aligned}x_1 + x_2 &\geq 1; \\ 2x_1 + 3x_2 &\geq 2; \\ x_1 &\geq 0; \quad x_2 \geq 0.\end{aligned}$$

Переходим к ограничениям-равенствам:

$$\begin{aligned}x_1 + x_2 - x_3 &= 1; \\ 2x_1 + 3x_2 - x_4 &= 2; \\ x_1 &\geq 0; \quad x_2 \geq 0.\end{aligned}$$

Чтобы переменные x_3 и x_4 стали базисными, умножим ограничения на (-1) . Получим

$$\begin{aligned}-x_1 - x_2 + x_3 &= -1; \\ -2x_1 - 3x_2 + x_4 &= -2.\end{aligned}$$

Целевую функцию z изменим таким образом, чтобы коэффициенты неизвестных в ней стали отрицательными. Запишем ее в виде

$$-3x_1 - x_2 \rightarrow \max.$$

Строим симплекс-таблицу (табл. П7.1).

Таблица П7.1

Исходная таблица

Базис	b_i	x_1	x_2	x_3	x_4
x_3	-1	-1	-1	1	0
x_4	-2	-1	-3	0	1
z	0	-3	-1	0	0

Максимальное значение $|b_i|$ находится в строке x_4 (ведущей строке). Переменную x_4 переводим в свободные.

Рассмотрим отношения отрицательных элементов в строке целевой функции z к соответствующим отрицательным элементам в

ведущей строке x_4 : получим $-3/-1$ и $-1/-3$. Выбираем наименьшее значение и свободную переменную x_2 переводим в базисные на место x_4 . Генеральным элементом будет элемент (2, 2). Методом Жордана выполним операцию замещения. Получим следующую симплекс-таблицу (табл. П7.2).

Таблица П7.2

Первая итерация

Базис	b_i	x_1	x_2	x_3	x_4
x_3	$-1/3$	$-1/3$	0	1	$-1/3$
x_2	$2/3$	$2/3$	1	0	$-1/3$
z	$-2/3$	$-7/3$	0	0	$-1/3$

В этой таблице есть отрицательная базисная переменная, т. е. процесс не завершён. Ведущей строкой будет строка x_3 . Только в столбце x_4 отрицательные элементы находятся и в строках, и в целевой функции. В качестве генерального элемента выбираем элемент (1, 4). Свободная переменная x_4 переходит в базисные на место x_3 . После операции замещения получаем таблицу, которая оказывается оптимальной (табл. П7.3).

Таблица П7.3

Оптимальное решение

Базис	b_i	x_1	x_2	x_3	x_4
x_4	1	1	0	-3	1
x_2	1	1	1	-1	0
z	-1	-2	0	-1	0

Решение: $x_1 = 0$; $x_2 = 1$; $x_3 = 0$; $x_4 = 1$.

Приложение 8

Что не поищешь, того не сыщешь.

Русская народная поговорка

КРАТКИЙ МАТЕМАТИЧЕСКИЙ СЛОВАРЬ

ВЕКТОРЫ И МАТРИЦЫ

Вектор — упорядоченная совокупность действительных чисел: $\mathbf{a} = (a_1, a_2, \dots, a_n)$.

Линейная зависимость векторов — векторы $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ линейно зависимы, если найдутся такие действительные числа $\alpha, \beta, \dots, \gamma$, не все равные нулю, что линейная комбинация векторов $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ равна нулю: $\alpha \mathbf{a}_1 + \beta \mathbf{a}_2 + \dots + \gamma \mathbf{a}_m = \mathbf{0}$. Если же это равенство выполняется только тогда, когда все числа $\alpha, \beta, \dots, \gamma$ равны нулю, то векторы $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ линейно независимы. Из определения линейной зависимости векторов следует, что если векторы линейно зависимы, то один из них может быть представлен в виде линейной комбинации остальных, и наоборот, если один из векторов есть линейная комбинация остальных, то векторы линейно зависимы.

Скалярное произведение двух векторов $\mathbf{a} = \{a_1, a_2, \dots, a_n\}$ и $\mathbf{b} = \{b_1, b_2, \dots, b_n\}$ — действительное число, равное сумме произведений соответствующих координат векторов $(\mathbf{a}, \mathbf{b}) = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$ или произведению длин этих векторов на косинус угла между ними: $(\mathbf{a}, \mathbf{b}) = |\mathbf{a}| \cdot |\mathbf{b}| \cos(\widehat{\mathbf{a}, \mathbf{b}})$.

Единичный орт \mathbf{a}_0 — вектор единичной длины, совпадающий по направлению с вектором \mathbf{a} ; $\mathbf{a}_0 = \mathbf{a}/|\mathbf{a}|$, где $|\mathbf{a}|$ — длина вектора \mathbf{a} .

Матрица — прямоугольная таблица чисел, применяемая для формальной записи систем линейных алгебраических уравнений.

Абсолютно унимодулярная матрица — матрица, все миноры которой равны либо 0, либо ± 1 . Проверить унимодулярность матрицы A , вычислив все возможные миноры, сложно. Существуют достаточные, но не необходимые условия абсолютной унимодулярности матрицы A , которые проверить гораздо легче. Матрица A абсолютно унимодулярна, если:

менной x_s , называют *генеральным элементом*, i -ю строку и s -й столбец — *разрешающими*. Прежде всего, разрешающую строку делят на a_{is} , и она остается без изменения. Чтобы исключить переменную x_s из первого уравнения, умножают разрешающую строку на $(-a_{is})$ и складывают с первой строкой. В результате получают первую строку с нулевым элементом на месте a_{is} . Аналогично исключают x_s в остальных строках. Получают эквивалентную запись системы алгебраических уравнений. В ней i -я строка имеет прежний вид, но все коэффициенты у нее поделены на a_{is} ; s -й столбец состоит из нулевых элементов (кроме единичного, стоящего в i -й строке). Остальные элементы матрицы системы и столбец свободных переменных пересчитывают по *правилу прямоугольника*. Например, новое значение элемента

$$a_{ln}^{\text{нов}} = \frac{a_{is}a_{ln} - a_{is}a_{in}}{a_{is}},$$

а новое значение элемента столбца свободных членов

$$b_m^{\text{нов}} = \frac{a_{is}b_m - a_{ms}b_i}{a_{is}}.$$

Из правила прямоугольника следует, что когда в разрешающей строке (столбце) есть нулевые элементы, то элементы столбцов (строк), пересекающих эти нулевые элементы, остаются без изменения.

Ранг матрицы — наивысший порядок не равного нулю минора этой матрицы.

Минор k -го порядка матрицы A ($k \leq n \cap k \leq m$; k, m — соответственно число строк и столбцов) — определитель k -го порядка, построенный из элементов, стоящих на пересечении произвольных k столбцов и k строк матрицы A .

Определитель — число, которое ставят в соответствие каждой квадратной матрице; существует определенное правило его нахождения.

Транспонирование — операция замены элементов строк матрицы (вектора) на соответствующие элементы столбцов и наоборот. Обозначают эту операцию символом «т», например

$$(a_1, a_2, \dots, a_n)^t = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix}.$$

МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Запись задачи математического программирования:

$$[\bullet] f(x) \rightarrow \min (\max);$$

$$[!] g_i(x) \geq 0, \quad i = 1, \dots, n;$$

$$h_j(x) = 0, \quad j = 1, \dots, p,$$

где $f(x)$ — целевая функция; $g_i(x)$ — условия-ограничения в виде неравенств; $h_j(x)$ — условия-ограничения в виде равенств.

Каноническая форма записи — все ограничения (кроме неотрицательности переменных) имеют вид равенств.

Стандартная форма записи — все ограничения имеют вид неравенств.

Целевая функция — в задаче математического программирования описывает оптимизируемый критерий.

Аддитивная целевая функция — ее значение за рассматриваемый период получают простым суммированием значений, достигнутых на отдельных шагах (на отдельных переменных).

Мультиплексная целевая функция — ее значения за рассматриваемый период равны произведению значений, достигнутых на отдельных шагах (на отдельных переменных).

Многокритериальная задача линейного программирования имеет вид

$$\max \{c_1 x = \zeta_1\};$$

$$\max \{c_2 x = \zeta_2\};$$

$$\dots\dots\dots$$

$$\max \{c_k x = \zeta_k\},$$

$$\text{при } x \in D,$$

или в матричной форме

$$\max \{Cx = \zeta \mid x \in D\},$$

где k — число целевых функций (критериев); c_i — градиент (вектор коэффициентов) i -й целевой функции (критерия); ζ_i — значение i -го критерия (целевой функции); D — множество допустимых значений переменных; \max — означает, что нужно максимизировать все целевые функции одновременно; C — матрица критериев размерностью $k \times n$ (матрица коэффициентов целевой функции, ее строки c_i являются градиентами критериев); z — вектор критериев.

Доминирование. Доминируемость относится к векторам в пространстве критериев.

Пусть $z_1, z_2 \in R^k$ — критериальные векторы. Вектор z_1 *доминирует* вектор z_2 тогда и только тогда, когда $z_1 \geq z_2$ и $z_1 \neq z_2$ (т. е. $z_{1i} \geq z_{2i}$ для всех i и $z_{1i} > z_{2i}$ по крайней мере для одного i). Вектор z_1 *сильно доминирует* вектор z_2 тогда и только тогда, когда $z_1 > z_2$ (т. е. $z_{1i} > z_{2i}$ для всех i).

Критериальный вектор $\bar{z} \in Z$ является *недоминируемым* тогда и только тогда, когда не существует другого вектора $z \in Z$ такого, что $z \geq \bar{z}$ и $z \neq \bar{z}$; иначе \bar{z} является доминируемым критерием. Критериальный вектор не может быть *оптимальным*, если он не является *недоминируемым*.

Эффективность — понятие, которое используется в пространстве решений. Точка $\bar{x} \in D$ эффективна тогда и только тогда, когда не существует другой точки $x \in D$ такой, что $Cx \geq C\bar{x}$ и $Cx \neq C\bar{x}$. В противном случае точка *неэффективна*. Точка $\bar{x} \in D$ эффективна, если ее критериальный вектор не доминируется критериальными векторами других точек из D . Из эффективной точки *невозможно* сдвинуться допустимым образом так, чтобы увеличить один из критериев, не уменьшив по крайней мере один из остальных. Вместе с термином «эффективность» используются термины «неулучшаемость», «оптимальность по Парето».

Программирование выпуклое — раздел математического программирования, рассматривающий выпуклые функции на выпуклых множествах, т. е. когда целевая функция и допустимое множество выпуклы.

Программирование динамическое — метод поэтапного решения задач математического программирования при наличии аддитивной целевой функции, или, другими словами, поэтапное планирование многошагового процесса при наличии аддитивной целевой функции, при котором на каждом этапе оптимизируют только по одной переменной с учетом всех последствий от других переменных.

Принцип оптимальности — оптимальная стратегия обладает тем свойством, что, каковы бы ни были первоначальные состояние и решение, последующее решение должно определять оптимальную стратегию относительно состояния, полученного в результате первоначального решения.

Программирование квадратичное — целевая функция $f(x)$ является квадратичной, а ограничения линейны (более точно $f(x)$ должна быть квазиопределенной квадратичной формой).

Программирование линейное — все функции $f(x)$, $g_i(x)$, $h_j(x)$ в задаче математического программирования (выражениях для целевой функции и для условий-ограничений) линейны.

Решения задачи линейного программирования. *Базисное решение* — решение системы ограничений, в котором все свободные переменные равны нулю. Если система имеет решение, то есть и базисное решение. *Допустимое решение* существует, если все компоненты решения неотрицательны. Если система уравнений обладает допустимым решением, то она имеет и *базисное допустимое решение*. Совокупность всех допустимых решений системы ограничений есть выпуклое множество. Базисное допустимое решение соответствует крайней точке выпуклого многогранника (грани или вершине этого многогранника), являющегося геометрическим образом системы ограничений. *Оптимальное решение* — это то из допустимых базисных решений задачи, при котором целевая функция достигает своего максимума (минимума).

Программирование нелинейное — хотя бы одна из функций $f(x)$, $g_i(x)$, $h_j(x)$ нелинейна.

Программирование параметрическое — коэффициенты в функциях $f(x)$, $g_i(x)$, $h_j(x)$ являются функциями некоторого параметра t .

Программирование сепарабельное — целевая функция $f(x)$ представляет собой сумму функций, различных для каждой переменной. Условия-ограничения могут быть как линейными, так и нелинейными (но все недиагональные элементы матрицы, состоящей из вторых частных производных любой функции задачи, равны нулю).

Программирование целевое — в целевом программировании (ЦП) устанавливается некоторый уровень достижения целей по каждому критерию. От обычного линейного программирования его отличает:

- 1) понимание критериев как целей;
- 2) приписывание приоритетов и/или весов достижению отдельных целей;

3) присутствие переменных d_i^+ и d_i^- , являющихся мерой отклонения от целевых уровней t_i сверху и снизу соответственно;

4) минимизация взвешенных сумм переменных отклонений для того, чтобы найти решения, которые наилучшим образом удовлетворяют целям.

Обычно точка, удовлетворяющая сразу всем целям, не является допустимой. Стараются найти допустимую точку, которая достигает всех целей наилучшим образом. Для каждой цели (целевой функции) устанавливается значение критериев z_i , которые должны быть достигнуты (если возможно) по отношению к обусловленным числами t_i целям.

Пример. Задача целевого программирования:

$$\text{цель } \{\bar{c}_1 \bar{x} = z_1\}, \quad z_1 \geq t_1;$$

$$\text{цель } \{\bar{c}_2 \bar{x} = z_2\}, \quad z_2 \geq t_2;$$

$$\text{цель } \{\bar{c}_3 \bar{x} = z_3\}, \quad z_3 \in [t_H^3, t_B^3],$$

$$x \in D.$$

Программирование целочисленное — когда координаты искомого вектора X являются только целыми числами.

Эквивалентные преобразования, используемые в линейном программировании.

1. Вместо точки минимума целевой функции $f(x)$ можно взять точку максимума функции $-f(x)$: $\min f(x) = \max [-f(x)]$.

2. Ограничения в форме неравенств можно представить в виде равенств, используя новые переменные x_i , $i = n + 1, n + 2, \dots, m$, $x_i \geq 0$, называемые *слабыми*:

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \rightarrow \sum_{j=1}^n a_{ij} x_j - x_i = b_i; \quad x_i \geq 0; \quad i = n + 1, \dots, m.$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \rightarrow \sum_{j=1}^n a_{ij} x_j + x_i = b_i; \quad x_i \geq 0; \quad i = n + 1, \dots, m.$$

3. Ограничение, записанное в виде равенства, можно заменить двумя неравенствами:

$$\sum_{j=1}^n a_{ij} x_j = b_i \rightarrow \left\{ \sum_{j=1}^n a_{ij} x_j \geq b_i; \sum_{j=1}^n a_{ij} x_j \leq b_i. \right\}$$

Если имеется m равенств, их можно заменить $(m + 1)$ неравенствами

$$\left\{ \begin{array}{l} \sum_{j=1}^n a_{1j} x_j = b_1 \\ \dots\dots\dots \\ \sum_{j=1}^n a_{mj} x_j = b_m \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \sum_{j=1}^n a_{1j} x_j \geq b_1 \\ \dots\dots\dots \\ \sum_{j=1}^n a_{mj} x_j \geq b_m \\ \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} - b_i \right) \leq 0. \end{array} \right.$$

Если на переменную $x_j, j = 1, \dots, n$, не наложено условие неотрицательности, ее можно заменить двумя неотрицательными переменными x_j^+ и x_j^- , положив

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

Если имеется n таких переменных x_j , то их можно заменить $(n + 1)$ неотрицательными переменными x_j^1 и x_0 , положив

$$x_1 \leq 0 \Rightarrow x_1 = x_1^1 - x_0;$$

$$\dots\dots\dots$$

$$x_n \leq 0 \Rightarrow x_n = x_n^1 - x_0;$$

$$x_1^1 \geq 0; \dots; x_n^1 \geq 0; x_0 \geq 0.$$

СЕТИ

Граф — абстрактное математическое понятие. В теории графов рассматривают конфигурации, состоящие из точек и соединяющих их линий. Пусть имеем некоторые множества N и A . Назовем N множеством *вершин* (узлов), A — множеством *дуг*. Граф $G = G(N, A)$ с множеством вершин N и множеством дуг A есть некоторое семейство сочетаний или пар вида (N, A) , указывающих, какие вершины считаются соединенными. В зависимости от вида дуг граф называют *ориентированным* или *неориентированным*.

Дерево — частный случай связных ориентированных и неориентированных сетей. Если множество всех узлов и дуг задать в виде графа $G(N, A)$, где N — множество узлов; A — множество дуг, то дерево определяется как связное подмножество (подграф) \bar{G} множества (графа) G , не содержащее циклов, т. е. для любых двух узлов дерева существует единственный путь, соединяющий их.

В сети, содержащей n узлов, подграф из k узлов ($k \leq n$) является деревом, если выполнены любые два из следующих условий:

- 1) подграф является связным;
- 2) подграф не имеет циклов;
- 3) число дуг в подграфе равно $k - 1$.

Вес дерева — сумма весов (длин) его дуг.

Остов (остовное связующее дерево) — дерево, содержащее все узлы сети. Если сеть содержит n узлов, то дерево с n узлами и с $n - 1$ дугами является остовом.

Кратчайший (максимальный) остов графа (сети) — дерево с минимальным (максимальным) весом среди всех связующих деревьев этого графа.

Дуга — элемент сети (графа), имеющий определенную ориентацию (направление), называется ориентированной или направленной дугой. В противном случае дуга называется *неориентированной*. Одну неориентированную дугу можно заменить двумя ориентированными.

Вес (длина) дуги — число, соответствующее некоторой характеристике дуги (расстояние, стоимость и т. д.).

Пропускная способность дуги — некоторое положительное число b_{ij} , поставленное в соответствие дуге A_{ij} .

Контур — конечная цепь, начальный и конечный узлы которой совпадают. Очевидно, что контур является замкнутой цепью.

Петля — это *вырожденный цикл*. Петля образуется одним узлом и одной дугой и поэтому является как *контуром*, так и *циклом*.

Поток из источника N_s в сток N_t — множество неотрицательных чисел x_{ij} , поставленных в соответствие некоторой дуге сети, если числа удовлетворяют следующим линейным ограничениям:

$$\sum_i x_{ij} - \sum_k x_{jk} = \begin{cases} -v, & \text{если } i = s; \\ 0, & \text{если } j \neq s, j \neq t; \\ v, & \text{если } j = t; \end{cases} \quad (\text{П8.1})$$

$$v \geq 0; \quad 0 \leq x_{ij} \leq b_{ij} \quad \text{для всех } i, j. \quad (\text{П8.2})$$

Здесь первую сумму берут по дугам, ведущим в узел N_j , а вторую сумму — по дугам, ведущим из узла N_j . Неотрицательное число v называют величиной потока. Число x_{ij} называют *потоком* по дуге A_{ij} , или *дуговым потоком*. Ограничения (П8.1) означают, что в каждый узел (кроме источника и стока) приходит столько пото-

ка, сколько из него уходит (условие сохранения потока). Ограничение (П8.2) означает, что поток по дуге ограничен пропускной способностью дуги b_{ij} .

Теорема о максимальном потоке и минимальном разрезе — обозначим через F_{st} множество неотрицательных чисел x_{ij} , удовлетворяющих ограничениям (П8.1), (П8.2). Будем называть путь из N_s в N_t *увеличивающим поток F_{st}* , если $x_{ij} < b_{ij}$ на всех прямых дугах и $x_{ji} > 0$ на всех обратных дугах этого пути. Поток F_{st} максимален тогда и только тогда, когда не существует пути, увеличивающего поток F_{st} .

Величина максимального потока в любой сети принимает, безусловно, единственное значение. Но может существовать несколько различных максимальных потоков F_{st} , имеющих одинаковую величину. Может существовать и несколько минимальных разрезов в сети.

Путь — последовательность $N_1, A_{12}, N_2, \dots, N_{k-1}, A_{k-1,k}, N_k$, где $N_1, A_{12}, N_2, \dots, N_{k-1}, A_{k-1,k}, N_k$ — узлы сети, и либо $A_{i,i+1}$, либо $A_{i+1,i}$ ($i = 1, 2, \dots, k-1$) является дугой сети. Путь отличается от цепи тем, что при движении по пути от N_1 до N_k можно пройти дугу сети и в направлении, противоположном ее ориентации. Для неориентированных сетей понятия «цепь» и «путь» совпадают.

Разрез (X, \bar{X}) — множество всех дуг A_{ij} , для которых $N_i \in X$, $N_j \in \bar{X}$, и удаление которых из сети превращает сеть в несвязную (X — некоторое подмножество узлов сети, \bar{X} — дополнение подмножества X , объединение X и \bar{X} определяет множество узлов сети). Разрез называют *разделяющим узлы N_s и N_t* (или *отделяющим узел N_s от N_t*), если $N_s \in X$, $N_t \in \bar{X}$.

Минимальный разрез — разрез, разделяющий N_s и N_t и обладающий минимальной пропускной способностью.

Пропускная способность или величина разреза $c(X, \bar{X})$ — сумма $\sum_{i,j} b_{ij}$, где сумма берется по всем ориентированным дугам,

соединяющим и $N_i \in X$ и $N_j \in \bar{X}$. При определении разреза учитываются все дуги между подмножествами X и \bar{X} , а при определении пропускной способности разреза — только пропускные способности дуг из X в \bar{X} , ориентированные дуги из \bar{X} в X при этом не учитывают. Поэтому в общем случае $c(X, \bar{X}) \neq c(\bar{X}, X)$.

Ясно, что вследствие ограничений (П8.1), (П8.2) величина максимального потока меньше или равна пропускной способности любого разреза, разделяющего N_s и N_t , так как

$$v = \sum_{\substack{N_i \in X, \\ N_j \in \bar{X}}} x_{ij} - \sum_{\substack{N_i \in X, \\ N_j \in \bar{X}}} x_{ji}.$$

В любой сети величина максимального потока из источника N_s в сток N_t всегда равна минимальной пропускной способности всех разрезов, разделяющих N_s и N_t .

Сеть — множество узлов (называемых также *вершинами*, или *точками соединения*) и множество дуг (называемых также *звеньями*, или *ребрами*), которые связывают эти узлы. В сети выделяют два специальных узла: один из них называют источником N_s , а другой — стоком N_t . Сеть можно рассматривать как водопроводную систему, в которой трубы соответствуют дугам, источник воды — источнику N_s , сток воды — стоку N_t а соединения между трубами — остальным узлам сети. В качестве пропускной способности дуги выступает поперечное сечение трубы.

Сеть связная — сеть, при любом разбиении множества узлов N которой на подмножества X и \bar{X} , найдется дуга A_{ij} , когда i -й узел N_i принадлежит подмножеству X , $N_i \in X$, j -й узел N_j принадлежит подмножеству \bar{X} , $N_j \in \bar{X}$. В связной сети для любых двух различных узлов существует, по крайней мере, один соединяющий их путь или цепь.

Цепь. Последовательность узлов и дуг сети $N_1, A_{12}, N_2, A_{23}, \dots, N_{k-1}, A_{k-1,k}, N_k$ называют *цепью* или *ориентированной цепью*, ведущей из узла N_1 в узел N_k . Если $N_1 = N_k$, то такую последовательность называют ориентированным циклом. Цикл является замкнутым путем.

Цепь простая — цепь, которая не содержит циклов. Если сеть является цепью $N_1, A_{12}, N_2, \dots, N_k$ с источником N_1 и стоком N_k , максимальная величина потока, который может быть пропущен через сеть, ограничивается минимальной из пропускных способностей дуг этой сети. Дуга с минимальной пропускной способностью является «узким местом» в сети. В произвольной сети «узкое место» определяется *разрезом*.

ТЕОРИЯ ИГР

Задача теории игр — выработка игроками стратегии, которая обеспечивает одной стороне максимальный выигрыш, а другой — минимальный проигрыш. Теория игр используется в конфликтных ситуациях, которые повторяются многократно.

Доминирование. Если в платежной матрице для двух строк элементы a_{kj} не меньше элементов a_{sj} ($a_{kj} \geq a_{sj}$) и хотя бы один элемент k -й строки строго больше соответствующего элемента s -й строки, то для игрока A стратегия A_k предпочтительнее стратегии A_s . Стратегия A_k называется *доминирующей*, стратегия A_s — *доминируемой*, ее можно удалить (в смешанной стратегии получим вероятность, равную нулю). Доминируемая стратегия *заведомо невыгодная*.

Аналогично для столбцов. Столбцы и строки, у которых соответствующие элементы равны, называются дублирующими. В платежной матрице игры следует оставить только одну строку или один столбец из дублирующих.

Игры. Игры, состоящие только из случайных ходов, называют *азартными*. Игры, в которых имеются *личные ходы*, называются *стратегическими*. Существуют стратегические игры, состоящие *только из личных ходов*. Теория игр занимается *только стратегическими играми*.

Дифференциальные игры — игры, когда число ходов в игре становится бесконечным (континуум) и игроки имеют возможность принимать решения непрерывно. Здесь траектории движения игроков представляют собой решения систем обыкновенных дифференциальных уравнений, правые части которых зависят от параметров, находящихся под контролем игроков. Рассматриваются две системы обыкновенных дифференциальных уравнений с начальными условиями (x_0, y_0) :

$$x = f(x, u) \text{ и } y = g(y, v).$$

Конечные парные игры с нулевой суммой — игры, в которых участвуют два игрока. Участников игр может быть более двух, они могут быть объединены в группы. Тогда игра называется *коалиционной*. Игра называется *конечной*, если у каждого игрока есть конечное число стратегий, т. е. конечное число вариантов поведения, и *бесконечной*, если хотя бы у одного из игроков имеется бесконечное число стратегий. Игра с *нулевой суммой* есть игра, в которой выигрыш одного игрока равен проигрышу другого.

Оптимальная стратегия игрока — это такая стратегия, которая при многократном повторении игры обеспечивает игроку максимально возможный средний выигрыш, а другому — минимальный средний проигрыш.

Платежная матрица игры — набор выигрышей a_{ij} для разных значений i и j располагают в виде матрицы, строки которой отвечают стратегиям игрока A , а столбцы — стратегиям игрока B .

Принцип минимакса. С точки зрения игрока A ему необходимо выбрать такую стратегию A_i , которая *максимизирует* минимально возможный выигрыш $\alpha = \max_i \min_j a_{ij}$ — *максиминная стратегия (максимин)*. Величину α называют *нижней ценой игры*. Меньше чем α игрок A выиграть не может. Для игрока B оптимальная стратегия та, при которой максимальный возможный выигрыш игрока A оказывается наименьшим и в любом случае не превосходит $\beta = \min_j \max_i a_{ij}$. Величина β называется минимаксом, или верхней ценой игры. *Принцип осторожности, диктующий игрокам выбор таких стратегий, называется принципом минимакса.*

Седловая точка. Рассмотрим некоторую игру, в которой максиминный и минимаксный выигрыши равны, т. е. нижняя и верхняя цены игры совпадают: $\alpha = \beta$. Выигрыш является одновременно и максиминным из минимальных выигрышей для игрока A , и минимальным из максимальных выигрышей для игрока B . Точку на поверхности, являющуюся одновременно точкой минимума по одной оси координат и точкой максимума по другой оси, называют седловой точкой. Соответствующий элемент $a_{ij} = \alpha$ (или β) платежной матрицы игры называют седловой точкой (седловым элементом) матрицы. Об игре говорят, что она имеет седловую точку. В такой ситуации каждый из игроков должен придерживаться максиминной (минимаксной) стратегии. Любое отклонение от этой стратегии будет невыгодно для игрока, допустившего отклонение.

Смешанные стратегии. Пусть A_1, A_2, \dots, A_m — возможные стратегии игрока A . Для получения наибольшего эффекта игрок A должен использовать все или некоторые из этих стратегий случайным образом с разными вероятностями: стратегия A_i используется с вероятностью p_i . Говорят, что игрок A применяет смешанную стратегию $S_A(p_1, p_2, \dots, p_m)$. В отличие от смешанных стратегий S_A стратегии A_i называют чистыми. При надлежащем выборе вероятностей p_i смешанная стратегия может оказаться оптимальной (аналог седловой точки). При этом выигрыш игрока A будет не меньше

некоторого значения v , называемого ценой игры. Это значение больше нижней цены игры, но меньше верхней в принципе минимакса.

Ходы. Личный ход предполагает сознательный выбор игроком того или иного действия, разрешенного правилами игры. Случайный ход не зависит от воли игрока — он может быть определен с помощью датчика случайных чисел.

ФУНКЦИЯ И ПРОИЗВОДНЫЕ

График функции. Если независимую переменную x (аргумент) и зависимую переменную y рассматривать как декартовы координаты на плоскости, то действительную функцию $y = f(x)$ действительного переменного x можно изобразить кривой — графиком функции y от x . Для функций многих переменных $y = f(x_1, x_2, \dots, x_n)$ упорядоченному множеству значений независимых переменных x_1, x_2, \dots, x_n ставят в соответствие значения переменного y . Множества значений x_1, x_2, \dots, x_n , для которых определено соотношение $y = f(x_1, x_2, \dots, x_n)$, есть область определения функции $f(x_1, x_2, \dots, x_n)$. Графиком функции многих переменных является поверхность (для функции двух переменных) и гиперповерхность (для большего числа переменных).

Гиперплоскость — аналог плоскости в пространствах с размерностью больше трех.

Линии (поверхности) уровня — геометрическое место точек, в которых функция принимает одно и то же значение. Уравнение поверхности уровня имеет вид $f(x_1, x_2, \dots, x_n) = c$. Давая константе c различные значения, получают семейство поверхностей уровня, определяющих поведение функции. Линии уровня вводят для функции двух переменных: $f(x_1, x_2) = c$. Семейство линий уровня дает возможность представить функцию двух переменных $y = f(x_1, x_2)$ на плоскости. Например, семейство линий уровня на географических картах дает представление и о морских глубинах, и о горных хребтах.

Множество выпуклое — для любых точек M_1 и M_2 , принадлежащих области D , отрезок M_1M_2 принадлежит множеству (области) D (см. рис. 1.6, *a*). Другими словами, любая точка $[\lambda M_1 + (1 - \lambda)M_2]$ принадлежит области D для любого λ , $0 \leq \lambda \leq 1$, и для любых точек M_1 и M_2 области D . При этом пересечение конечного числа выпуклых множеств выпукло. На рис. 1.6, *б* показаны невыпуклые множества.

Область (множество) связная — любая замкнутая ломаная линия, целиком принадлежащая этой области, может быть стянута в точку с помощью непрерывной деформации и не выходит из данной области.

Производная по направлению — действительное число, которое определяет скорость изменения скалярной функции $f(x_1, x_2, \dots, x_n)$ в произвольном направлении, задаваемом единичным вектором $\mathbf{u} = \cos \alpha \mathbf{i} + \cos \beta \mathbf{j} + \dots + \cos \gamma \mathbf{k}$ с направляющими косинусами $\cos \alpha, \cos \beta, \dots, \cos \gamma$;

$$\frac{df}{du} = \frac{\partial f}{\partial x_1} \cos \alpha + \frac{\partial f}{\partial x_2} \cos \beta + \dots + \frac{\partial f}{\partial x_n} \cos \gamma.$$

Производная по направлению и градиент ∇f скалярной функции связаны скалярным произведением: $df/du = (\nabla f, \mathbf{u})$. Производная df/du по направлению касательной к поверхности (линии) уровня равна нулю.

Производная частная — характеризует скорость изменения функции многих переменных относительно одной из переменных, например x_i , при фиксированных значениях остальных независимых переменных. Частная производная может быть найдена дифференцированием функции $f(x_1, x_2, \dots, x_n)$ по x_i , если остальные $n - 1$ независимых переменных рассматривать как постоянные.

Форма квадратичная — числовая функция $f(\mathbf{x})$ одного векторного аргумента $\mathbf{x} = (x_1, x_2, \dots, x_n)$ вида

$$f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j \equiv \sum_{i=1}^n a_{ij} x_i^2 + 2 \sum_{1 \leq i < j \leq n} a_{ij} x_i x_j,$$

где a_{ij} — элементы симметричной матрицы (квадратной таблицы чисел) $A = \| a_{ij} \|_{n \times n}$ порядка n , $a_{ij} = a_{ji}$.

Форма квадратичная знакопеременная. Форма $f(\mathbf{x})$ называется квадратичной знакопеременной, если существуют такие векторы \mathbf{x}_1 и \mathbf{x}_2 , для которых $f(\mathbf{x}_1) > 0$ и $f(\mathbf{x}_2) < 0$.

Форма квадратичная квазизнакоопределенная. Знакоопределенная форма $f(\mathbf{x})$ называется квадратичной квазизнакоопределенной, если имеется ненулевой вектор \mathbf{x} , для которого $f(\mathbf{x}) = 0$.

Форма квадратичная положительно (отрицательно) определенная. Квадратичная форма $f(\mathbf{x})$ называется положительно (отрицательно) определенной, если для любого ненулевого вектора \mathbf{x} выполняется неравенство $f(\mathbf{x}) > 0$ ($f(\mathbf{x}) < 0$). Положительно и отри-

цательно определенные квадратичные формы называются знако-постоянными.

Функция вогнутая. Функцию $f(x)$ называют вогнутой на выпуклом множестве D , если функция $f(x)$ выпукла на D . Ограничения в задаче математического программирования $g_i(x) \geq 0$, $i = 1, \dots, m$, образуют выпуклое множество D (выпуклую область), если все функции $g_i(x)$ вогнуты.

Функция выпуклая. Функцию $f(M)$ называют выпуклой на непустом множестве D , если для любых двух точек M_1 и M_2 , принадлежащих области D , и любого числа λ , $0 \leq \lambda \leq 1$, справедливо неравенство

$$f[\lambda M_1 + (1 - \lambda)M_2] \leq \lambda f(M_1) + (1 - \lambda)f(M_2).$$

Геометрический смысл: выпуклая функция лежит над своими касательными. Примером выпуклой функции является парабола.

Функция строго выпуклая. Функцию $f(M)$ называют строго выпуклой, если для $0 < \lambda < 1$ и $M_1 \neq M_2$ выполняется строгое неравенство

$$f[\lambda M_1 + (1 - \lambda)M_2] < \lambda f(M_1) + (1 - \lambda)f(M_2).$$

Функция дифференцируемая. Функция $f(x)$, имеющая в точке $x = x_0$ производную, называется дифференцируемой в этой точке. Функция, имеющая производную во всех точках некоторого промежутка (a, b) , называется дифференцируемой в этом промежутке. Функция многих переменных, имеющая полный дифференциал (в данной точке, области), называется дифференцируемой (в этой точке, области).

Достаточные условия дифференцируемости функции многих переменных — существование и непрерывность всех частных производных первого порядка (в точке, области).

Необходимые условия дифференцируемости функции многих переменных — существование частных производных первого порядка (в точке, области).

Функция Лагранжа — вспомогательная функция в математическом программировании, с помощью которой задача на условный экстремум сводится к задаче на безусловный экстремум. Функция Лагранжа представляет собой сумму целевой функции и сумму уравнений связи, умноженных на действительные числа, называемые множителями Лагранжа.

Функция скалярная — функция $y = f(x)$ называется скалярной, если любое значение y есть скалярная величина.

Градиент скалярной функции — векторная функция точки, определяемая как

$$\text{grad } f = \nabla f(x_1, x_2, \dots, x_n) = \frac{\partial f}{\partial x_1} \mathbf{i} + \frac{\partial f}{\partial x_2} \mathbf{j} + \dots + \frac{\partial f}{\partial x_n} \mathbf{k},$$

где ∇ — знак градиента (набла); $\mathbf{i}, \mathbf{j}, \mathbf{k}$ — единичные векторы (орты), направленные по координатным осям: $\mathbf{i} = \{1, 0, \dots, 0\}$, $\mathbf{j} = \{0, 1, \dots, 0\}$, $\mathbf{k} = \{0, 0, \dots, 1\}$.

Иногда применяют обозначение градиента в виде $\nabla_x f$, индекс x у оператора набла показывает переменную, по которой определяют градиент. Другими словами, градиент скалярной функции — это вектор, координатами которого являются частные производные заданной функции. Градиент указывает направление, в котором скорость возрастания скалярной функции наибольшая. Противоположное направление определяют антиградиентом. Градиент ∇f всегда ортогонален (перпендикулярен) поверхности (линии) уровня функции $f(x, x_2, \dots, x_n)$.

Экстремум. Функция $y = f(x)$, определенная в точке x_0 , достигает экстремума (максимума или минимума) в окрестности точки x_0 , если для всех точек x этой окрестности удовлетворяется неравенство $f(x) \leq f(x_0)$ или $f(x) \geq f(x_0)$.

Достаточные условия существования экстремума. Для того, чтобы в точке x_0 был экстремум, производная $f'(x)$ в окрестности точки x_0 при переходе через $x = x_0$ должна менять свой знак с плюса на минус в точке максимума и с минуса на плюс в точке минимума. Другой признак: если в точке x_0 первая производная $f'(x_0) = 0$ и существует вторая производная $f''(x_0)$, то в точке x_0 будет максимум при $f''(x_0) < 0$ и минимум при $f''(x_0) > 0$. В общем случае, если существуют производные от $f(x)$ до n -го порядка включительно и если $f'(x_0) = f''(x_0) = \dots = f^{(n-1)}(x_0) = 0$ и $f^{(n)}(x_0) \neq 0$, то функция $f(x)$ имеет в точке x_0 максимум при n четном, если $f^{(n)}(x_0) < 0$ и минимум при n четном и $f^{(n)}(x_0) > 0$. Если n нечетно, то функция $f(x)$ в точке x_0 не имеет ни минимума, ни максимума, а имеет точку перегиба.

Необходимые условия существования экстремума. Если существует производная $f'(x)$ в точке x_0 , то функция $f(x)$ может иметь в точке x внутренний экстремум лишь в том случае, когда

при $x = x_0$ производная $f'(x_0) = 0$. Экстремум может быть и в тех точках x_0 , где производная $f'(x_0)$ не существует. Но выполнение необходимого условия еще не означает, что в точке x_0 будет экстремум.

Экстремум внутренний. Локальный максимум (минимум) называется внутренним или граничным, если x_0 является внутренней или граничной точкой области определения функции $f(x)$. Например, функция $f_1(x) = x$ при $-\infty < x < \infty$ не имеет максимума, а функция $f_2(x) = x$ при $x \leq 1$ имеет при $x = 1$ граничный максимум.

Экстремум глобальный. Неравенства $f(x) < f(x_0)$ ($f(x) > f(x_0)$) выполняются для любой точки x , принадлежащей области определения функции $f(x)$.

Экстремум локальный. Действительная функция $f(x)$, определенная при $x = x_0$, имеет в точке x_0 (локальный) максимум или (локальный) минимум $f(x_0)$, если существует такое положительное число δ , что при всех приращениях $\Delta x = x - x_0$ независимого переменного x , для которых выполняется неравенство $0 < |\Delta x| < \delta$ и существует значение $f(x_0 + \Delta x)$, приращение данной функции соответственно $\Delta f \equiv f(x_0 + \Delta x) - f(x_0) < 0$ или $\Delta f \equiv f(x_0 + \Delta x) - f(x_0) > 0$.

Если в каждом из этих случаев выполняются нестрогие неравенства, то говорят, что функция $f(x)$ имеет в точке x_0 нестрогий максимум (минимум).

Экстремум условный. Экстремум функции m переменных в предположении, что переменные подчинены еще n уравнениям связи (условиям). При отсутствии уравнений связи имеет место безусловный экстремум.

СПИСОК МАТЕМАТИЧЕСКИХ СИМВОЛОВ

$=$	равно
\equiv	тождественно равно
∞	бесконечность
\neq	не равно
\approx	приближенно равно
\sim	пропорционально
$<$	меньше
$>$	больше
\leq	меньше или равно
\geq	больше или равно
\subset	знак содержания для множеств
\in	знак принадлежности
\notin	знак непринадлежности
\cup	объединение двух множеств
\cap	пересечение двух множеств
\emptyset	пустое множество
\Leftrightarrow	эквивалентно
\Rightarrow	следует
\exists	квантор существования
\forall	квантор общности
$!$	знак факториала: $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$
Σ	знак суммы: $\sum_{i=1}^n a_i = a_1 + a_2 + \dots + a_n$

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

- Агрегирование взвешенное 245
- Алгоритм венгерский 150
 - Флойда 144
- Анализ почти оптимальности 289
 - решения содержательный 15
 - — формальный 15
 - сетевой 24
 - системный 18, 20
- Антиградиент 30, 52
 - целевой функции 38
- Вариационное исчисление 17
- Вектор весовой 287, 427
 - доминируемый 293
 - Куна — Таккера 40
 - недоминируемый 284, 293
 - признаков 334
 - разделяющий 458
 - решения 458
- Величина потока 133
 - разреза 134
- Вершина 131
- Вершины инцидентные 261
- Вес дерева 133
 - дуги 133
- Вид анализа экстенсивный 327
- Возврат 180
- Возможность узла 136
- Время упреждения 502
- Выборка последовательная 343
- Выборки линейно разделяемые 458
- Выигрыш вероятностный 222
- Вырождение 77, 94
- Гиперповерхность 29
- Градиент 30, 94
- Граница выигрыша верхняя 221
 - — нижняя 220, 221
 - останавливающая 337
 - решающая 334
 - стоимости маршрута нижняя 173
- Граф 132
- График функции 29
 - многих переменных 29
- Дезагрегирование с фиксированными весами 246
- Дерево 132
 - остовное связующее (остов) 132
 - разрезов 164, 168
- Дискриминант Фишера линейный 456
- Длина дуги 133
- Доминирование 284
 - сильное 284
- Дуга 131, 171
 - направленная 131
 - обратная 158
 - ориентированная 131
 - прямая 158
- Задача, принадлежащая значению 230
 - анализа сети 163
 - выпуклого программирования 35
 - динамическая 18
- Задача исходная 228
 - квадратичного программирования 26, 43
 - коммивояжера 131, 172
 - двойственная 85
 - математического программирования 322
 - многокритериальная 253
 - многопродуктовая о перевозках 239
 - — сетевая 227
 - нахождения кратчайших целей между всеми парами узлов сети 143
 - нахождения максимума функции 27
 - — минимума функции 27
 - некорректная 434
 - о брахистохроне 18
 - о Кёнигсбергских мостах 17
 - о кратчайшей цепи 136
 - о максимальном потоке 136
 - о многополосной цепи с максимальной пропускной способностью 181
 - о многополосном максимальном потоке 163

- Задача о многопродуктовом потоке 138, 227, 238
- о назначениях 110
 - о наикратчайшей сети 139
 - о питании 15
 - о покрытии множества 115
 - о потоке в сети с несколькими источниками и стоками 137
 - о потоке минимальной стоимости 135
 - о рюкзаке 190
 - о сети наименьшей стоимости 139
 - о частичном покрытии 115
- Задача оптимального синтеза структуры сети 252
- оптимального синтеза структуры сети 252
 - оптимизации безусловной 17
 - — двойственная 40, 85
 - оценки параметров общего распределения 349
 - параметрическая 229
 - поиска потока минимальной стоимости 22
 - принятия решения статистическая 323
 - прямая 42, 85
 - сетевая 22, 131
 - сетевая (потоковая) 21
 - синтез сети 163
 - статическая 18
 - транспортная 22, 93
 - целочисленного программирования 101
 - частично целочисленная 109
- Зацикливание 77
- Звено 131
- маршрута 172
- Звенья запрещенные 177
- Значение критическое 231
- Зона неопределенности в принятии решений 445
- нулевая 445
- Игра азартная 206
- бесконечная 207
 - коалиционная 207
 - — конечная 207
 - парная 207
 - — конечная с нулевой суммой 207
 - полностью усредненная 217
 - стратегическая 206
- Игрок 206
- Игры дифференциальные 225
- с седловой точкой 212
- Исследование операций 19
- Источник 133
- Квадратичная форма 29
- Знакоопределенная 29
- — знакопеременная 29
 - — квазизнакоопределенная 29
 - — отрицательно определенная 29
 - — положительно определенная 29
- Класс полный 316
- Классификатор по минимальному расстоянию 325
- Контур 132
- Коэффициент покрытия 116
- правдоподобия 314
- Кривая регрессии 364
- Критерий Байеса 318
- минимаксный 319
 - Неймана — Пирсона 320
 - оптимальности аддитивный 186
 - — мультиплексный 186
 - оптимизируемый 11
 - отношения вероятностей последовательный 345
 - — — обобщенный 340
 - последовательный отношения вероятностей Вальда 336
- Линейная зависимость 18
- — векторов 31
 - независимость векторов 31
- Линия выигрыша 220
- уровня 29
- Максимизация лексикографическая 291
- Максимум внутренний 27

- Максимум глобальный 27
— граничный 27
— локальный 26
— нестрогий 27
Маршрут допустимый 172
Математическое программирование 16
Матрица абсолютно унимодулярная 135
— маршпутов 182
— ортогональная 434
— платежная 208
— расширенная 67
— редуцированная 173
— стоимости возврата 180
— унимодулярная 241
Машина линейная 455
Метод ϵ -ограничений 289
— агрегирования 245
— Бенсона 292
— ветвей и границ 172
— взвешенных сумм 291
— взвешенных сумм с точным оцениванием весов 286
— внешней точки 227
— внутренней точки 227, 272
— внутренней и внешней точки комбинированный 227
— градиентный 52
— динамического программирования 186
— итераций 224
— максимума правдоподобия 363, 382
— минимакса Неймана 316
— наименьших квадратов 363
— неопределенных множителей Лагранжа 18
— Ньютона 49, 60
— поиска по дереву решений 172
— полного исключения Жордана 76
— последовательного сокращения невязок 92
— потенциалов 96
— проекции градиента 227, 278
— северо-западного угла 94
— сопряженных градиентов 54
— сопряженных направлений 54
Метод учета наименьших расстояний (стоимостей) 96
— штрафных (барьерных) функций 227
— эвристический 115, 226
— Эккера — Куоды 292
Методы линейные 52
Минимакс 208
Минимизация безусловная 48
— вероятности 329
Минимум внутренний 27
— глобальный 27, 35
— граничный 27
— локальный 26, 35
— нестрогий 27
Минор k -го порядка 135
Множество вращения 116
— выпуклое 35
— утопическое в пространстве критериев 301, 302
— — решений 301
— эффективное 285
Множитель Лагранжа 18, 32
Модель Архимедова 302
— детерминированная 499
— неполного ранга 372
— параметрическая 499
— полного ранга 372
— с приоритетами 302
— сетевая 22
— статистическая 499
— стохастическая 499
Мощность критерия 313
Назначение 150
Направления сопряженные 48
— — относительно матрицы 48
Невязка 151, 559
Неопределенность поведения 205
Неравенство Крамера — Рао 382
— линейное 18
Область критическая 230
— недопустимая 274
— определения функции 27
— Парето 285

- Область решений 459
Обучение без поощрения 347
— без учителя 347
— в линейном классификаторе 336
— с поощрением 347
— с учителем 347
Ограничения линейные 25
Оператор разностный 508
— сдвига вперед 508
— — назад 508
— суммирования 508
Операция трехместная 147
Оптимальное управление услов-
ное 189, 195, 196
Оптимальность по Парето 285
Оптимальный доход условный 196
Остов 132
— графа (сети) 132
— — кратчайший (максималь-
ный) 131, 132
— дерева максимальный 164
Отклонения нежелательные 302
Отношение правдоподобия 314, 331
— решающее 230
Отсечение Гомори 105
Оценка смещенная 363
Ошибка второго рода 313
— остаточная 544
— первого рода 313

Партия игры 206
Переменная базисная 66
— контролируемая 391, 417
— свободная 66
— слабая 65, 122
— слабая Гомори 105
Петля 132
Поведение бескоалиционное 224
— кооперативное 225
Поверхность 29
— уровня 29
Подграф 132
Подзадача-тест 291
Подход Байесовский 316
— классический 316
Пометка 159
Пометка временная 142
— постоянная 142
Порог верхний 337
— нижний 337
Поставки условные 95
Потенциал 96
Потеря апостериорная 316
Поток 133
— дуговой 133
Правило Байесовское 330
Правило прямоугольника 76
— решения 315
— — допустимое 316
Преодоление конфликта 205
Принцип минимакса 209
Принятие решений многоэтапное 309
— — одноэтапное 309
Проблема мультиколлинеарности 521
Программирование булево 256
— динамическое 19
— квадратичное 26, 43
— линейное 18, 26
— нелинейное 26, 43
— параметрическое 226, 228
— стохастическое 226
— целочисленное 26, 101
Проекция точки на множество 278
Произведение векторов скалярное 31
Производная по направлению 30
— частная 30
Пропускная способность дуги (реб-
ра) 133
— — остаточная 158
— — разреза 134
Пространство наблюдений 314
— параметров 315
— решений 315
Процедура решающая последователь-
ная 336
— решения 315
Процесс авторегрессии порядка p 508
— авторегрессии—проинтегриро-
ванного скользящего сред-
него 500, 510
— нестационарный 504
— скользящего среднего порядка q 509

- Процесс стационарный 500, 504
Псевдоплан 89
Путь 132
—, увеличивающий поток 135, 159
— аугментальный 159
— замкнутый 132
Равенство линейное 18
Разрез 134
— минимальный 134, 164
— разделяющий 134
Распределение апостериорное 312, 325
— априорное 325
Расстановка пометок 166
Ребро 131
Регрессия 364, 365
— линейная средняя квадратическая 366
Редукция столбцов 174
— строк 174
— строк и столбцов 173
Решение байесовское 317
— допустимое 66, 173
— задачи 14
— опорное 82
— оптимальное 206
— параметрической задачи 230
— системы базисное оптимальное 66
— — уравнений базисное 66
— — базисное допустимое 66
— — допустимое 66
Риск 312, 317
— апостериорный 315
— байесовский 317, 318, 324
— выборочный 454
— общий 326
— условный 312

Свойство теоретико-информационное 18
Сеть 131
— потоко-эквивалентная по отношению к заданному множеству узлов 169
— связанная 131
Сечение сезонное 538
Симплекс 71
Симплекс-метод 68
— двойственный 91
Система линейных уравнений избыточная 66
— — — несовместная 66
— — — совместная 66
— нестационарная 500
— плохо обусловленная 433
— сложная 23
— стационарная 500
Ситуация конфликтная 205
Соотношение двойственности 43
Состояние природы 312
Спрос 137
Средняя экспоненциальная k -го порядка 517
Стабилизатор 436
Сток 133
Столбец дублирующий 215
— разрешающий 76
Стратегия активная 214
— байесовская 318
— доминируемая 215
— доминирующая 215
— заведомо невыгодная 215
— максиминная (максимин) 208
— оптимальная 207, 213
— — смешанная 223
— смешанная 213
— чистая 212
Строка ведущая 105
— дублирующая 215
— производящая 105
Структура сети 252

Таблица выигрышей 293
Теоремы дополняющей нежесткости 38, 88
Теория двойственности 40, 85
— игр 19, 206
— статистических решений 207
Топология сети 252
Точка крайняя (вершина) 66, 72
— неэффективная 285
— параметрически оптимальная 294
— перегиба 26

Точка седловая 44, 211, 212

— соединения 131

— эффективная 285

Транспонирование 86

Узел 131

—, помеченный из узла символом 159

— конденсированный 165

Узкое место 134

Уравнение правдоподобия 382

— связи 13

Условие дифференцируемости необ-
ходимое 27

— дополняющей нежесткости 38, 88

— Куна — Таккера 37, 40

— локальной оптимальности достаточ-
ное 27

Условия-ограничения 11

— достаточные 27

Ущерб 317

Форма каноническая 64

— стандартная 64

Функционал 16

Функция, дифференцируемая в облас-
ти 28

—, дифференцируемая в промежут-
ке 28

—, дифференцируемая в точке 28

— аддитивная 205

— векторная 29, 31

— вогнутая 34

— выпуклая 34

— — строго 34

— задачи решающая 230

— квадратичная 26, 43

— Лагранжа 18

— обобщенная линейная разделяю-
щая 457

— полезности 283, 316

— — по координатно возрастающая 284

— потенциальная 462

— потеря 315

— — неотрицательная 317

— правдоподобия 380

— прогнозирующая 502

Функция разделяющая 331, 334

— — квадратичная 335

— — линейная 334

— — полиномиальная 335

— решающая 315, 323, 329

— — байесовская 324

— риска 315

— стохастического процесса автоко-
вариационная 526

— — автокорреляционная 526

— целевая 11

— — выпуклая 34

— — многокритериальная 18

— штрафная (барьерная) 272

— — логарифмическая 271

Характеристика рабочая 320

Ход 206

— личный 206

— случайный 206

Цена игры 213

— — верхняя 209

— — нижняя 208

— наблюдения 326, 327

Цепь 132

— замкнутая 132

— ориентированная 132

— простая 132

Цикл вырожденный 132

— ориентированный 132

— пересчета 97

Число информационное 347

Штраф вторичный 176

Экстремум 26

— безусловный 32, 48

— внутренний 26

— условный 16, 32

Экстремума условие необходимое 27

Элемент генеральный 76, 80

— матрицы седловой 212

— разрешающий 80

Эффективность 285

К 75-ЛЕТИЮ АВТОРА



Анатолий Антонович Грепилов, доктор технических наук, профессор, родился 19 марта 1939 г., в 1964 г. окончил факультет экспериментальной и теоретической физики Московского инженерно-физического института. С 1964 по 1977 г. был непосредственным участником испытаний ядерного оружия на Семипалатинском и Новоземельском испытательных полигонах. В 1967–1968 гг. предложил и обосновал метод определения параметров ядерных за-

рядов по газообразным продуктам деления — изотопам криптона и ксенона, востребованный и в наше время для обнаружения запрещенных ядерных взрывов. А.А. Грепилов — победитель международного конкурса по разработке методов обнаружения тайно проведенных ядерных взрывов, объявленного правительством США. В 1968 г. предложил оригинальный метод измерения активности изотопа ксенона-133 в естественных смесях по его характеристическому рентгеновскому излучению.

В 1980-х годах под его руководством была разработана методика построения прогнозов и пятилетних планов для одной из отраслей народного хозяйства — отрасли «связь». Он предложил методы учета погрешностей всех исходных данных (конфлюэнтный анализ) при обработке результатов наблюдений и ряд методов решения некорректных задач, а также методы определения пеленгов радиоизлучения в пассивной пеленгации, которые позволили сократить время получения результата при одновременном повышении точности определения пеленгов.

В последние годы А.А. Грепилов уделяет большое внимание написанию и выпуску книг по математике с вложенными дисками с программными продуктами, с помощью которых можно решать задачи математического программирования.

А.А. Грепилов — автор более 200 научных работ, в том числе более 30 монографий, 30 авторских свидетельств и патентов в области разработки математических методов учета неопределенности исходной информации в задачах математической физики, распознавания образов, прогнозирования и в других технических приложениях. В настоящее время работает профессором на кафедре «Высшая математика» МГТУ им. Н.Э. Баумана.

Оглавление

Предисловие	3
Часть I. Математическое программирование	
Глава 1. Введение в математическое программирование	13
§ 1.1. Общие положения математического программирования	13
§ 1.2. Общая запись задачи математического программирования и ее виды	25
§ 1.3. Некоторые сведения об экстремуме функции, частных производных, градиенте и производной по направлению.....	27
§ 1.4. Особенности нахождения оптимальных решений в задачах математического программирования	31
§ 1.5. Необходимые и достаточные условия экстремума в задачах математического программирования	35
§ 1.6. Теория двойственности и недифференциальные условия оптимальности в задаче выпуклого программирования.....	40
§ 1.7. Графическое решение задач математического программи- рования.....	45
§ 1.8. Методы безусловной оптимизации.....	48
Глава 2. Линейное программирование	64
§ 2.1. Математическая постановка задачи линейного программи- рования.....	64
§ 2.2. Симплекс-метод — основной метод решения задач линей- ного программирования.....	67
§ 2.3. Метод полного исключения Жордана для решения систем линейных алгебраических уравнений	76
§ 2.4. Задача планирования выпуска продукции пошивочного предприятия	78
§ 2.5. Двойственность в задачах линейного программирования.....	85
§ 2.6. Задача оптимальной организации поставки грузов от по- ставщиков к потребителям (транспортная задача).....	93
§ 2.7. Задача о перевозках с перегрузкой	99
§ 2.8. Целочисленное линейное программирование	101
§ 2.9. Задача о назначениях (проблема выбора)	111
§ 2.10. Задачи о покрытии множества	115
§ 2.11. Дробно-линейное программирование.....	117
§ 2.12. Анализ устойчивости оптимального решения задачи ли- нейного программирования	121

Глава 3. Сетевые и потоковые задачи	131
§ 3.1. Основные определения и приложения сетевых и потоковых моделей	131
§ 3.2. Задача о покупке автомобиля	139
§ 3.3. Задача о многополюсной кратчайшей цепи	144
§ 3.4. Анализ сложности алгоритмов поиска кратчайших путей	149
§ 3.5. Венгерский алгоритм задачи о назначениях	150
§ 3.6. Задача размещения производства	155
§ 3.7. Задача о максимальном потоке	158
§ 3.8. Задача о многополюсном максимальном потоке	163
§ 3.9. Методы ветвей и границ. Задача коммивояжера	168
§ 3.10. Задача о многополюсной цепи с максимальной пропускной способностью	181
Глава 4. Основы динамического программирования и теории игр	185
§ 4.1. Условия применимости динамического программирования	185
§ 4.2. Задача об оптимальной загрузке транспортного средства неделимыми предметами	189
§ 4.3. Задача о вкладе средств в производство	194
§ 4.4. Задача о распределении средств поражения	199
§ 4.5. Вычислительные аспекты решения задач методом динамического программирования	204
§ 4.6. Теория игр. Игры в чистых стратегиях	205
§ 4.7. Поиск оптимальной смешанной стратегии	212
§ 4.8. Решение матричных игр размерностью $m \times n$	214
Глава 5. О развитии методов решения задач математического программирования	226
§ 5.1. Основные направления развития методов решения задач математического программирования	226
§ 5.2. Понятие о параметрическом программировании	227
§ 5.3. Многопродуктовые потоки в сетях	237
§ 5.4. Специальный класс целочисленных задач о многопродуктовом потоке	242
§ 5.5. Приближенное решение многопродуктовой транспортной задачи методом агрегирования	244
§ 5.6. Приложения задач о многопродуктовом потоке	247
§ 5.7. Эвристический алгоритм решения задачи синтеза сети связи	252
§ 5.8. Методы внутренней точки для решения задачи математического программирования	270
§ 5.9. Методы внешней точки для решения задачи математического программирования	273
§ 5.10. Комбинированный метод внутренней и внешней точек	276
§ 5.11. Метод проекции градиента	277

§ 5.12. Многокритериальные задачи линейного программирования	281
§ 5.13. Метод взвешенных сумм с точечным оцениванием весов ...	286
§ 5.14. Сжатие множества допустимых решений	289
§ 5.15. Минимальные значения критериев на множестве эффективных точек	292
§ 5.16. Параметризация целевой функции	293
§ 5.17. Целевое программирование.....	301

Часть II. Статистические методы принятия решений

Глава 6. Анализ методов принятия решений и постановка

задачи учета погрешностей признаков	309
§ 6.1. Основные понятия и определения	309
§ 6.2. Статистические задачи решения с наблюдениями	322
§ 6.3. Статистическая классификация при фиксированном объеме выборки	328
§ 6.4. Методы детерминистской классификации.....	333
§ 6.5. Последовательная решающая модель для классификации образов	335
§ 6.6. Байесовская последовательная решающая процедура.....	341
§ 6.7. Байесовские методы обучения	346
§ 6.8. Обучение с помощью стохастической аппроксимации	350
§ 6.9. Математическая постановка задачи учета погрешности признаков	353

Глава 7. Методы регрессионного и конфлюэнтного анализа

как инструмент в процедурах принятия решений.....	361
§ 7.1. Понятие регрессии. Основные определения	361
§ 7.2. Линейные регрессии η на ξ и ξ на η	366
§ 7.3. Регрессионный парадокс.....	368
§ 7.4. Ортогональная регрессия.....	369
§ 7.5. Метод наименьших квадратов. Оценка свободных параметров функций, линейных по параметрам	371
§ 7.6. Оценка параметров моделей с помощью функции правдо- подобия.....	379
§ 7.7. Байесовский подход к оцениванию параметров моделей.....	385
§ 7.8. Интервальные оценки линии регрессии и прогнозируемых значений функции	386
§ 7.9. Активный и пассивный эксперименты. Оценивание парамет- ров функции известного вида в пассивном эксперименте	390
§ 7.10. Анализ других методов оценки параметров функции известного вида с учетом ошибок в значениях функций и аргументов	396
§ 7.11. О единственности оценок параметров. Состоятельность оценок и алгоритм их получения.....	401
§ 7.12. Оценка параметров многомерной линейной модели	409

§ 7.13. Оценка параметров полиномиальной зависимости.....	411
§ 7.14. Оценка значений параметров в сигноме	413
§ 7.15. Анализ систем в активном эксперименте.....	416
Глава 8. Принятие решений по выборке фиксированного объема с учетом погрешности признаков	421
§ 8.1. Статистические свойства параметров функции Гаусса, определенных непосредственно и с помощью операций линеаризации	421
§ 8.2. Оценка параметров функции плотности распределения вероятностей с учетом погрешности вектора признаков.....	425
§ 8.3. Плохая обусловленность и некорректность в задачах оценки параметров функции	433
§ 8.4. Классификация образов по измеренному с ошибкой вектору признаков	441
§ 8.5. Классификация летательных аппаратов с учетом погрешностей в измерениях признаков	445
Глава 9. Распознавание образов при неизвестном законе распределения значений признаков	452
§ 9.1. Оценка параметров классификаторов по выборке фиксированного объема	452
§ 9.2. Обобщенные линейные разделяющие функции	457
§ 9.3. Оценка разделяющего вектора с помощью методов математического программирования	463
§ 9.4. Разделяющие функции для случая многих классов	466
§ 9.5. Учет погрешностей наблюдений при оценке значений параметров классификаторов.....	467
§ 9.6. Распознавание образов по измеренному вектору признаков	472
§ 9.7. Алгоритм идентификации объектов с учетом погрешности признаков	475
§ 9.8. Идентификация землетрясений и искусственных взрывов по сейсмическим проявлениям	483
§ 9.9. Учет интервальных оценок функций плотности вероятности в последовательных методах распознавания образов.....	486
§ 9.10. Сравнение зон неопределенности. Общий алгоритм принятия решений	491
Глава 10. Построение прогнозов.....	497
§ 10.1. Особенности процедуры прогнозирования.....	497
§ 10.2. Модели для получения прогнозов.....	505
§ 10.3. Сглаживание рядов с помощью скользящей средней	511
§ 10.4. Прогнозирование с помощью экспоненциального сглаживания.....	515
§ 10.5. Многофакторное прогнозирование.....	520
§ 10.6. Идентификация моделей типа АРПСС.....	524

§ 10.7. Методы уточнения прогнозов по модели АРПСС.....	529
§ 10.8. Байесовские прогнозы.....	535
§ 10.9. Анализ сезонных рядов.....	537
§ 10.10. Диагностическая проверка моделей и ошибка прогноза	542
§ 10.11. Пример прогнозирования газопотребления	549
Литература.....	557
Приложение 1. Описание программы «Регрессия» (инструкция для пользователя).....	563
Приложение 2. Программы для решения задач линейного про- граммирования	574
Приложение 3. Транспортная задача	586
Приложение 4. Задача о максимальном потоке	591
Приложение 5. Динамическое программирование, задача о рюкзаке.....	606
Приложение 6. Целочисленное линейное программирование	613
Приложение 7. Пример решения задачи линейного программи- рования двойственным симплекс-методом.....	615
Приложение 8. Краткий математический словарь.....	617
Список математических символов	635
Предметный указатель.....	636
К 75-летию автора.....	642

Учебное издание

Грешилов Анатолий Антонович

**МАТЕМАТИЧЕСКИЕ МЕТОДЫ
ПРИНЯТИЯ РЕШЕНИЙ**

Технический редактор *Э.А. Кулакова*

Корректор *А.Б. Сорокина*

Художник *А.К. Ездовой*

Компьютерная графика *М.А. Голуба*

Компьютерная верстка *Н.Ф. Бердавцевой*

В оформлении обложки использованы шрифты

Студии Артемия Лебедева

Оригинал-макет подготовлен

в Издательстве МГТУ им. Н.Э. Баумана.

Сертификат соответствия № РОСС RU. AE51. Н 16228 от 18.06.2012.

Подписано в печать 06.02.2014. Формат 60×90 1/16.

Усл. печ. л. 40,5. Тираж 1000 экз. (1-й з-д 1–500).

Заказ

Издательство МГТУ им. Н.Э. Баумана.

105005, Москва, 2-я Бауманская, д. 5, стр. 1.

press@bmstu.ru <http://www.baumanpress.ru>

Отпечатано в типографии МГТУ им. Н.Э. Баумана.

105005, Москва, 2-я Бауманская, д. 5, стр. 1.

baumanprint@gmail.com