

Л.С.Лэсдон

ОПТИМИЗАЦИЯ
БОЛЬШИХ
СИСТЕМ

•

Л. С. ЛЭСДОН

ОПТИМИЗАЦИЯ БОЛЬШИХ СИСТЕМ

Перевод с английского
Т. Н. Первозванской, Г. В. Шалабина

Под редакцией
А. А. Первозванского



ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1975

OPTIMIZATION THEORY
FOR LARGE SYSTEMS

Leon S. Lasdon
Case western reserve university

THE MACMILLAN COMPANY
COLLIER — MACMILLAN LIMITED, LONDON

Оптимизация больших систем, Л. Лэсдон. Главная редакция физико-математической литературы изд-ва «Наука», М., 1975, 432 стр.

Книга содержит достаточно полное и методически цельное изложение новых методов математического программирования, основанных на использовании специфики структуры связей в оптимизируемых больших системах. Дается детальное обоснование алгоритмов оптимизации, описываются численные примеры, приводятся сведения о полезных вычислительных приемах и результатах практического применения.

Книга представляет интерес для специалистов, студентов и аспирантов, работающих в области прикладной математики, в особенности экономической кибернетики и теории управления технологическими комплексами.

Табл. 50. Илл. 55. Библ. 140 назв.

ОГЛАВЛЕНИЕ

От редактора перевода	7
Предисловие автора к русскому переводу	10
Из предисловия автора	11
Глава I. Задачи большой размерности со специальной структурой матрицы условий	13
§ 1.1. Введение	13
§ 1.2. Анализ технологических способов	16
§ 1.3. Задачи планирования производства и управления запасами	19
§ 1.4. Динамическая модель затраты — выпуск (модель Леонтьева)	20
§ 1.5. Блочные задачи со связывающими переменными и ограничениями	27
§ 1.6. Задачи линейного программирования с большим числом строк или столбцов	33
§ 1.7. Нелинейные задачи со связующими переменными	42
§ 1.8. Частично-целочисленные задачи и проблема размещения	50
Задачи	54
Литература	54
Глава II. Метод декомпозиции Данцига — Вулфа	56
§ 2.1. Введение	56
§ 2.2. Теорема о выпуклых комбинациях	57
§ 2.3. Метод генерации столбцов	59
§ 2.4. Описание принципа декомпозиции	61
§ 2.5. Числовой пример использования принципа декомпозиции	69
§ 2.6. Экономическая интерпретация принципа разложения	75
§ 2.7. Нижняя граница значений величины минимальных затрат	78
§ 2.8. Использование метода декомпозиции для решения задач транспортного типа	81
§ 2.9. Обобщенные транспортные задачи и задача о рубке леса	84
§ 2.10. Оптимальное распределение ресурсов	88
2.10.1. Общая формулировка задачи	88
2.10.2. Модификация модели: определение размера партии выпуска и распределение рабочей силы	94
2.10.3. Результаты вычислительных экспериментов	99

§ 2.11. Применение метода одновременного решения прямой и двойственной задач к координирующей задаче	103
2.11.1. Дробно-линейное программирование	103
2.11.2. Применение метода одновременного решения прямой и двойственной задач к координирующей задаче	112
2.11.3. Числовой пример использования метода одновременного решения прямой и двойственной задач	116
§ 2.12. Сравнение трех алгоритмов решения координирующей задачи	122
Задачи и упражнения	125
Литература	127
Глава III. Решение задач линейного программирования с большим числом столбцов с использованием процедуры генерации столбцов	128
§ 3.1. Задача о раскрое	128
§ 3.2. Использование метода генерации столбцов при составлении производственных графиков	139
§ 3.3. Задача обобщенного линейного программирования	152
§ 3.4. Линеаризация на сетке и нелинейное программирование	165
3.4.1. Общее изложение	165
3.4.2. Нелинейный вариант принципа декомпозиции Данцига — Вулфа	174
§ 3.5. Расчет потоков в сетях	177
Задачи	185
Литература	187
Глава IV. Расчленение и релаксационные процедуры в линейном программировании	189
§ 4.1. Введение	189
§ 4.2. Релаксация	190
§ 4.3. Задачи со связывающими ограничениями и связывающими переменными	199
§ 4.4. Процедура расчленения Розена	207
4.4.1. Описание алгоритма	207
4.4.2. Соображения по поводу вычислений	215
4.4.3. Вычислительный опыт	220
4.4.4. Пример применения метода расчленения	221
Задачи	224
Литература	225
Глава V. Методы компактного обращения матриц	227
§ 5.1. Введение	227
§ 5.2. Модифицированный симплекс-метод с мультипликативным представлением обратной матрицы	228

§ 5.3. Методы, учитывающие двусторонние ограничения	243
§ 5.4. Задачи с обобщенными двусторонними ограничениями	249
5.4.1. Изложение алгоритма	249
5.4.2. Пример применения метода учета обобщенных двусторонних ограничений на переменные	260
§ 5.5. Обобщение на блочно-диагональную структуру	265
Задачи	279
Литература	280
Глава VI. Процедуры расчленения в нелинейном программировании	281
§ 6.1. Введение	281
§ 6.2. Алгоритм расчленения Розена для нелинейного программирования	283
6.2.1. Изложение алгоритма	284
6.2.2. Использование алгоритма для оптимизации работы нефтехимических установок	293
§ 6.3. Алгоритм расчленения Бендерса для частично-целочисленных задач	295
6.3.1. Описание алгоритма	295
6.3.2. Связь принципа декомпозиции и алгоритмов отсечения	305
6.3.3. Приложение к задаче о размещении складов	309
6.3.4. Численный пример	312
6.3.5. Вычислительный опыт	314
Задачи	316
Литература	317
Глава VII. Двойственность и декомпозиция в математическом программировании	319
§ 7.1. Введение	319
§ 7.2. Декомпозиция, использующая механизм цен	320
§ 7.3. Седловые точки функций Лагранжа	322
7.3.1. Основные теоремы	322
7.3.2. Теорема Эверетта	325
7.3.3. Применение к линейным целочисленным задачам	327
§ 7.4. Минимаксные двойственные задачи	330
§ 7.5. Дифференцируемость двойственной целевой функции	342
§ 7.6. Вычислительные методы решения двойственной задачи	351
§ 7.7. Специальные результаты для выпуклых задач	358
§ 7.8. Приложения	363
7.8.1. Задачи для связанных подсистем	363
7.8.2. Пример — оптимальное управление дискретной по времени динамической системой	369
7.8.3. Задачи, в которых допустимое множество конечно: многопродуктовые задачи распределения плана	372
Задачи	379
Литература	380

Глава VIII. Декомпозиция путем распределения ресурсов (разделение ограничений)	383
§ 8.1. Введение	383
§ 8.2. Формулировка проблемы	383
§ 8.3. Алгоритм возможных направлений для координирующей задачи	388
§ 8.4. Альтернативный подход к решению задачи выбора направления	398
§ 8.5. Аппроксимация касательными	404
Задачи	412
Литература	412
Приложение 1. Выпуклые функции и им сопряженные	413
Приложение 2. Субградиенты и производные по направлению выпуклых функций	420
Литература	429
Дополнительный список литературы	430

ОТ РЕДАКТОРА ПЕРЕВОДА

Публикуемая книга Л. С. Лэсдона посвящена вычислительным аспектам теории больших систем. Само несколько расплывчатое понятие «большая система» в ней заменено более узким и четким математическим представлением о системе, в которой

а) функциональные связи описываются набором большого числа уравнений и неравенств с относительно большим числом переменных,

б) в структуре функциональных связей имеется специфика, указывающая на возможность более или менее полного расчленения системы уравнений (и неравенств) на подсистемы меньшей размерности.

При такой трактовке проблемы теории оптимизации больших систем оказываются проблемами создания методов решения задач математического программирования большой размерности, использующими специфику структуры связей, и, вообще говоря, специфику критериальной функции.

В этом узком смысле трактуется и различие между «большой системой» и просто «системой»: система именуется «большой», если задача ее оптимизации не поддается решению «в лоб» стандартными методами математического программирования, не использующими специфики структуры.

Возможность решения той или иной задачи зависит, конечно, от возможностей используемой вычислительной техники, и те задачи, которые еще лет десять назад казались практически неразрешимыми, сейчас успешно решаются без каких-либо методических ухищрений. Однако требования повышения вычислительной эффективности остаются в силе и даже становятся более острыми из-за удорожания техники, а следовательно, практический интерес к методам, позволяющим резко ускорить решение задач, ни в коей мере не уменьшается.

Существует и еще одно важное обстоятельство: разработка «структурно ориентированных» методов оптимизации стимулирует разработку новых структур управления с частичной или полной децентрализацией процесса принятия решений.

Хотя такая проблематика отражена в данной книге лишь весьма ограничено, ознакомление с представленным материалом полезно для ее понимания. Это справедливо и в *позитивном* смысле, поскольку ряд методов, излагаемых в ней, поддается четкой интерпретации в духе теории решений, и в *негативном*, поскольку четкое формализованное изложение может способствовать рассеиванию накопившихся с момента появления принципа декомпозиции Данцига — Вулфа спекулятивных построений и легковесных «квазиэкономических» интерпретаций.

В книге охвачен и систематизирован большой алгоритмический материал, овладение которым будет, несомненно, полезно для математиков-вычислителей. Вместе с тем, она ни в коей мере не может претендовать на полный охват проблематики, соответствующей ее названию.

Во-первых, изложение в основном ограничено процедурами, в той или иной мере примыкающими к классическим схемам линейного программирования — прямому и двойственному симплекс-методу. Бесконечно-шаговые процедуры, связанные с нахождением седловой точки функции Лагранжа и модифицированной функции Лагранжа, или с введением штрафных функций совсем не освещены, либо освещены весьма кратко. Вместе с тем, эти процедуры, восходящие к работам Эрроу, Гурвица и Удзавы, в настоящее время начинают приобретать серьезное теоретическое и практическое значение.

Во-вторых, совсем вне рассмотрения оставлена проблематика, связанная с информационными аспектами теории управления большими системами (отсутствие полной информации в центре принятия решений, запаздывание в передаче и обработке информации, эффективность агрегирования и т. п.).

В определенной мере указанные пробелы связаны с тем, что автор книги, по-видимому, мало знаком с результатами советских авторов, полученными в последние годы. Возможно, однако, что дополнительное рассмотрение указанных выше аспектов привело бы к чрезмерному увеличению объема книги и к потере

той методической цельности, которой она сейчас, несомненно, обладает.

При переводе объем несколько сокращен за счет изъятия вводной главы, содержавшей хорошо известные общие факты математического программирования. Можно надеяться, что это изъятие не помешает усвоению основного материала, поскольку почти все изложение во вводной главе и по духу, и даже по обозначениям следовало изданной в СССР книге Дж. Данцига «Линейное программирование, его обобщения и применения». Предполагалось, что читатели знакомы с содержанием хотя бы первой половины этого прекрасного учебника, и считалось, что данная монография может рассматриваться как специализированное дополнение к нему.

В заключение стоит отметить, что хотя перевод осуществлялся людьми, давно работающими в этой области и следящими за периодической литературой, переводчики обнаружили в книге много совершенно новых и весьма полезных сведений. Тем более, можно надеяться, что книга Л. Лэсдона окажется полезной для всех, кто начинает работать над интересной и важной проблематикой теории больших систем, в особенности для специалистов по прикладной математике и экономической кибернетике, а также студентов соответствующих факультетов.

А. Первозванский

ПРЕДИСЛОВИЕ АВТОРА К РУССКОМУ ПЕРЕВОДУ

Я весьма удовлетворен тем, что моя книга будет издана в СССР. Советские ученые внесли огромный вклад в изучение систем большой размерности. Вместе с тем, я надеюсь, что они найдут содержание этой книги интересным и что ее появление хотя бы в малой степени будет способствовать техническому прогрессу в такой стране, как СССР, являющейся одним из мировых лидеров в технологии. Я надеюсь также, что будучи одной из компонент возрастающего потока двухстороннего научного обмена, книга послужит и дальнейшему сближению ученых Соединенных Штатов и СССР.

Я хотел бы выразить мою признательность проф. А. А. Первозванскому, Т. Н. Первозванской и Г. В. Шалабину за осуществление перевода, который был нелегкой задачей.

Л. Лэсдон

ИЗ ПРЕДИСЛОВИЯ АВТОРА

Многие проблемы выбора решений удастся сформулировать в виде задач математического программирования, состоящих в максимизации или минимизации целевой функции при заданных ограничениях. Такие задачи зачастую имеют специальную структуру. В частности, в задачах линейного программирования ненулевые элементы матриц ограничений могут появляться лишь в диагональных блоках и в относительно малом числе связующих строк или столбцов.

Задачи нелинейного программирования часто имеют структуру, позволяющую путем фиксации значений некоторых переменных превращать их в линейные. Характерным является и появление аддитивной сепарабельности функций, определяющих критерий и ограничения.

Некоторая специфическая структура почти всегда обнаруживается в действительно больших задачах, которые обычно возникают в результате учета связей, существующих между независимыми подсистемами во времени или пространстве. Создание специализированных алгоритмов решения, учитывающих структурную специфику, позволяет достичь значительного выигрыша в вычислительной эффективности и сокращения требований к объему памяти ЦВМ. Подобные алгоритмы открывают путь к решению действительно больших задач, которые не поддаются стандартным методам в силу ограничений по времени или объему памяти.

За последнее десятилетие было осуществлено выделение многих характерных структур и создано большое число алгоритмов для их решения. Этот период начался с опубликования в 1960 году принципа декомпозиции Данцига — Вулфа. Хотя значительная активность наблюдалась и до этого, дальнейший ряд исследований носил «взрывной» характер. Быстро были созданы методы сначала для линейных, а затем и для некоторых классов нелинейных задач. Соответствующая литература теперь очень велика, что вынуждает исследователя, начинающего работать в этой области, вести поиски массы статей во многих журналах. Более того, даже после обнаружения этих статей могут

оказаться неясными связи, существующие между различными методами. Стала очевидной необходимость собрать и объединить лучшее из опубликованного.

Данная книга и представляет попытку выполнить эту задачу. В ней рассматриваются некоторые из наиболее важных алгоритмов оптимизации больших систем. Особое значение придается логическому построению различных методов на базе малого числа основных идей и принципов, что позволяет выяснить соотношения между разнообразными процедурами. Применение многих методов иллюстрируется численными примерами. Усвоению их будет способствовать самостоятельное решение читателями задач, приведенных в конце каждой главы. Описан и существующий вычислительный опыт по применению методов, а также замечания об использовании дополнительных вычислительных приемов. Эта информация должна оказаться полезной для читателей, интересующихся приложениями. Той же цели служат параграфы, описывающие содержательные формулировки задач. Первая глава полностью посвящена изложению того, как возникают задачи математического программирования большой размерности при исследовании реальных проблем. Ряд других примеров постановок задач дается во всех остальных главах.

Приложения в книге содержат основные сведения о выпуклых функциях и им сопряженных, а также о субградиентах и производных по направлению выпуклых функций. Они основаны на работе Р. Т. Рокафеллара и представляют определенный интерес сами по себе. Содержащиеся в них идеи интенсивно используются в главах 7 и 8 и лишь в очень малой степени в остальной части книги.

Л. Лэсдон

ГЛАВА I

ЗАДАЧИ БОЛЬШОЙ РАЗМЕРНОСТИ СО СПЕЦИАЛЬНОЙ СТРУКТУРОЙ МАТРИЦЫ УСЛОВИЙ

§ 1.1. ВВЕДЕНИЕ

Решение многих задач, описывающих функционирование реальных систем, часто оказывается затруднительным из-за их большой размерности. Размерность задачи математического программирования определяется количеством переменных, числом и характером ограничений, а также видом целевой функции. При этом действительное содержание термина «задача большой размерности» обычно зависит от возможностей вычислительных алгоритмов, скорости, емкости оперативной памяти и других характеристик используемых ЭВМ. Однако на практике часто возникают задачи, решение которых может быть с трудом осуществлено даже с использованием самых совершенных алгоритмов и ЭВМ. Примером является задача линейного программирования, матрица условий которой имеет свыше 4 тыс. строк и значительно большее число столбцов. Такая задача может быть решена модифицированным симплекс-методом с использованием системы IBM MPS/360. При этом необходимо хранить в памяти машины 248 000 32-х разрядных позиций (слов), записанных на магнитной ленте. Д. Данцигу [1] удалось решить задачу с 10^4 уравнениями и 10^6 переменными. Однако для этого ему потребовалось создать специальный алгоритм, учитывающий специфические особенности структуры матрицы условий.

Вообще, специфичность структуры матрицы условий является характерной особенностью задач большой размерности. Эти структурные особенности определяются соотношением между числом нулевых и ненулевых элементов матрицы, составленной из коэффициентов при неизвестных (переменных) задачи (речь идет о задачах линейного программирования). Плотность заполнения такой матрицы ненулевыми элементами для многих задач большой размерности не превышает нескольких процентов. Кроме того, часто ненулевые элементы расположены в матрице особым образом, например, вдоль главной диагонали.

Иногда в задачах с большим числом строк или столбцов (или же строк и столбцов одновременно) множество строк (столбцов) можно генерировать в процессе вычислений, что позволяет осуществлять операцию поиска соответствующего столбца (строки) более эффективно.

Формирование информации для задач большой размерности, не обладающих «хорошими» структурными особенностями, является почти неразрешимой проблемой. Например, для задачи с плотно заполненной матрицей условий размерности $(10^4 \times 10^6)$ *) требуется собрать и обработать 10^{10} чисел!

Значительно труднее выявить структурные особенности нелинейных задач большой размерности. Можно, например, структуру нелинейной задачи характеризовать специальной матрицей, элемент которой (i, j) принимает значение 0 или 1 в зависимости от наличия переменной « i » в ограничении « j ». При этом важными характеристиками являются общее число линейных переменных и выполнение свойства аддитивности для целевой функции.

Особый интерес представляют так называемые «смешанные задачи», т. е. задачи, которые становятся линейными при фиксации значений некоторых нелинейных переменных.

Можно выделить два основных подхода к решению задач большой размерности: прямые методы и методы, основанные на идеях декомпозиции (разложения).

В первом случае для решения задач специального вида используются общие методы математического программирования (например, симплекс-метод для решения линейных задач). При этом основные вычислительные трудности связаны с обращением матрицы базиса. Если матрица условий задачи имеет специальную структуру, то необходимые вычисления могут быть выполнены с использованием матрицы меньшей размерности. Такой подход, например, реализуется при применении метода учета двусторонних ограничений [2] или метода приведения базиса к треугольному виду [3].

Методы, принадлежащие ко второму классу, основаны на разложении исходной системы на подсистемы, для каждой из которых необходимо решать подзадачу меньшей размерности (по сравнению с исходной задачей). Так как подсистемы взаимосвязаны, то общее решение, как правило, нельзя получить в результате изолированного решения таких подзадач. Известен метод [4], когда выделяются подсистемы первого и второго уровней таким образом, что подсистемы второго уровня определяют соответствующие изменения в подсистемах первого уров-

*) Как уже отмечалось, размерность задачи линейного программирования определяется числом строк и столбцов матрицы условий.

ня. Учет связей, четкое выявление которых является обязательным при любом способе разбиения системы, может осуществляться в различных формах. Последние в основном определяются характером исходной задачи и методом выделения подсистем первого, второго и др. уровней. Основная задача

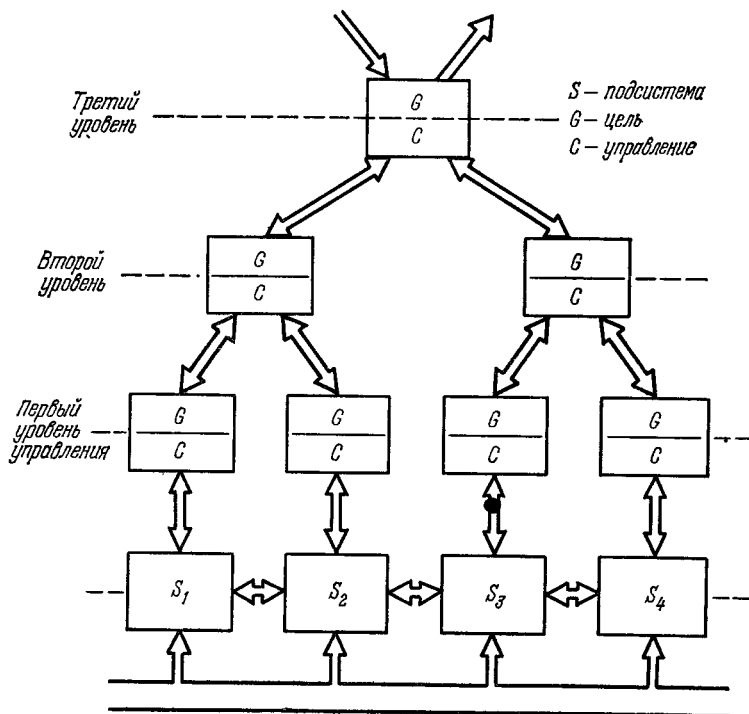


Рис. 1.1. Структура многоуровневой системы управления.

второго уровня состоит в координации функционирования элементов первого уровня с целью получения общего решения исходной задачи. Нетрудно понять, что такой характер взаимодействия является типичным для крупных организаций. Можно идти дальше и выделить подсистемы третьего уровня, координирующие функционирование подсистем второго уровня и т. д. В результате мы приходим к системе принятия решений, имеющей пирамидальную (иерархическую) структуру (рис. 1.1).

Все известные в настоящее время методы разложения по своей структуре являются двухуровневыми. Координирующий «орган» влияет на решение подзадач путем изменения значений коэффициентов целевой функции, введением дополнительных

ограничений и т. д. Преимущества такого подхода очевидны. Они заключаются в возможности использования специальных алгоритмов для решения подзадач, в создании и отладке программ для ЭВМ для каждой из них независимо друг от друга и в значительной экономии оперативной памяти машин. Главный же недостаток метода состоит в необходимости многократного возвращения к решению отдельных подзадач. А это в свою очередь ведет к значительным затратам машинного времени.

Основная часть нашей книги посвящена рассмотрению методов решения различных задач большой размерности. Анализ этих методов целесообразнее всего начать с характеристики основных типов таких задач, так как именно структура задачи определяет особенности используемых для их решения вычислительных методов.

Для этого мы рассмотрим ряд примеров. При этом часто бывает полезным абстрагироваться от конкретной физической ситуации, используя для анализа более общий подход и соответствующую ему терминологию. Такой подход основан на использовании понятия *технологического способа*, что обеспечивает весьма удобный и продуктивный прием построения моделей.

§ 1.2. АНАЛИЗ ТЕХНОЛОГИЧЕСКИХ СПОСОБОВ

Изучая реально функционирующие системы, специалист в области математического программирования прежде всего выделяет в ней основные технологические способы. Под этим он понимает различные неагрегированные операции (виды работ), без выполнения которых невозможно функционирование изучаемой системы. Каждый из таких способов можно рассматривать в качестве «черного ящика» с входящими и выходящими потоками затрат (различные виды труда, сырья, материалов и др. ресурсов) и выпуска (конечной продукции, услуг, прибыли и т. д.). Различные виды затрат и выпуска будем называть *ингредиентами*, а качественные показатели, характеризующие изменение этих ингредиентов — *интенсивностями использования технологических способов*.

Чтобы лучше разобраться в этих понятиях, рассмотрим в качестве примера транспортную задачу. Смысл этой задачи заключается в нахождении плана перевозок (поставок) некоторой однородной продукции из пунктов складирования в пункты потребления. При этом спрос на продукцию должен быть удовлетворен полностью, а суммарные транспортные затраты должны быть минимальными. В этой задаче понятию технологического способа соответствует перевозка единицы груза со склада «*i*» в пункт «*j*». Показатель интенсивности использования этого способа есть число x_{ij} — количество продукта, перевезенного со

склада « i » в пункт потребления « j ». Ингредиентами способа, соответствующими затратами, в данном случае являются отправляемая продукция и деньги, а ингредиентами выпуска — доставленная в пункт « j » продукция. Рассмотренный технологический способ можно представить графически в виде рис. 1.2.

Нетрудно понять, что в рассматриваемой задаче имеется множество технологических способов перевозок. Все способы перевозок взаимосвязаны в силу наличия ограничений на объемы потребления и емкости складов.

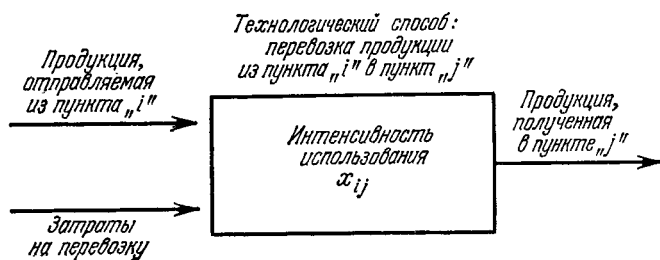


Рис. 1.2. Технологический способ перевозки.

В задачах линейного программирования относительно технологических способов должны выполняться следующие предположения [5]:

Предположение 1 (пропорциональность).

Затраты (выпуск) каждого ингредиента пропорциональны интенсивности его использования.

Предположение 2 (неотрицательность). Показатели интенсивности использования технологических способов являются неотрицательными числами.

Предположение 3 (аддитивность).

Общее количество израсходованных ресурсов или выпущенной продукции равно сумме соответствующих ингредиентов по всем используемым технологическим способам.

Предположение 4 (линейность целевой функции).

Целевая функция является линейной функцией (функционалом) интенсивностей использования технологических способов, нахождение максимума (минимума) которой является целью решения задачи.

В соответствии с предположением 1, если x_j — интенсивность использования j -го способа, то затраты (выпуск) i -го ингредиента будут равны a_{ij} , где a_{ij} — коэффициент затрат (выпуска) i -го ингредиента при использовании j -го способа с единичной интенсивностью. Для определенности будем предполагать, что если

i — расходуемый фактор технологического процесса, то $a_{ij} > 0$; если i соответствует ингредиенту выпуска, то $a_{ij} < 0$. Поэтому любой технологический способ можно описать с помощью m -мерного вектора

$$p_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}. \quad (1)$$

Предположение 2 означает, что

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

Для многих практических задач характерным является наличие жестких ограничений на используемые ресурсы и объемы выполнения полезной работы (например, в виде требования удовлетворения спроса на определенный вид продукции). Эти ограничения, учитывая предположение 3, записываются в форме следующего основного балансового ограничения:

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad (3)$$

где b_i — ограничение на потребление (выпуск) i -го ингредиента. (Если ограничения не являются жесткими, то им соответствует система неравенств, которые в результате введения дополнительных переменных могут быть приведены к виду (3).)

Обозначим теперь коэффициенты целевой функции через c_j . Тогда целевую функцию можно записать так:

$$z = \sum_{j=1}^n c_j x_j, \quad (4)$$

причем для определенности будем ее минимизировать. Если условия 1—4 не выполняются, то возникает задача более общего вида:

$$\sum_j f_j(x_j) = \min, \quad (5)$$

$$\sum_j g_{ij}(x_j) = b_i \quad (i = 1, \dots, m), \quad (6)$$

$$x_{ij} \geq 0, \quad (7)$$

где теперь f_j и g_{ij} — нелинейные функции.

§ 1.3. ЗАДАЧИ ПЛАНИРОВАНИЯ ПРОИЗВОДСТВА И УПРАВЛЕНИЯ ЗАПАСАМИ

Предположим, что необходимо найти распределение выпуска двух различных продуктов по кварталам года так, чтобы полностью удовлетворить спрос на эти продукты, а общие затраты на производство и хранение (складирование) продукции были бы минимальными. При этом необходимо выполнить ограничения на использование рабочей силы и емкость склада. Таким образом, расходуемыми ингредиентами технологических способов являются рабочая сила, производственные мощности и текущие издержки, а выпуск представлен двумя продуктами, доставленными потребителям. Система в каждом периоде в квартале может быть описана с помощью следующих шести технологических способов:

Технологический способ	Интенсивность использования	Единица измерения интенсивности использования
Производство 1-го продукта	$x_{1t} \geq 0$	единица 1-го продукта
Производство 2-го продукта	$x_{2t} \geq 0$	единица 2-го продукта
Хранение 1-го продукта	$y_{1t} \geq 0$	единица 1-го продукта
Хранение 2-го продукта	$y_{2t} \geq 0$	единица 2-го продукта
Недоиспользование емкости склада	$s_{1t} \geq 0$	единица емкости
Недоиспользование рабочей силы	$s_{2t} \geq 0$	человеко-час

Введем обозначения:

a_i — емкость склада, необходимая для хранения единицы i -го продукта;

v_{it} — затраты (в долларах) на хранение единицы i -го продукта в t -м периоде (квартале);

b_i — удельная трудоемкость (в человеко-часах) производства i -го продукта;

c_{it} — оплата 1 человеко-часов, используемого для производства i -го продукта в t -м периоде;

l_t — лимит на использование рабочей силы в t -м периоде;

f_t — общая емкость склада в t -м периоде;

d_{it} — прогноз спроса на i -й продукт в t -м периоде.

Теперь ограничения задачи для каждого периода могут быть записаны следующим образом:

ограничение на использование рабочей силы:

$$b_1 x_{1t} + b_2 x_{2t} + s_{2t} = l_t; \quad (1)$$

ограничения на емкость склада:

$$a_1 y_{1t} + a_2 y_{2t} + s_{1t} = f_t; \quad (2)$$

уравнение для определения величины запаса:

$$y_{it} = y_{it-1} + x_{it} - d_{it}; \quad (3)$$

уравнение для определения суммарных затрат на производство и хранение продукции:

$$z_t = v_{1t}y_{1t} + v_{2t}y_{2t} + c_{1t}b_1x_{1t} + c_{2t}b_2x_{2t}. \quad (4)$$

Эти ограничения можно записать в матричной форме (табл. 1.3.1). Из таблицы видно, что только в ограничения типа

Таблица 1.3.1

Ингредиенты	Технологические способы							
	Хранение продукта 1 в периоде $t-1$: $y_{1,t-1}$	Хранение продукта 2 в периоде $t-1$: $y_{2,t-1}$	Хранение продукта 1 в периоде t : y_{1t}	Хранение продукта 2 в периоде t : y_{2t}	Производство продукта 1 в периоде x_{1t}	Производство продукта 2 в периоде x_{2t}	Недоиспользование емкости склада s_{1t}	Недоиспользование рабочей силы s_{2t}
Рабочая сила					b_1	b_2		
Емкость склада			a_1	a_2			1	
Продукт 1			1		-1			l_t
Продукт 2		-1		1		-1		f_t
Издержки			v_{1t}	v_{2t}	$c_{1t}b_1$	$c_{2t}b_2$		$-d_{1t}$
								$-d_{2t}$
								0

(3) входят неизвестные, относящиеся к разным периодам времени. Поэтому общая матрица условий рассматриваемой задачи имеет вид, представленный в таблице 1.3.2. Эта матрица имеет специальную структуру: в ней элементы, отличные от нуля, располагаются блоками вдоль главной диагонали. Блоки соответствуют производственно-технологическим способам каждого периода. Связь блоков осуществляется через показатели переходящих запасов продукции.

Такая структура матрицы является типичной для многих динамических задач наилучшего распределения ограниченных ресурсов.

§ 1.4. ДИНАМИЧЕСКАЯ МОДЕЛЬ ЗАТРАТЫ — ВЫПУСК (МОДЕЛЬ ЛЕОНТЬЕВА)

Схема «затраты — выпуск», впервые предложенная В. В. Леонтьевым, нашла широкое применение в планировании и прогнозировании. Простейший статический вариант в этой схеме (модели) включает n технологических способов, соответствующих отдельным отраслям или группам отраслей. Предполагается,

что эти способы удовлетворяют условиям 1—3, сформулированных нами в § 1.2. При использовании технологических способов производится только один продукт, который может потребляться внутри системы или выступать в качестве конечного продукта, спрос на который в форме индивидуального или общественного потребления, а также экспорта предполагается заданным.

Пусть x_i — интенсивность использования i -го технологического способа — общий (валовый) выпуск продукции i -й отрасли; d_i — спрос (конечный продукт) на продукцию i -й отрасли; a_{ij} — количество продукции i -й отрасли, необходимое для производства единицы продукции j -й отрасли.

Основное балансовое уравнение, описывающее распределение продукции отрасли, имеет вид

$$x - \sum_{j=1}^n a_{ij} x_j = d_i \quad (i=1, \dots, n) \quad (1)$$

или, в матричной форме,

$$(I - A)x = d, \quad (2)$$

где A — технологическая матрица, составленная из элементов a_{ij} , I — единичная матрица. Для того чтобы определить значения неизвестных x_i , необходимо найти неотрицательное решение линейной системы (1), состоящей из n уравнений и n неизвестных. Как уже указывалось выше, такая система либо имеет единственное неотрицательное решение либо такого решения не существует вообще. Вагнер [6] рассмотрел динамические варианты основной модели (1), допускающие наличие нескольких решений. В исследованных им модификациях модели предусматривалась возможность ввода новых производственных мощностей и наличие переходящих запасов продукции. Естественно, что при этом возникают различные варианты функционирования системы, что значительно расширяет свободу выбора окончательного решения, а введение целевой функции позволяет выбрать из всех возможных решений оптимальное.

Будем предполагать, что так же, как и в случае использования статической модели (4), нам известна технологическая матрица A . Кроме того, задана квадратная $(n \times n)$ -матрица капитальных затрат B , элементы которой b_{ij} обозначают затраты продукции i -й отрасли на увеличение выпуска на одну единицу в j -й отрасли. Элементы матриц A и B под влиянием технического прогресса изменяются во времени.

Пусть d_t — конечный спрос на продукцию в t -м периоде; c_t — производственная мощность в конце периода t (измеряется в единицах выпуска продукции); x_t — выпуск продукции в перио-

де t ; n_t — увеличение (прирост) мощности в периоде t ; s_t — уровень запаса продукции в конце t -го периода; u_t — недоиспользованная производственная мощность в течение периода t .

Теперь основное балансовое уравнение для динамической модели «затраты — выпуск» примет следующий вид:

$$x_t = Ax_t + d_t + Bn_t + s_t - s_{t-1}, \quad t = 1, \dots, T. \quad (3)$$

Производственная мощность, которая может быть использована в периоде t , равна сумме мощности предшествующего периода и прироста (увеличению) мощности, т. е.

$$c_t = c_{t-1} + n_t \quad (4)$$

или

$$c_t = c_0 + \sum_{\tau=1}^t n_\tau. \quad (5)$$

Выпуск продукции в периоде t ограничен существующей мощностью, так что $x_t \leq c_t$ или, вводя дополнительную переменную, получим

$$x_t + u_t = c_t. \quad (6)$$

Все неизвестные, очевидно, должны быть неотрицательными:

$$d_t, c_t, x_t, n_t, s_t, u_t \geq 0. \quad (7)$$

Чтобы более детально рассмотреть структуру этой динамической модели, предположим, что являются известными: c_0 — начальная мощность, s_0 — начальный запас и конечный спрос на продукцию для всех периодов планового горизонта — $d_1, d_2, \dots, d_T$. Теперь, используя уравнение (5) для исключения переменных c_t , представим условия (3) и (6) в виде специальной таблицы (табл. 1.4.1).

Матрица коэффициентов имеет блочно-треугольную структуру, сходную со структурой матрицы, рассмотренной в § 1.3, но по сравнению с ней является более сложной. В этой матрице также имеются блоки, расположенные вдоль главной диагонали и связанные между собой через технологические способы хранения продукции. Но так как производственная мощность, введенная в t -м периоде, может быть использована во всех последующих периодах, столбец, соответствующий переменной n_t , содержит ненулевые элементы для периодов $t, t+1, \dots, T$.

Если количество технологических способов нового строительства и хранения (запасания) продукции значительно меньше числа технологических способов, соответствующих производству продукции и недоиспользованию мощности, тогда, фиксируя

Т а б л и ц а 1.4.1

Технологические способы												Ингредиенты
x_1	u_1	n_1	s_1	x_2	u_2	n_2	s_2	x_3	u_3	n_3	s_3	
$(I-A)$		$-B$	$-I$									d_1-s_0
I	I	$-I$										c_0
			I	$(I-A)$		$-B$	$-I$					d_2
		$-I$		I	I	$-I$						c_0
							I	$I-A$		$-B$	$-I$	d_3
		$-I$			$-I$			I	I	$-I$		c_0
		$-I$			$-I$					$-I$		$\downarrow$
		$\downarrow$			$\downarrow$					$\downarrow$		$\downarrow$

значения интенсивностей этих технологических способов, можно привести матрицу задачи к блочно-диагональному виду. Такую систему обычно называют «слабо связанной». Для соответствующей матрицы характерно наличие малого числа ненулевых элементов, расположенных вне диагональных блоков.

В связи с рассмотрением таких моделей возникает много интересных вопросов. Один из них связан с выявлением условий существования допустимого решения системы (3)–(6). Простейший вариант этой проблемы заключается в следующем: может ли быть удовлетворен заданный спрос $d_1, \dots, d_T$, когда известны начальная мощность c_0 и начальный запас продукции s_0 . Ответ, понятно, будет положительным, если начальный запас таков, что за счет его может быть удовлетворен спрос для всех периодов. Тогда не возникает необходимости в увеличении мощности и образовании новых запасов и поэтому, положив

$$n_t = s_t = 0, \quad t = 1, \dots, T$$

в уравнении (3), получим

$$x_1 = (I - A)^{-1}(d - S_0), \quad (8)$$

$$x_t = (I - A)^{-1}d_t, \quad (9)$$

$$t = 2, \dots, T).$$

Эти значения переменных x_1 и x_t ($t = 2, \dots, T$) являются допустимыми, если значение переменной u_t в уравнении (6) является

неотрицательным, т. е. если

$$c_0 \geq (I - A)^{-1}(d_1 - s_0), \quad (10)$$

$$c_0 \geq (I - A)^{-1}d_t \quad (t = 2, \dots, T). \quad (11)$$

Нетрудно понять, что спрос на продукцию не может быть удовлетворен, если начальная мощность и начальный запас недостаточны для удовлетворения спроса первого периода, т. е. если

$$c_0 \not\geq (I - A)^{-1}(d_1 - s_0). \quad (12)$$

Но, конечно, с наиболее интересным и трудным случаем мы встречаемся, когда условие (10) выполняется, а условия (11) нарушаются для некоторых значений $t \geq 2$. В этом случае необходимо наращивать производственные мощности и создавать запасы продукции с целью полного удовлетворения спроса.

Напомним, что начальная фаза*) симплекс-метода может быть использована для того, чтобы найти допустимый план или доказать отсутствие допустимого решения системы. Другой вопрос, связанный с исследованием существования допустимого решения, касается проблемы определения максимального уровня спроса для одного или нескольких периодов, который может быть удовлетворен.

В качестве примера рассмотрим следующую ситуацию. Предположим, что необходимо увеличить конечный спрос на продукцию первой и третьей отраслей в периоде t , причем это увеличение должно происходить всегда в отношении 2:1. Чтобы выяснить возможность такого увеличения, необходимо ввести дополнительный способ — вектор с компонентами, равными —2 и —1 в первой и третьей строке соответственно и нулю во всех остальных случаях, затем решить задачу, стремясь максимизировать показатель интенсивности использования этого дополнительного способа. В случае увеличения спроса в каждом периоде необходимо вводить уже не один, а несколько (по числу периодов) дополнительных технологических способов, используя заданные отношения увеличения спроса в качестве коэффициентов при неизвестных в целевой функции.

Подобная ситуация характерна и для стратегических задач планирования обороны. Предположим, что задан минимально необходимый уровень производства продукции для нужд гражданского населения. Необходимо с помощью оставшихся ресурсов произвести максимальное количество военного снаряжения.

*) Здесь и далее имеются в виду стандартные формы записи задачи линейного программирования и описания симплекс-метода, представленные в книге: Дж. Данциг, *Линейное программирование, его обобщения и приложения*, «Прогресс», М., 1966 г.

Чтобы решить эту задачу, прежде всего необходимо ввести в рассмотрение новый технологический способ с учетом затрат продукции каждой отрасли на производство единицы военной продукции. Этот способ записывается в правой части балансового уравнения (3). Если необходимо максимизировать общий выпуск военной продукции за T периодов, то коэффициенты при неизвестных в целевой функции полагаются равными единице. Если при этом для отдельных периодов устанавливается приоритет на выпуск продукции в виде, например, некоторой шкалы предпочтения, то в соответствии с этой шкалой необходимо рассчитать новые коэффициенты целевой функции. Другие сходные задачи рассмотрены в уже упоминавшейся работе Вагнера [6]. Для всех этих задач характерным является неединственность допустимого решения (если, конечно, оно существует). Это связано с тем, что в рассматриваемой модели число неизвестных превышает число ограничений. Выбор наилучшего решения, как мы уже подчеркивали, осуществляется с помощью целевой функции.

Предположим, что мы заинтересованы в минимизации затрат рабочего времени, необходимого для производства продукции с целью удовлетворения заданного потребительского спроса. Пусть h_{1t} — вектор, компоненты которого суть затраты в часах рабочего времени на производство единицы каждого вида продукции в периоде t , и h_{2t} — аналогичный вектор затрат труда в периоде t на создание единицы каждого вида новых производственных мощностей. Целевая функция, которую следует минимизировать, имеет вид

$$z = \sum_t (h_{1t}x_t + h_{2t}n_t). \quad (13)$$

Уже отмечалось, что с точки зрения существования допустимого плана большое значение имеет величина начальной производственной мощности c_0 . Предположим, что она такова, что не существует плана, в котором за счет этой мощности спрос мог бы быть удовлетворен во все периоды времени. В этом случае может возникнуть вопрос о минимальной начальной мощности, обеспечивающей существование допустимого плана задачи. Для ответа на этот вопрос необходимо поместить c_0 в левую часть таблицы 1.4.1 и записать целевую функцию в виде

$$z = v_0 c_0, \quad (14)$$

где v_0 — вектор затрат, соответствующий вектору c_0 . Такая задача, например, может возникнуть в период предвоенной мобилизации, когда правительству желательно оценить прирост мощностей, необходимый для удовлетворения требований на военную продукцию, которые возникнут в будущем.

§ 1.5. БЛОЧНЫЕ ЗАДАЧИ СО СВЯЗЫВАЮЩИМИ ПЕРЕМЕННЫМИ И ОГРАНИЧЕНИЯМИ

Пусть имеется компания (фирма) в составе двух предприятий, производящих соответственно два или три различных вида продукции. Каждое предприятие для производства продукции использует свои внутренние, локальные ресурсы, например, рабочую силу и оборудование. Предприятия связаны между собой, совместно потребляя некоторый лимитированный ресурс (сырье). Обозначим лимиты на использование локальных ресурсов для первого предприятия через b_1 и b_2 , а через b_3 — лимит на использование ресурса для второго предприятия. Пусть теперь b_0 — ограничение на использование общего ресурса. Тогда возникает следующая задача:

$$\min \{z = c_1x_1 + c_2x_2 + d_1y_1 + d_2y_2 + d_3y_3\}$$

при ограничениях

$$\begin{aligned} & \boxed{a_{11}x_1 + a_{12}x_2 + a_{13}y_1 + a_{14}y_2 + a_{15}y_3} \leq b_0, \\ & \boxed{\begin{matrix} a_{21}x_1 + a_{22}x_2 \\ a_{31}x_1 + a_{32}x_2 \end{matrix}} \begin{matrix} \leq b_1, \\ \leq b_2 \end{matrix} \quad (1) \\ & \boxed{a_{41}y_1 + a_{42}y_2 + a_{43}y_3} \leq b_3, \\ & x_i \geq 0, \quad y_i \geq 0, \end{aligned}$$

где x_i и y_i — объемы производства различных видов продукции на первом и втором предприятиях; a_{ij} — количество i -го ресурса, необходимого для производства единицы j -го вида продукции.

Общая система ограничений задачи состоит из двух подмножеств ограничений, соответствующих условиям производства на каждом предприятии, и одного связывающего ограничения на использование общего ограниченного ресурса.

Задача (1) является частным случаем следующей более общей задачи:

$$\begin{aligned} & \min \{z = c_1x_1 + c_2x_2 + \dots + c_px_p\} \\ & \text{при условиях} \\ & \begin{matrix} A_1x_1 + A_2x_2 + \dots + A_px_p = b_0, \\ B_1x_1 & & & = b_1, \\ & B_2x_2 & & = b_2, \\ & & \ddots & \\ & & & B_px_p = b_p, \\ x_1 \geq 0, x_2 \geq 0, \dots, & & x_p \geq 0, \end{matrix} \quad (2) \end{aligned}$$

где x_i , b_i — векторы-столбцы. Элементами матриц B_i ($i=1, \dots, p$) являются коэффициенты затрат — выпуска ингредиентов, связанных только с использованием технологических способов x_i , в то время как b_i задает объемы ингредиентов, разделяемых между всеми способами. Матрица A_i содержит коэффициенты затрат — выпуска этих ингредиентов. Структуры, подобные (2), называют *блочно-треугольными*. Если число общих связывающих ограничений системы невелико, то такую систему называют *слабо связанной*. Анализ таких систем может быть эффективно осуществлен с помощью методов разложения.

Часто задача линейного программирования имеет большую размерность из-за того, что изучаемая система представляет совокупность однотипных, многократно повторяющихся (во времени или пространстве) подсистем, слабо связанных между собой. Такая структура типична для многих практических задач. Классическим примером являются различные транспортно-производственные задачи, в которых предприятия-поставщики связаны ограничениями на общие ресурсы и удовлетворение потребительского спроса. Особо следует отметить, что известная транспортная задача линейного программирования *) может быть представлена в виде блочной задачи со связывающими ограничениями, как это показано ниже:

$$\begin{array}{c}
 x_{11} \quad \dots \quad x_{12} \quad \dots \quad x_{13} \quad \dots \quad x_{14} \\
 \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{array} \right] = \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \\ b_4 \end{array} \right] \\
 \left[\begin{array}{ccc} 1 & 1 & 1 \end{array} \right] = a_1 \\
 \left[\begin{array}{ccc} 1 & 1 & 1 \end{array} \right] = a_2 \\
 \left[\begin{array}{ccc} 1 & 1 & 1 \end{array} \right] = a_3 \\
 \left[\begin{array}{ccc} 1 & 1 & 1 \end{array} \right] = a_4
 \end{array} \quad (3)$$

В этой задаче ограничения на объемы производства выполняют роль локальных ограничений, а показатели спроса выступают в качестве связывающих условий (такое представление матрицы условий транспортной задачи осуществляется путем перестановки столбцов этой матрицы). Методы решения задач

*) В западной специальной литературе эта задача часто называется *задачей Хичкока — Купманса*. (Прим. перев.)

с подобной структурой технологической матрицы будут специально рассмотрены в главе II.

Запишем *) теперь задачу, двойственную к задаче (2):

$$\begin{array}{rcl} b'_1\pi_1 + b'_2\pi_2 + \dots + b'_r\pi_r + b'_0\pi & = & m\alpha, \\ B'_1\pi_1 & + & A'_1\pi \leq c'_1, \\ B'_2\pi_2 & + & A'_2\pi \leq c'_2, \\ & \vdots & \vdots \\ B'_r\pi_r + A'_r\pi & \leq & c'_r. \end{array} \quad (4)$$

Задача (4) является задачей с блочно-треугольной матрицей условий и связывающими переменными π , заменяющими общие ограничения в задаче (2). Такие задачи называются *блочными задачами со связывающими переменными*, и они являются слабо связанными, если число переменных типа π значительно меньше числа остальных переменных. Легко показать, что методы решения подобных задач могут быть использованы для решения прямых блочно-диагональных задач и наоборот.

Блочные задачи со связывающими переменными возникают тогда, когда основную часть технологических способов можно представить в виде непересекающихся подмножеств способов. В этом случае показатели интенсивности использования остальных технологических способов будут соответствовать связывающим переменным. Хорошим рассмотрением такой структуры является рассмотренная уже нами (см. § 1.2) задача управления производством и запасами, матрица условий которой, как было показано, имеет «ступенчатую» структуру. В этой задаче в качестве связывающих переменных выступают только интенсивности использования технологических способов хранения продукции. Поэтому матрица условий задачи в результате перестановки столбцов, соответствующих таким способам, легко может быть приведена к виду матрицы блочной задачи со связывающими переменными, как это показано в таблице 1.5.1. Такой прием можно использовать и по поводу блочно-треугольных систем, примером которых является динамическая межотраслевая модель В. В. Леонтьева (см. § 1.4) и другие динамические модели планирования производства.

Большая размерность задачи линейного программирования может быть также обусловлена учетом неопределенности исходной информации. В качестве иллюстрации рассмотрим следующую двухэтапную модель линейного стохастического программирования:

$$Ax + By = b, \quad (5)$$

$$x \geq 0, \quad y \geq 0, \quad (6)$$

*) Здесь и далее штрих означает транспонирование (Прим. ред.)

Таблица 1.5.1

Представление задачи управления производством и запасами в двойственной блочно-диагональной форме

[illegible]

где b — случайный вектор. Возможен следующий подход к решению задачи (5)—(6): сначала выбирается некоторый отрицательный вектор x , затем «Природа» выбирает реализацию случайного вектора в соответствии с заданным законом распределения, и, наконец, принимающий решение находит вектор $y \geq 0$, удовлетворяющий условию (5). Вектор y называется *корректирующим вектором* и используется принимающим решение для обеспечения совместности системы

$$Ax \geq b, \quad x \geq 0, \quad (7)$$

которая может быть нарушена из-за случайного выбора вектора. Предполагается, что x выбирается так, чтобы сохранилась возможность выбора допустимого y .

Иначе говоря, множество

$$F = \{x \mid \text{такие, что соотношения } By = b - Ax, y \geq 0 \text{ выполняются для всех возможных значений вектора } b\} \quad (8)$$

непусто, и выбор x должен осуществляться из этого множества. Пусть теперь $c'x$ и $d'y$ — затраты, соответствующие выбранным значениям векторов x и y . После того как зафиксировано значение x и осуществлена реализация вектора b , выбор вектор y заключается в решении детерминированной линейной задачи

$$\min \{d'y \mid By = b - Ax, y \geq 0\}.$$

Этот минимум, зависящий от b и x , обозначим через $\varphi(x, b)$. Так как на первом этапе b — случайный вектор, то $\varphi(x, b)$ — случайная величина при всех значениях x . Выбор x осуществляется в результате решения следующей задачи:

$$\min_{x \in F} \{c'x + E_b \min \{d'y \mid By = b - Ax, y \geq 0\}\}. \quad (9)$$

Функция $E_b \varphi(x, b)$ (где E_b — знак математического ожидания) в общем случае является нелинейной функцией. Двухэтапная задача (9) была рассмотрена Д. Данцигом, который доказал, что $\varphi(x, b)$ и $E_b \varphi(x, b)$ — выпуклые функции.

Покажем способ сведения двухэтапной задачи к задаче линейного программирования. Пусть выбор b осуществлен из конечного множества $\{b_1, \dots, b_n\}$ с вероятностями $p_1, \dots, p_n$. Если y_i является переменной второго этапа при $b = b_i$, то $x \in F$ тогда и только тогда, когда следующая система имеет допустимое решение:

$$\left. \begin{array}{rcl} Ax + By_1 & & = b_1, \\ Ax + & By_2 & = b_2, \\ \vdots & & \vdots \\ & \ddots & \\ Ax & + By_n & = b_n, \\ x \geq 0, y_i \geq 0, & i = 1, \dots, n. \end{array} \right\} \quad (10)$$

Переменные первого и второго этапов выбираются одновременно в результате нахождения минимума функции

$$z = c'x + p_1 d' y_1 + \dots + p_n d' y_n \quad (11)$$

при ограничениях (10). Это — двойственная блочно-диагональная задача, которая может иметь большую размерность, если матрица A содержит хотя бы несколько строк, а n — достаточно большое число. Данциг [7, 8] предложил заменить эту задачу двойственной к ней и использовать для ее решения метод декомпозиции. Метод расчленения Розена также может быть использован для ее решения (см. § 4.4). Однако при использовании любого алгоритма для повышения его эффективности необходимо учитывать повторяемость матриц A и B в условиях (10).

В качестве следующего примера рассмотрим задачу управления запасами. Фиксированное количество продукции q хранится на складе, откуда s единиц перевозится в пункт потребления для удовлетворения спроса d . При этом затраты на доставку продукции равны $c_s s$. Если $d > s$, то взимается штраф в размере $p(d-s)$, под которым, например, можно понимать затраты на покупку продукции на рынке по цене p долларов за единицу.

Введем новые обозначения и запишем условия задачи. Пусть

$$n = d - s, \quad d > s. \quad (12)$$

Если $s > d$, то продукция в объеме

$$e = s - d, \quad s > d \quad (13)$$

должна храниться у потребителя, а затраты на хранение составят $c_{st}e$. При заданном спросе d возникает следующая задача линейного программирования:

$$\left. \begin{aligned} s + y &= q, \\ s + u - l &= d, \\ z = c_s s + pu + c_{st} e &\rightarrow \min. \end{aligned} \right\} \quad (14)$$

Очевидно, все переменные должны быть неотрицательными и для выполнения условий (12) — (13) должно соблюдаться условие $ue = 0$. (Нетрудно понять, что это условие выполняется автоматически для любого базисного решения.)

Если d — случайная величина, то мы приходим к рассмотренной выше задаче, в которой все компоненты вектора b являются случайными величинами. Так как спрос есть целочисленная величина, и возникновение очень большого спроса мало вероятно, можно сделать предположение, что d принимает значение множества $\{d_1, \dots, d_n\}$ с вероятностями $p_1, \dots, p_n$. Тогда соответствующая такой ситуации двухэтапная задача эквивалентна следующей блочной задаче со связывающими

переменным:

$$\min \{c_s s + p_s(pu_1 + c_{st}e_1) + \dots + p_n(pu_n + c_{st}e_n)\}$$

при ограничениях

$$\begin{array}{rcl} s + y & & = q, \\ s & + u_1 - e_1 & = d_1, \\ s & & + u_2 - e_2 = d_2, \\ \dots & \dots & \dots \\ s & & + u_n - e_n = d_n, \\ s \geq 0, \quad u_i \geq 0, \quad e_i \geq 0 & (i = 1, \dots, n). \end{array}$$

Методы решения подобных задач будут нами рассмотрены в главе IV. Представляют интерес задачи со смешанной структурой вида:

$$\begin{array}{rcl} \min \{z = c_1 x_1 + c_2 x_2 + \dots + c_p x_p + cy\}, \\ A_1 x_1 + A_2 x_2 + \dots + A_p x_p + D_0 y = b_0, \\ B_1 x_1 & & + D_1 y = b_1, \\ & B_2 x_2 & + D_2 y = b_2, \\ & \vdots & \vdots \\ & B_p x_p + D_p y = b_p, \\ x_i \geq 0, \quad y \geq 0. \end{array}$$

Такие задачи также будут рассмотрены в главе IV.

§ 1.6. ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С БОЛЬШИМ ЧИСЛОМ СТРОК ИЛИ СТОЛБЦОВ

На практике часто встречаются задачи, технологические матрицы которых имеют большое число строк или столбцов (или одновременно столбцов и строк). Во многих случаях имеет место только одна из этих трудностей. Более того, принцип разложения Данцига — Вульфа (см. главу II) позволяет уменьшить число строк за счет значительного увеличения столбцов. В большинстве случаев почти всегда можно воспользоваться структурными особенностями строк и столбцов. Так, при использовании метода Данцига — Вульфа j -й столбец координирующей задачи представляется в виде Ax^j , где x^j — j -я крайняя точка выпуклого ограниченного многогранника. (Другие примеры будут детально проанализированы в главе III.)

Естественно попытаться решать такие задачи, не просматривая всех столбцов или строк. Этот прием можно продемонстрировать на примере использования симплексного или двойствен-

ного симплексного метода. Запишем для этого в общем виде прямую и двойственную задачи линейного программирования:

Прямая	Двойственная
$\min c'x$	$\max b'\pi$
при ограничениях	при ограничениях
$Ax = b$	$A\pi \leq c$
$x \geq 0$	

При использовании как прямого, так и двойственного симплекс-методов необходимо выполнение трех следующих операций:

1. Выбрать столбец s матрицы A
2. Выбрать строку r матрицы A
3. Провести ведущую операцию, используя a_{rs} в качестве ведущего элемента, и вернуться к операциям 1 и 2.

Для осуществления ведущей операции не обязательно использовать всю матрицу A , размерность которой $(m \times n)$, причем $n > m$.

При использовании, например, модифицированного симплекс-метода необходимо работать только с неособенной квадратной $(m \times m)$ -матрицей B и ведущие операции служат для обновления матрицы B^{-1} .

Пусть $\pi = c_B B^{-1}$ — двойственные переменные, соответствующие матрице B ; $\bar{b} = B^{-1} b$ — вектор значений базисных переменных прямой задачи; p^j — j -й столбец матрицы A ; $\bar{c}_j = c_j - \pi p^j$ — характеристическая разность для x_j ; $\bar{a}_{ij}$ — элемент (i, j) матрицы $B^{-1}A$.

Теперь операции 1 и 2 прямого и двойственного симплекс-методов можно охарактеризовать с помощью таблицы 1.6.1.

Кратко поясним таблицу 1.6.1. При использовании прямого симплекс-метода операции выполняются в последовательности 1, 2, 3. Проведение минимизации в операции 3 необходимо для обеспечения допустимости плана прямой задачи. Минимизация же в операции 2 не обязательна, а лишь желательна, так как введение в базис любого вектора p^j , для которого $\bar{c}_j < 0$, приводит (за исключением случаев вырожденности) к убыванию значения функционала. При использовании двойственного симплекс-метода операции выполняются в последовательности 1, 3, 2 и минимизация в операции 2 таблицы 1.6.1 является обязательным условием получения допустимого решения двойственной задачи. Минимизация же в операции 3 является только желательной. В стандартных приложениях как прямого, так и двойственного симплекс-метода указанные операции минимизации реализуются путем вычисления всех входящих в них чисел и их сравнения. Для задач с большим числом переменных или ограничений такой путь является нерациональным.

Таблица 1.6.1

		Прямой симплекс-метод	Двойственный симплекс-метод
1	Критерий оптимальности	$\bar{c}_j \geq 0$ для всех j	$\bar{b}_i \geq 0$ для всех i
2	Правило определения вектора, подлежащего включению в базис	Если $\bar{c}_s = \min \bar{c}_j < 0$, p^s вводится в базис	Если $-\frac{\bar{c}_s}{a_{rs}} = \min_{a_{ri} < 0} \frac{\bar{c}_i}{-a_{ri}}$, то вектор p^s вводится в базис. Если все $\bar{a}_{ri} \geq 0$, то ограничения несовместимы
3	Правило определения вектора, подлежащего исключению из базиса	Если $\bar{b}_r / \bar{a}_{rs} = \min_{a_{is} > 0} \bar{b}_i / \bar{a}_{is}$, то p_r выводится из базиса. Если все $\bar{a}_{is} < 0$, то значение целевой функции задачи не ограничено снизу	Если $\bar{b}_r = \min \bar{b}_i < 0$, то вектор p^i , стоящий на месте r -го столбца в базисе, выводится из него
4	Ведущий элемент	$\bar{a}_{rs}$	$\bar{a}_{rs}$

Более эффективно, как показал Гомори [9], выполнять операции минимизации, используя алгоритмические процедуры типа линейного или динамического программирования. При этом* используется задача специального вида, решение которой позволяет получить те строки и столбцы матрицы, которые необходимы для применения прямого или двойственного симплекс-методов. Так как формирование строки таблицы двойственного симплекс-метода эквивалентно формированию столбца таблицы прямого симплекс-метода, будет в дальнейшем подобную процедуру называть *процедурой получения столбца*. В зависимости от вида выполняемой минимизации, решается задача линейного или дробно-линейного программирования. Последняя задача может быть эффективно решена с помощью одной из модификаций симплекс-метода (см. 2.11.1), при условии что множество ограничивающих условий является выпуклым многогранником. Если матрица условий задачи содержит большое число столбцов, то ее целесообразно решать непосредственно прямым или двойственным симплекс-методом с использованием отмеченных выше процедур выполнения операции минимизации. От задачи

с технологической матрицей, содержащей большое число строк, удобно предварительно перейти к двойственной, в матрице которой опять-таки окажется много столбцов. Перейдем теперь к рассмотрению некоторых примеров, которые не только иллюстрируют использование общего метода, но и имеют самостоятельный интерес.

Использование метода генерации столбцов для решения линейной задачи оптимального управления. Рассмотрим линейную непрерывную по времени динамическую систему, состояние которой в момент времени t определяется вектором $x(t) = (x_0(t), \dots, x_n(t))$. Система описывается с помощью следующих дифференциальных уравнений:

$$\frac{dx}{dt} = Ax + Bu, \quad 0 \leq t \leq T, \quad (1)$$

$$x(0) = (0, x_1^0, \dots, x_n^0) \text{ (вектор } x(0) \text{ — задан)}, \quad (2)$$

где A и B — постоянные матрицы размерности $((n+1) \times (n+1))$ и $((n+1) \times m)$ соответственно. Компоненты вектора $u = (u_1, \dots, u_m)$ суть управляющие воздействия (управления), определенные на $(0, T)$. Выбор u ограничен условием, что значения $u(t)$ при любом $t \in [0, T]$ должны принадлежать компактному выпуклому множеству $S(t)$:

$$u(t) \in S(t), \quad 0 \leq t \leq T. \quad (3)$$

Предположим для простоты, что при $t=T$ заданы значения последних n компонент вектора состояния, т. е.

$$x_i(T) = x_{ij}^T \quad (i = 1, \dots, n). \quad (4)$$

Задача оптимального программного управления системой теперь состоит в том, чтобы выбрать u , минимизирующее $x_0(T)$. Если, например,

$$\frac{dx_0}{dt} = 1, \quad (5)$$

$$\text{тогда} \quad x_0(0) = 0, \quad (6)$$

$$x_0(T) = T, \quad (7)$$

и задача заключается в переходе из исходного состояния в конечное за минимальное время T (задача об оптимальном быстродействии). Данциг показал [10], что подобные задачи могут быть решены методами линейного программирования. Общее решение системы (1) можно представить в виде

$$x(T) = e^{AT}x(0) + \int_0^T e^{A(T-t)}Bu(t)dt. \quad (8)$$

Пусть

$$p = \int_0^T e^{A(T-t)} Bu(t) dt. \quad (9)$$

Если обозначить

$$R = \left\{ p \mid p = \int_0^T e^{A(T-t)} Bu(t) dt, u \in S \right\}, \quad (10)$$

то в силу выпуклости S и линейности операции (9) нетрудно показать, что и R — выпуклое множество. Предположим, что известно множество векторов управления $u^i \in S$, $i \in 1, \dots, q$, $q \geq n+1$, определяющих множество векторов $\{p^i\}$. Из свойства выпуклости следует, что выпуклая оболочка $\{u^i\}$ принадлежит множеству S , а выпуклая оболочка $\{p^i\}$ — множеству R . Предположим далее, что элемент u^j выпуклой оболочки $\{u^i\}$ генерирует вектор p^j , удовлетворяющий условию (8), т. е. известно начальное допустимое управление. Теперь нетрудно сформулировать задачу линейного программирования для определения вектора управления, минимизирующего $x_0(T)$.

Найти $\min x_0(T)$ при условиях

$$e^{AT} x(0) + \sum_i \lambda_i p^i = x(T), \quad (11)$$

$$\sum_i \lambda_i = 1, \quad \lambda_i \geq 0. \quad (12)$$

Пусть на k -й итерации с помощью модифицированного симплекс-метода решена эта задача и при этом получены оптимальные базис B^k , решение $x_0^k(T)$, $\{\lambda_i^k\}$ и вектор двойственных переменных $\{\pi^k, \pi_0^k\}$, где π^k — $(n+1)$ -мерный вектор, π_0^k — скаляр. Тогда вектор управления можно записать в виде

$$u^k = \sum_i \lambda_i^k u^i. \quad (13)$$

Этот вектор является аппроксимацией оптимального управления. Пусть необходимо улучшить эту аппроксимацию в результате присоединения некоторой точки в начальной «сетке» точек $\{u^i\}$. Для того чтобы выбрать наилучшую точку, предположим, что уже осуществлен выбор большого числа точек и для каждой из них найден вектор p^i . Ограничения (11) соответствует много столбцов p^i с переменными λ_i . Для каждого столбца p^i определим характеристическую разность

$$\bar{c}_i = -\pi^k p^i - \pi_0^k, \quad (14)$$

которая должна быть минимизирована. Поскольку p^i генерируется произвольным $u^i \in S$, то поиск минимума среди всех $\bar{c}_i$, эквивалентен максимизации функционала вида

$$\int_0^T \pi^k e^{A(T-t)} B u(t) dt \quad (15)$$

при условиях

$$u \in S. \quad (16)$$

Решение этой задачи *) может быть осуществлено в результате нахождения максимума подынтегрального выражения для каждого t :

$$\left. \begin{array}{l} \max \lambda^k(t) B u(t) \\ \text{при ограничениях} \\ u(t) \in S(t) \end{array} \right\} \begin{array}{l} \text{подзадача (одна} \\ \text{для каждого } t), \end{array} \quad (17)$$

где

$$\lambda^k(t) = \pi^k e^{A(T-t)}. \quad (19)$$

Эти подзадачи заменяют шаг 1 двойственного симплекс-метода. Если $S(t)$ — многогранник, тогда для каждого t необходимо решать задачу линейного программирования. Если же $S(t)$ не зависит от t , тогда в подзадачах будут изменяться лишь целевые функции. Если $S(t)$ определяется заданием верхней и нижней границ по каждой из компонент вектора u , то решение подзадачи характеризуется тем, что u_i «переключаются», т. е. принимают значения на уровне нижней или верхней границ в зависимости от изменения знака i -й компоненты $\lambda^k(t)B$. Пусть $\hat{u}^{k+1}$ — решение подзадач. Условию $\min \bar{c}_i < 0$ здесь соответствует выражение

$$-\int_0^T \pi^k e^{A(T-t)} B \hat{u}^{k+1}(t) dt - \pi^k < 0. \quad (20)$$

Если это условие выполнено, то (за исключением случаев вырожденности) введение в базис нового столбца $\hat{p}^{k+1}$ приведет к уменьшению значения $x_0(T)$. В противном случае полученное решение является оптимальным (см. § 3.4.1). Данциг доказал [11] сходимость такого итеративного процесса при $k \rightarrow \infty$. Теперь — о числе точек. Вообще говоря, имеется несчетное множество точек $u^j(p^j)$, однако использование метода генерации столб-

*) Тот же результат сразу получается из принципа максимума Понтрягина, хотя из него и не следует сама итеративная процедура. Максимизируемое в (17) выражение обычно именуют *гамильтонианом* задачи. Поиск π^k эквивалентен поиску неизвестных граничных условий сопряженной задачи (12). (Прим. ред.)

пов позволяет найти вполне реализуемый (хотя и не конечный) алгоритм.

Если переменные $\lambda_{j_1}, \dots, \lambda_{j_{n+1}}$ являются оптимальными базисными и входят в оптимальный план со значениями $\lambda_{j_i}^*$, $i=1, \dots, n+1$, тогда оптимальное решение можно представить в виде следующей выпуклой комбинации:

$$u^* = \sum_{i=1}^{n+1} \lambda_{j_i}^* u_i^l. \quad (21)$$

Так как столбцы, «порождаемые» u_i^l , входят в состав оптимального базиса, то соответствующие им характеристические разности равны нулю:

$$\int_0^T \pi^* e^{A(T-t)} B u_i^l dt = -\pi_0^* \quad (i=1, \dots, n+1). \quad (22)$$

Если

$$\lambda^* = \pi^* e^{A(T-t)}, \quad (23)$$

то из (21) и (22) следует

$$\int_0^T \lambda^* B u^* dt = -\pi_0^*. \quad (24)$$

Здесь (π^*, π_0^*) — оценка оптимального плана. Вычисленные с помощью этих оценок характеристики разности для всех возможных генерируемых столбцов должны быть неотрицательными. В терминах подзадачи (15)–(16) это эквивалентно выражению

$$\max_{u \in S} \int_0^T \lambda^* B u dt \leq -\pi_0^*. \quad (25)$$

Теперь, принимая во внимание (24) и (25), можно записать:

$$\max_{u \in S} \int_0^T \lambda^* B u dt = \int_0^T \lambda^* B u^* dt, \quad (26)$$

или

$$\lambda^*(t) B u^*(t) = \max_{u(t) \in S(t)} \lambda^*(t) B u(t) \quad (27)$$

для всех точек отрезка $[0, T]$. Таким образом, нами получено выражение для принципа максимума Понтрягина [12] применительно к условиям рассматриваемой задачи. Интересно, что мы пришли к нему, используя аппарат линейного программирования. Все эти вопросы более детально будут проанализированы в §§ 3.3 и 3.4.

Задача со многими строками. Рассмотрим задачу линейного программирования, ограничения которой разбиты на два подмножества. Найти

$$\min cx \quad (28)$$

при ограничениях

$$A_1 x = b_1 \quad (m_1 \text{ строк}), \quad (29)$$

$$A_2 x = b_2, x \geq 0 \quad (m_2 \text{ строк}). \quad (30)$$

Предположим, что многогранник

$$S = \{x \mid A_2 x = b_2, x \geq 0\} \quad (31)$$

ограничен, а x^j — j -я крайняя точка S . Представим произвольную точку x этого многогранника в виде

$$x = \sum_j \lambda_j x^j, \quad \sum_j \lambda_j = 1, \lambda_j \geq 0 \quad (32)$$

(см. теорему 2.2.2) и подставим (32) в (28) и (29). В результате получим следующую «координирующую задачу»:

Найти

$$\min \sum_j (cx^j) \lambda_j \quad (33)$$

при условиях

$$\sum_j (A_1 x^j) \lambda_j = b_1, \quad (34)$$

$$\sum_j \lambda_j = 1, \quad (35)$$

$$\lambda_j \geq 0. \quad (36)$$

Задача (33) — (36) эквивалентна исходной задаче, так как если $\{\lambda_j^0\}$ — решение задачи (33) — (36), то

$$x^0 = \sum_j \lambda_j^0 x^j \quad (37)$$

есть решение задачи (28) — (30). Координирующая задача имеет много столбцов, но зато только $m_1 + 1$ строку по сравнению с $m_1 + m_2$ строками исходной задачи.

Задачу двойственную к координирующей можно записать так:

Найти

$$\max \pi b_1 + \pi_0 \quad (38)$$

при условиях

$$\pi A_1 x^j + \pi_0 \leq (cx^j) \text{ для всех } j. \quad (39)$$

Эта задача имеет только $m_1 + 1$ переменных и включает большое число неравенств.

Абади и Вильямс [13] указали метод решения пары задач (33)—(36) и (38)—(39), предложив применить двойственный симплекс-метод к задаче (33)—(36). Предположим, что уже известны начальный базис и соответствующий ему план двойственной задачи. Такая ситуация может возникнуть, когда в уже решенной задаче приходится изменить значение B_1 . Пусть, далее, используется двойственный симплекс-метод. Единственная трудность возникает при реализации операции 2 (см. стр. 35), связанной с выбором ведущего столбца. Предположим, что ведущий элемент находится в строчке r и пусть B — текущий (недопустимый) базис координирующей задачи. Пусть, далее, b_r — r -я строка матрицы B^{-1} . В соответствии с процедурой двойственного метода необходимо найти

$$\min_{\bar{a}_{rj} < 0} \frac{-\bar{c}_j}{\bar{a}_{rj}}. \quad (40)$$

Пусть вектор b_r представлен в виде

$$b_r = (\hat{b}_r, b_r^0),$$

где $\hat{b}_r$ — m_1 -мерный вектор, а b_r^0 — скаляр. Тогда для координирующей задачи можно записать:

$$\hat{a}_{rj} = (\hat{b}_r A_1) x^j + b_r^0, \quad (41)$$

$$-\bar{c}_j = (\pi A_1 - c) x^j + \pi_0. \quad (42)$$

Поэтому мы должны найти минимум отношения

$$\frac{(\pi A_1 - c) x^j + \pi_0}{(\hat{b}_r A_1) x^j + b_r^0} \quad (43)$$

на множестве всех крайних точек x^j многогранника S , для которых знаменатель отрицателен. Ниже будет показано (см. п. 2.11.1), что если отношение линейных функций (дробно-линейный функционал) достигает конечного минимума на выпуклом многограннике, то имеется крайняя точка, которая является оптимальной. Нахождение такой точки связано с решением подзадачи вида

$$\min \frac{(\pi A_1 - c) x + \pi_0}{(\hat{b}_r A_1) x + b_r^0} \quad (44)$$

при ограничениях

$$A_r x = b_r, \quad (45)$$

$$x \geq 0, \quad (46)$$

$$(\hat{b}_r A_1) x + b_r^0 < 0. \quad (47)$$

Решение этой задачи заменяет выполнение операции 2 двойственного симплексного метода, а остальные операции остаются без изменения.

Если x^{j*} — решение задачи (44) — (47), то в базис вводится вектор

$$\begin{bmatrix} A_1 x^{j*} \\ 1 \end{bmatrix} \quad (48)$$

с коэффициентом в целевой функции, равным $s x^{j*}$.

Конечный алгоритм для решения задачи (44) — (48) будет рассмотрен в п. 2.11.1. Другие задачи с большим числом строк исследуются в §§ 3.5 и 6.3.

§ 1.7. НЕЛИНЕЙНЫЕ ЗАДАЧИ СО СВЯЗУЮЩИМИ ПЕРЕМЕННЫМИ

Важный класс задач нелинейного программирования может быть представлен в виде

$$\left. \begin{array}{l} \min \{c_1(y)x_1 + c_2(y)x_2 + \dots + c_p(y)x_p + c_0(y)\}, \\ \begin{array}{rcl} A_1(y)x_1 & & \geq b_1(y), \\ A_2(y)x_2 & & \geq b_2(y), \\ & \vdots & \\ A_p(y)x_p & \geq & b_p(y), \\ Dy & \geq & \beta, \end{array} \end{array} \right\} \quad (1)$$

где коэффициенты целевой функции c_i , технологические матрицы A_i и векторы b_i являются функциями вектора связывающих переменных y . Если зафиксировать значения этих переменных, то задача (1) становится задачей линейного программирования с переменными x . Эта задача имеет блочно-диагональную структуру технологической матрицы и поэтому распадается на p независимых подзадач. Задачи подобного типа являются нелинейным обобщением блочных линейных задач со связывающими переменными. Большинство методов решения таких задач основано на идеях декомпозиции и сводится к решению последовательности линейных подзадач для различных фиксированных значений вектора y . На этих идеях, например, основан метод Розена [14], который будет рассмотрен в § 6.2.

Задачи типа (1) часто возникают при составлении графиков производства и распределения продукции нефтеперерабатывающей промышленности, когда оптимальный план составлен для целого комплекса предприятий. Примерная схема такого комплекса представлена на рис. 1.3.

Каждый нефтеперерабатывающий завод (НПЗ) представляет сложный комплекс, состоящий из многих технологических установок (агрегатов), осуществляющих операции дистилляции, крекинга и т. п. Основное назначение НПЗ заключается в переработке сырья в конечные продукты (авиационный бензин, дизельное топливо, мазут, масло), которые отправляются на базы

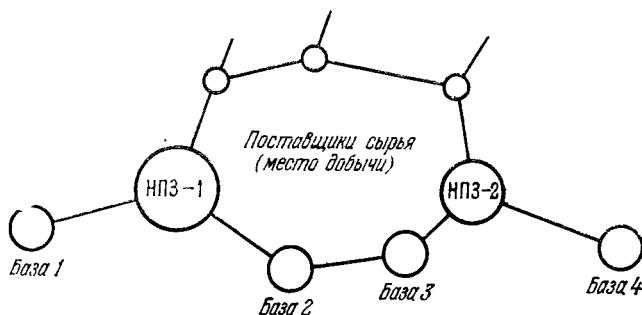


Рис. 1.3. Нефтеперерабатывающий комплекс.

для отправки потребителю. При планировании возникают следующие типичные задачи:

1. Наилучшим образом распределить различные виды сырья (нефти) между НПЗ, а внутри НПЗ между технологическими установками.

2. Рационализировать материальные потоки между различными установками. (Решение задачи усложняется наличием промежуточных продуктов, которые могут быть направлены на другие НПЗ для дальнейшей переработки.)

3. Определить оптимальные варианты, режимы использования технологических установок с учетом различных факторов и условий (температура, давление, коэффициент рецикла, катализирующие добавки). Зависимость свойств потоков от этих переменных, как правило, носит ярко выраженный нелинейный характер.

4. Обосновать наилучшие варианты смешения продуктов при определенных ограничениях на продукты смешения (например, на октановое число бензинов).

5. Найти рациональный вариант закрепления потребителей нефтепродуктов за НПЗ.

Все эти вопросы взаимосвязаны. Так, решение последнего вопроса зависит не только от транспортных затрат, но и от производственных мощностей по различным нефтепродуктам на предприятиях и от относительной эффективности производства этих продуктов. Решение осложняется еще и тем обстоятельством, что НПЗ различаются возрастом и структурой оборудования основного и вспомогательного производств. Будем в дальнейшем плановый горизонт (период) принимать равным кварталу или полугодью, что избавляет нас от необходимости

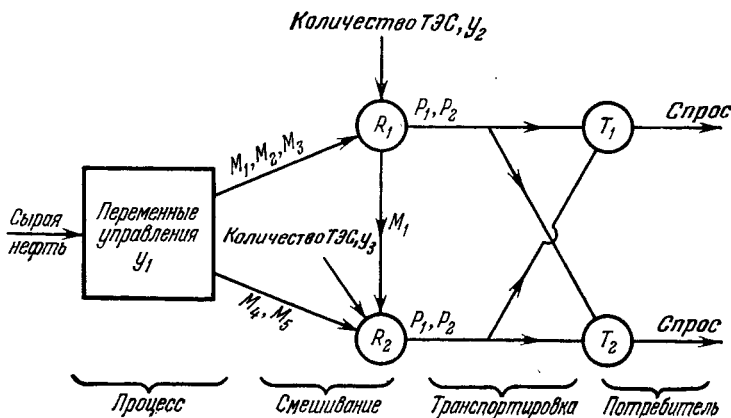


Рис. 1.4. Система производства и распределения продукции.

рассмотрения специальных вопросов оперативно-календарного планирования.

Покажем теперь возможность использования нелинейной модели (1) для решения последней задачи на следующем упрощенном примере [15]. Пусть предприятия R_1 и R_2 производят продукты P_1 и P_2 и отправляют их в пункты потребления T_1 и T_2 . Суммарный спрос (за весь период) на продукцию в этих пунктах предполагается известным и пусть $D(P_i, T_j)$ — величина спроса на продукт P_i в пункте T_j . На предприятии R_1 три вида материалов (сырья) M_1, M_2, M_3 могут быть использованы в смешении для получения продуктов P_1 и P_2 . Кроме того, материал M_1 может быть использован на предприятии R_2 для смешения с материалами M_4 и M_5 . Эту ситуацию схематически можно представить в виде схемы рис. 1.4.

Предположим далее, что в производстве продукции P_1 и P_2 должно соблюдаться ограничение на минимальное октановое число s_1 и s_2 . Последнее зависит как от количества материала, участвующего в смешении, так и от концентрации тетраэтил-

свинца (ТЭС) y_2 . Пусть $f_i(y)$ — октановое число материала M_i как функция от добавки y ($\text{см}^3/\text{баррель}$); x_{ij} — количество материала i , используемого для получения продукта j . Обычно предполагается [16], что вклад продукта M_i в октановое число продукта смешения равен $f_i(y)$ ($x_{ij} / \sum_i x_{ij}$). Если применяется несколько

материалов, октановое число конечного продукта принимается равным сумме соответствующих величин. Таким образом, если s_j — минимальное допустимое октановое число для продукта j , ограничение можно записать так:

$$\sum_{i=1}^3 f_i(y) \left(x_{ij} / \sum_{i=1}^3 x_{ij} \right) \geq s_j, \quad (2)$$

или, иначе,

$$\sum_{i=1}^3 (f_i(y) - s_j) x_{ij} \geq 0. \quad (3)$$

На переменные смешения x_{ij} также наложены ограничения: расход материалов должен находиться в пределах их наличия. Доступное же количество зависит от режима работы агрегатов, питающих операцию смешения, т. е. от управляющих переменных y_1 . Если $b_i(y_1)$ — располагаемое количество i -го материала, то ограничения на использование материалов можно записать в следующем виде:

$$x_{11} + x_{12} + (V_1, \text{ доставленное в } R_2) \leq b_1(y_1), \quad (4)$$

$$x_{21} + x_{22} \leq b_2(y_1), \quad (5)$$

$$x_{31} + x_{32} \leq b_3(y_1). \quad (6)$$

Ограничения (3) — (6) удобно представить в матричной форме:

P_2			P_2			
M_1	M_2	M_3	M_1	M_2	M_3	
1			1			$+(M_1 \text{ в } R_2) \leq b_1(y_1)$ $\leq b_2(y_1)$ $\leq b_3(y_1)$ ≥ 0 ≥ 0
	1			1		
		1			1	
$a_{11}(y_2)$	$a_{21}(y_2)$	$a_{31}(y_2)$	$a_{12}(y_2)$	$a_{22}(y_2)$	$a_{32}(y_2)$	

(7)

где

$$a_{ij}(y) = f_i(y) - s_j. \quad (8)$$

Аналогично записываются ограничения и для предприятия 2:

P_1			P_2		
M_1	M_2	M_3	M_1	M_2	M_3
1			1		
	1			1	
		1			1
$a_{11}(y_3)$	$a_{21}(y_3)$	$a_{31}(y_3)$	$a_{12}(y_3)$	$a_{22}(y_3)$	$a_{32}(y_3)$

$$\begin{aligned}
 &-(M_1 \text{ в } R_2)=0 \\
 &\leq b_4(y_1) \\
 &\leq b_5(y_1) \\
 &\geq 0 \\
 &\geq 0
 \end{aligned}$$

(9)

где y_3 — концентрация ТЭС, используемая на R_3 . Нетрудно понять, что переменная, соответствующая поставкам материала M_1 с R_1 в R_2 , играет роль связывающей переменной для блоков ограничений этих предприятий. Если имеется несколько таких переменных, то их целесообразно объединить в отдельный блок, так как, например, при использовании метода разбиения Розена время счета прежде всего зависит от числа связывающих переменных [17] более подробно этот вопрос рассматривается в работе [15].

Составной частью общей модели являются ограничения транспортного типа (ограничения на выпуск продукции и удовлетворение спроса). Эти ограничения в матричной форме можно представить следующим образом:

T_1				T_2			
R_1		R_2		R_1		R_2	
P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2
1				1			
	1				1		
		1				1	
			1				1
1		1					
	1		1	1		1	
					1		1

$$\begin{aligned}
 &-(P_1 \text{ из } R_1)=0, \\
 &-(P_2 \text{ из } R_1)=0, \\
 &-(P_1 \text{ из } R_2)=0, \\
 &-(P_2 \text{ из } R_2)=0, \\
 &=D(P_1, T_1) \\
 &=D(P_2, T_1) \\
 &=D(P_1, T_2) \\
 &=D(P_2, T_2)
 \end{aligned}$$

$$\left. \begin{array}{l} \text{ограничения} \\ \text{на вывоз} \\ \text{спрос удовлетворяется} \\ \text{полностью} \end{array} \right\}$$

(10)

Необходимо также учесть балансовые ограничения на использование сырья и материалов: количество конечной продукции P_i , отправленной из R_j , должно быть равным количеству материа-

лов, участвовавших в процессе смешения этого продукта. Поэтому, например,

$$(P_1 \text{ из } R_1) = x_{11} + x_{21} + x_{31}. \quad (11)$$

Ограничения типа (11) образуют другую группу связывающих ограничений. Если теперь объединить все перечисленные ограничения, то в результате мы получим общую матрицу условий рассматриваемой задачи, представленную в таблице 1.8.1.

Эта матрица имеет структуру, аналогичную структуре матрице задачи (1). Она включает три диагональных блока, соответствующих различным этапам технологического процесса и пять связывающих переменных. В качестве критерия оптимальности используется величина прибыли. Если спрос зафиксирован, тогда задача решается на минимум суммарных производственных и транспортных издержек.

Обозначим через c_{ij} транспортные затраты на перевозку единицы продукции из i в j . Тогда суммарные транспортные затраты составят

$$c_s = \sum_{i,j} c_{ij} [(P_1 R_i T_j) + (P_2 R_i T_j)], \quad (12)$$

где $(P_i R_j T_k)$ — переменная из соответствующей ей клетки транспортной матрицы (10).

Поясним принцип расчета затрат на смешение продуктов. Если c — стоимость единицы ТЭС, то общая стоимость израсходованного ТЭС будет равна

$$c_s = \sum_{i,j} [c \text{ (долл/см}^3\text{)}, y \text{ (см}^3\text{/баррель)}] x_{ij} \text{ (баррель)}. \quad (13)$$

Таким образом, затраты на смешение зависят от y .

Теперь можно определить функцию $\Phi_i(y_i)$ как затраты на получение материала в объеме $b_i(y_i)$, когда переменная управления принимает значение, равное y_i . Поэтому общие производственные издержки будут равны

$$c_s = \sum_{i=1}^5 \Phi_i(y_i), \quad (14)$$

а суммарные затраты — $c_s + c_p$.

Конечно, построение модели для практических целей потребует более детального описания всех технологических процессов, и прежде всего процессов, предшествующих смешению. Такая модель описана в работе [15]. Она включает 60 связывающих переменных, шесть блоков и 500—600 ограничений в форме неравенств.

Таблица 1.81

P_1			P_2		
M_1	M_2	M_3	M_1	M_2	M_3
1			1		
	1			1	
		1			1
1	1	1			
			1	1	1
$a_{11}(y_2) \ a_{21}(y_2) \ a_{31}(y_2)$					
			$a_{12}(y_2) \ a_{22}(y_2) \ a_{32}(y_2)$		

бензиновая смесь
в установке R_1

P_1			P_2		
M_1	M_4	M_5	M_1	M_4	M_5
1			1		
	1			1	
		1			1
1	1	1			
			1	1	1
$a_{11}(y_3) \ a_{41}(y_3) \ a_{51}(y_3)$					
			$a_{12}(y_3) \ a_{42}(y_3) \ a_{52}(y_3)$		

бензиновая смесь
в установке R_2 .

Т а б л и ц а 1.8.1 (продолжение).

M_1 θ R_2	P_1 u_3 R_1	P_2 u_3 R_1	P_1 u_3 R_2	P_2 u_3 R_2	
1					$\leq b_1(y_1)$
					$\leq b_2(y_1)$
					$\leq b_3(y_1)$
-1					$= 0$
	-1				$= 0$
					≥ 0
					≥ 0

-1					$= 0$
					$\leq b_4(y_1)$
					$\leq b_5(y_1)$
		-1			$= 0$
			-1		$= 0$
					≥ 0
					≥ 0

T_1				T_2			
R_1	R_2	R_1	R_2	R_1	R_2	R_1	R_2
P_1	P_2	P_1	P_2	P_1	P_2	P_1	P_2
1				1			
	1				1		
		1				1	
			1				1
1	1						
		1	1				
				1	1		
						1	1

-1				$= 0$
	-1			$= 0$
		-1		$= 0$
			-1	$= 0$
				$= D(P_1, T_1)$
				$= D(P_2, T_1)$
				$= D(P_1, T_2)$
				$= D(P_2, T_2)$

Перевозка бензина
с R_1 и R_2 на базы

§ 1.8. ЧАСТИЧНО-ЦЕЛОЧИСЛЕННЫЕ ЗАДАЧИ И ПРОБЛЕМА РАЗМЕЩЕНИЯ

Выше были рассмотрены задачи, которые преобразуются в блочно-диагональные линейные задачи, если зафиксировать значения некоторых переменных. Рассмотрим теперь новый класс задач, куда входят и некоторые из уже рассмотренных. Задачи этого класса можно привести к следующему виду. Минимизировать

$$\{c'x + f(y)\} \quad (1)$$

при ограничениях

$$Ax + F(y) \geq b, \quad (2)$$

$$x \geq 0, \quad y \in S, \quad (3)$$

где x — n -мерный вектор, A — постоянная матрица размерности $(m \times n)$, c — n -мерный вектор констант. Если зафиксировать значение y , то (1) — (3) переходит в задачу линейного программирования. Величина $f(y)$ — функция (возможно нелинейная) p -мерного вектора y , $F(y)$ — вектор-функция, S — произвольное подмножество E^p .

Многие важные в практическом отношении задачи могут быть приведены к виду (1) — (3). Если S — множество p -векторов с неотрицательными целочисленными компонентами, а F и f — линейны, то (1) — (3) окажется хорошо известной задачей частично-целочисленного линейного программирования. Если же $S = E^p$ или S — выпуклый многогранник в E^p , а f и F — нелинейные функции, мы приходим к классу задач, которые будут рассмотрены в § 7.2 (при этом не будет допускаться зависимость c и A от вектора y). Если A имеет блочно-диагональную структуру, f и F — линейные функции и $S = E^p$ или $(E^p)^+$, то возникает блочная задача, аналогичная задаче из § 2.5. Во всех этих случаях существенным является аддитивность вхождения функции y в ограничения и целевой функционал. Алгоритм разбиения Бендерса [18] (см. § 6.3) специально построен для решения подобных задач. В соответствии с этим алгоритмом задачи (1) — (3) решаются в такой последовательности: вначале фиксируется некоторое значение $y^* \in S$ и решается задача

$$\text{минимизировать } c'x \text{ при ограничениях } Ax \geq b - F(y^*). \quad (4)$$

Затем на основе решения двойственной задачи определяется возможность улучшения решения, полученного на первом шаге и находится новое значение вектора y . Таким образом, полностью используются преимущества частичной линейной задачи, что особенно важно, когда матрица A имеет специальную структуру (например, блочно-диагональную или транспортного

типа). В этом случае задача (4) сравнительно легко решается. Эти преимущества не реализуются при использовании алгоритмов, в которых x и y изменяются одновременно, так как структуры F и A при этом обычно смешиваются, и особенности матрицы A не могут быть использованы.

В качестве примера рассмотрим следующую задачу размещения. Фирма должна разместить предприятия и склады таким образом, чтобы спрос потребителей был удовлетворен полностью, а суммарные затраты на производство, хранение и перевозку продукции были бы минимальными (продукция может

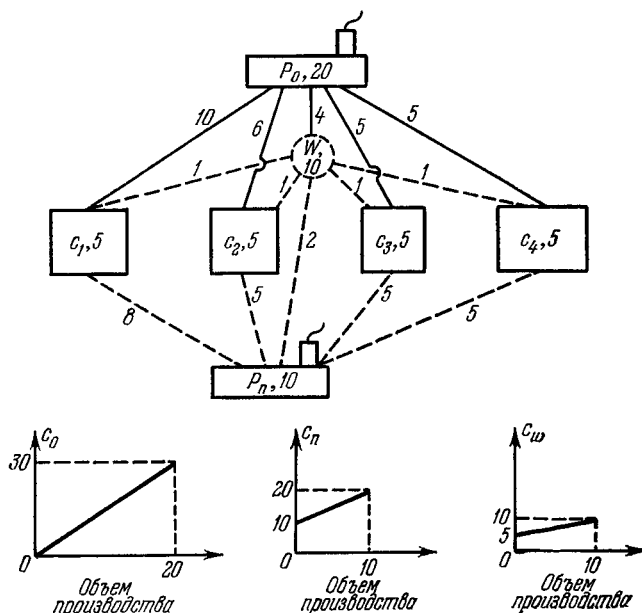


Рис. 1.5. Сетевая форма условий задачи производства и распределения продукции.

отправляться транзитом или направляться потребителю через склад).

При формулировке задачи используются как дискретные, так и непрерывные переменные (дискретные — строить или не строить предприятие и склад, непрерывные соответствуют объемам производства и поставок продукции). Условия одной из таких задач в сетевой форме представлены на рис. 1.5. Поясним этот рисунок. Существующее предприятие P_0 обслуживает потребителей $C_1 \div C_4$. Возможно строительство нового предприятия P_n и склада W . Числа в прямоугольниках и на дугах сети

обозначают мощность предприятий, спрос потребителей и удельные транспортные затраты на перевозку единицы продукции. На этом же рисунке даны графики функций общих издержек на производство продукции для двух предприятий (существующего и намечаемого к строительству) и склада. В состав этих издержек включены также капитальные затраты (для склада и нового предприятия), представленные скачком в нуле.

Так как новое предприятие и склад можно строить (1) и не строить (0), то при решении задачи необходимо оценить четыре различных варианта плана. Для каждого варианта определим минимальные суммарные издержки (производственные и транспортные). Например, если будет осуществляться строительство только одного нового предприятия P_0 на полную мощность (затраты=20), то существующее (более «дорогое») предприятие будет выпускать 10 единиц продукции (затраты=15). При этом новое предприятие будет снабжать потребителей c_1 и c_2 ($f_{n1}=f_{n2}=5$, общие транспортные затраты=65), существующее — $c_3=c_4$, $f_{o3}=f_{o4}=5$, общие транспортные затраты=50). Суммарные затраты для этого варианта равны 150. Понятно, что если таких вариантов немного, то подобная процедура поиска оптимального решения путем прямого перебора, вероятно,

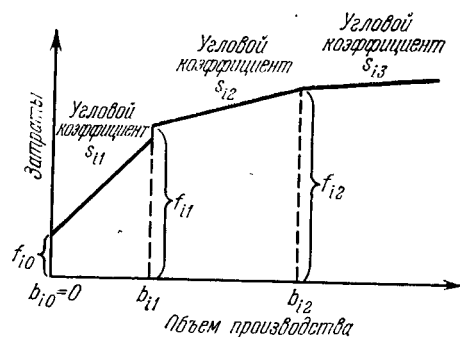


Рис. 1.6. Кусочно-линейная разрывная функция затрат.

является наиболее эффективной. Однако число альтернатив быстро растет как при более реалистичном способе задания функций производственных затрат (как это, например, показано на рис. 1.6), так и в случае большого числа новых предприятий. Функция затрат, представленная на рис. 1.6, учитывает хорошо известный экономический факт: с увеличением мощности предприятия возрастают условно-постоянные затраты, однако их увеличение компенсируется снижением условно-переменных затрат, что отражает преимущества крупных предприятий (т. е. на таких предприятиях затраты на единицу продукции меньше, чем на мелких предприятиях).

Рис. 1.6 соответствует случаю наличия трех вариантов мощности предприятия, каждый из которых соответствует определенной зоне линейности. В описанных проблемах само перечисление возможных вариантов затруднительно. Сведение задачи размещения к задаче частично-целочисленного ли-

нейного программирования вида (1) — (3) позволяет, как правило, сделать процедуру перебора вариантов более эффективной.

Введем обозначения:

r_j — спрос j -го потребителя; c_{ij} — удельные транспортные затраты на перевозку единицы груза из пункта i потребителю j ; x_{ij} — объем перевозок из пункта i в пункт j ; m_{ik} — количество продукции, производимой (накапливаемой) предприятием i (складом i) при работе по k -му варианту (с затратами на прирост единицы продукта, равными s_{ik}); J_p, J_w, J_c — множества номеров пунктов производства, складов и потребителей соответственно.

Переменные m_{ik} ограничены сверху и снизу и должны удовлетворять дополнительным условиям из-за наличия взаимоисключающих вариантов мощности K . Чтобы учесть эти условия, введем переменные y_{ik} , принимающие значения нуль или единица (0 или 1), так что

$$y_{ik} = 0 \Rightarrow m_{ik} = 0, \quad (5)$$

$$y_{ik} = 1 \Rightarrow b_{i,k-1} \leq m_{ik} \leq b_{ik} \quad (6)$$

и не более чем одна переменная $y_{ik} = 1$ для каждого пункта производства i . Соответствующие ограничения можно представить в виде

$$y_{ik} b_{i,k-1} \leq m_{ik} \leq y_{ik} b_{ik}, \quad (7)$$

$$\sum_k y_{ik} = 1, \quad y_{ik} = 0 \text{ или } 1. \quad (8)$$

Так как спрос должен быть удовлетворен полностью, возникают ограничения:

$$\sum_i x_{ij} = d_j, \quad j \in J_c. \quad (9)$$

Переменные x_{ij} и m_{ik} связаны между собой ограничением

$$\sum_k m_{ik} = \sum_{j \in J_c \cup J_w} x_{ij} \quad (10)$$

или вида

$$\sum_{i \in J_p} x_{ij} = \sum_{k \in J_c} x_{jk}, \quad j \in J_w \quad (11)$$

(т. е. для каждого склада входящие и выходящие потоки продукции равны между собой). Емкость склада ограничена, поэтому должно соблюдаться ограничение

$$\sum_{i \in J_p} x_{ij} \leq a_j, \quad j \in J_w. \quad (12)$$

Общие затраты должны быть минимальными, т. е.

$$z = \sum_{i,j} c_{ij}x_{ij} + \sum_{i,k} s_{ik}m_{ik} + \sum_{i,k} y_{ik}(f_{i,k-1} - s_{ik}b_{i,k-1}) = \min. \quad (13)$$

Первый член этого выражения — транспортные издержки, а второй и третий соответствуют затратам на строительство и производство продукции. Задача (7) — (13) может быть приведена к виду (1) — (3), в которой F и f линейные функции, а S — множество возможных значений переменных y_{ik} . Методом Бендерса задача (7) — (13) решается в два приема: вначале фиксируются некоторые значения переменных y_{ik} и решается линейная задача, затем в результате решения целочисленной задачи определяются новые значения y_{ik} . Оптимальное решение задачи, двойственной к линейной, используется для построения дополнительных ограничений в целочисленной задаче, что уменьшает множество допустимых альтернатив, проверяемых на оптимальность. Таким образом, эта процедура может быть рассмотрена как схема, «автоматизирующая» перебор вариантов, в которой информация о ранее рассмотренных случаях используется для выделения случаев, которые следует рассматривать далее.

Задачи

1. Дать выражение для e^{A^t} , когда A имеет различные характеристические числа.
2. Доказать, что если p определяется по формуле (1.6—9), тогда множество всех векторов, порождаемых $u \in S$, выпукло.
3. Указать метод нахождения начального множества векторов $\{p^i\}$, удовлетворяющих условию (1.6—10).
4. Рассмотрим линейную параметрическую задачу с единственным параметром в целевой функции или в правой части условий задачи.

Доказать, что стандартная процедура выбора нового базисного столбца может быть заменена решением задачи нахождения минимума дробно-линейной функции на выпуклом многограннике.

ЛИТЕРАТУРА

1. G. B. Dantzig, Large — Scale System Optimization: A Review, Operations Research Centre, University of California at Berkeley, Rept. ORC65—9, 1965.
2. G. B. Dantzig and R. M. Van Slyke, Generalized Upper Bounding Techniques, J. Computer System Sci., 1, 1967, pp. 213—226.
3. G. B. Dantzig, Compact Basis Triangularization for the Simplex Method, in «Recent Advances in Mathematical Programming», R. L. Graves and P. Wolfe, eds., McGraw-Hill, Inc., New York, 1963, pp. 125—132.
4. M. D. Mesarovic, D. Macko, and Y. Takahara, Theory of Multi-Level Hierarchical Systems, Academic Press, 1970. [Русский перевод: М. Месарович, Д. Мако, И. Такаха́ра, Теория иерархических многоуровневых систем, «Мир», М., 1973.]
5. G. B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, N. J., 1963, chap. 3. [Русский перевод: Данциг Дж.,

- Линейное программирование, его обобщения и применения. «Прогресс», М., 1966.]
6. H. M. Wagner, A Linear Programming Solution to Dynamic Leontief Type Models, *Management Sci.*, 3, No. 3, 1957, pp. 234—254.
 7. G. B. Dantzig, *Linear Programming and Extension*, Princeton University Press, Princeton, N. J., 1963, chap. 25, sec. 3. [Русский перевод [5].]
 8. G. B. Dantzig and A. Madansky, On the Solution of Two—Stage Linear Programs under Uncertainty. The RAND Corporation. Paper P—2039, 1960.
 9. R. Gomory, Large and Nonconvex Problems in Linear Programming. *Proc. Symp. Appl. Math.*, No. 15, 1963, pp. 125—139, also IBM Rept. RC—765.
 10. G. B. Dantzig, *Linear Control Processes and Mathematical Programming*, *J. SIAM Control*, 4, No. 1, 1966, pp. 56—60.
 11. G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N. J., 1963, chap. 22 and 24.
 12. Л. С. Понтрягин, В. Г. Болтянский, Р. В. Гамкрелидзе, Е. Ф. Мищенко, *Математическая теория оптимальных процессов*, изд. 2-е, «Наука», 1969.
 13. J. M. Abadie and A. C. Williams, Dual and Parametric Methods in Decomposition, in «Recent Advances in Mathematical Programming», R. L. Graves and P. Wolfe, eds., McGraw-Hill, Inc., New York, 1963.
 14. J. B. Rosen, Convex Partition Programming, in «Recent Advances in Mathematical Programming», R. L. Graves and P. Wolfe, eds., McGraw-Hill, Inc., New York, 1963, pp. 159—176.
 15. J. B. Rosen and J. C. Ornea, Solution of Nonlinear Programming Problems by Partitioning, *Management Sci.*, 10, No. 1, 1963, pp. 160—173.
 16. G. Hadley, *Linear Programming*, Addison—Wesley Publishing Company, Inc., Reading, Mass., 1962, pp. 458—463.
 17. J. C. Ornea and G. G. Eldredge, Nonlinear Partitioned Models for Plant Scheduling and Economic Evaluation, *Proceedings of the A. I. Ch. E—Inst. Chem. Eng. Joint Meeting*, London, June 13—17, 1965, Symposium on Application of Mathematical Models in Chemical Engineering Research, Design, and Production.
 18. J. F. Benders, Partitioning Procedures for Solving Mixed—Variables Programming Problems. *Numerische Mathematik*, 4, 1962, pp. 238—252.
 19. B. R. Buzby, Nonlinear Distribution Problems, private communication, Dec. 1965.

ГЛАВА II

МЕТОД ДЕКОМПОЗИЦИИ ДАНЦИГА — ВУЛФА

§ 2.1. ВВЕДЕНИЕ

Опубликованный в 1960 году метод Данцига — Вулфа [1] положил начало последующим обширным работам по математическому программированию в условиях большой размерности. Метод оказался особенно эффективным для решения линейных задач, матрицы условий которых имеют блочно-диагональную структуру с небольшим числом общих связывающих ограничений.

Характерной особенностью этого метода является использование так называемой «координирующей задачи», которая по сравнению с исходной имеет небольшое число строк (не намного превышающее число связывающих ограничений) и, вообще говоря, очень много столбцов. При этом для решения координирующей задачи не требуется задание всех столбцов в явном виде. Они генерируются в процессе использования симплекс-метода. Поэтому такой подход мы называем *методом генерации столбцов*. Алгоритм включает итерационный обмен между множеством независимых подзадач, целевые функции которых включают варьируемые параметры, и координирующей задачей. В подзадачу вводится ряд параметров (двойственные переменные, оценки), получаемые при решении координирующей задачи. В свою очередь в координирующую задачу вводятся решения подзадач, которые оптимальным образом комбинируются и служат для получения новых оценок. Последние вновь вводятся в подзадачу, и итеративный процесс продолжается вплоть до этапа, на котором удовлетворяется критерий оптимальности. Такая процедура имеет изящную экономическую интерпретацию: ее, например, можно понимать как процедуру децентрализованного планирования, когда основной планирующий (управляющий) орган системы координирует функционирование отдельных подсистем с помощью цен на ограниченные ресурсы.

Эта глава посвящается различным аспектам метода Данцига — Вулфа. В начале доказываются теоремы, часто используе-

мыс ниже и дается описание алгоритма генерации столбцов. Затем проводится описание основного алгоритма, рассмотрены его модификации и разбираются числовые примеры. Дается экономическое толкование принципа разложения и определяется оценка снизу минимума функционала исходной задачи. Возможности метода разложения демонстрируются на примере транспортной и обобщенной транспортной задач, а также задачи распределения ресурсов. В заключительной части главы рассмотрены методы решения координирующей задачи, включая метод одновременного решения прямой и двойственной задач.

§ 2.2. ТЕОРЕМА О ВЫПУКЛЫХ КОМБИНАЦИЯХ

Принцип разложения Данцига—Вулфа основан на двух основных понятиях. Первое из них связано с методом генерации столбцов (он будет рассмотрен в следующем параграфе), а второе — с теоремой, которая гласит: точка принадлежит замкнутому, ограниченному выпуклому многограннику тогда и только тогда, когда она может быть представлена в виде выпуклой комбинации крайних точек этого многогранника.

При доказательстве теоремы существенно используются следующие свойства выпуклых множеств (доказательство их читатель может обнаружить в [2]).

1. Пусть X — выпуклое множество и y — элемент, не принадлежащий X . Тогда существует гиперплоскость, отделяющая y от X .

2. Если X — компактное выпуклое множество, то любая опорная гиперплоскость *) к X содержит крайнюю точку X .

Представляет интерес следующая теорема:

Теорема 1. Пусть X — компактное выпуклое множество в E^n ; $E(x)$ — множество его крайних точек и $C[E(x)]$ — выпуклая оболочка **) $E(X)$. Тогда

$$C[E(X)] = X.$$

Доказательство. (а) Покажем, что $X \supseteq C[E(X)]$. Для этого представим в виде следующей выпуклой комбинации произвольную точку y из $C[E(X)]$:

$$y = \sum_i \lambda_i x^i, \quad \lambda_i \geq 0, \quad \sum_i \lambda_i = 1, \quad x^i \in E(X), \quad (1)$$

и так как $x^i \in X$ и X — выпуклое множество, то $y \in X$.

*) Если z — граничная точка X , то гиперплоскость $cx = p$ называется его опорной гиперплоскостью в точке z , если $cz = p$ и X целиком лежит в одном из полупространств, выделяемых этой гиперплоскостью.

**) Наименьшее выпуклое множество, содержащее $E(X)$.

(в) Покажем теперь, что $C[E(X)] \ni X$. Пусть $x^* \in X$, но $x^* \notin C[E(X)]$. Тогда существует гиперплоскость, отделяющая x^* от $C[E(X)]$, т. е. существуют (c, α) такие, что

$$c'x^* = \alpha, \quad (2)$$

$$c'y < \alpha \text{ для всех } y \in C[E(X)]. \quad (3)$$

Пусть $\alpha^0 = \max\{c'x \mid x \in X\}$. Число α^0 существует, так как X — компактное множество. Тогда (c, α^0) определяет некоторую опорную гиперплоскость к X , так как $c'x \leq \alpha^0$ для всех $x \in X$. По определению эта гиперплоскость должна содержать x^* — крайнюю точку X , такую, что

$$c'x^* = \alpha^0. \quad (4)$$

Но это противоречит тому, что все точки $C[E(X)]$ (и таким образом, и все крайние точки множества X) удовлетворяют условию (3), так как $\alpha^0 \geq \alpha$.

Для случая, когда X имеет конечное число крайних точек, удобнее пользоваться другой формулировкой теоремы 1.

Теорема 2. Пусть $X = \{x \mid Ax = b, x \geq 0\}$ непустое ограниченное множество и x^i ($i = \overline{1, r}$) — его граничные точки. Тогда любой элемент $x \in X$ может быть представлен в виде

$$x = \sum_{i=1}^r \lambda_i x^i, \quad \lambda_i \geq 0, \quad i = 1, \dots, r, \quad (5)$$

$$\sum_{i=1}^r \lambda_i = 1.$$

Сформулируем также обобщение этой теоремы на случай, когда X не является ограниченным.

Теорема 3. Пусть множество $X = \{x \mid Ax = b, x \geq 0\}$ непусто. Тогда точка x принадлежит множеству X , тогда и только тогда, если она может быть представлена как сумма выпуклой комбинации граничных точек и линейной комбинации с неотрицательными коэффициентами крайних лучей однородных решений множества X , т. е.

$$x = \sum_i \lambda_i x^i,$$

где

$$\sum_i \lambda_i \delta_i = 1, \quad \lambda_i \geq 0$$

и δ_i равно 1 или 0, в зависимости от того, является x^i крайней точкой или лучом множества X .

Эта теорема будет нами использована при распространении принципа декомпозиции на случай, когда допустимые области в подзадачах могут быть неограничены (§ 3.7).

§ 2.3. МЕТОД ГЕНЕРАЦИИ СТОЛБЦОВ

Рассмотрим задачу линейного программирования:
минимизировать

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

при ограничениях

$$\sum_{j=1}^m p_j x_j = b, \quad x_j \geq 0 \quad (j = 1, \dots, n),$$

где p_j и b — m -мерные векторы, $m < n$. Предположим, что нам известно начальное базисное решение x_B и соответствующие ему матрица базиса B и вектор коэффициентов целевой функции c_B . Такое решение, если оно существует, может быть получено в результате выполнения начальной фазы общей процедуры симплекс-метода. Одновременно получаем вектор двойственных переменных

$$\pi = c_B B^{-1}.$$

Для того чтобы определить возможность улучшения первого допустимого решения, необходимо для каждого небазисного вектора p_j вычислить характеристическую разность (оценку способа):

$$c_j = c_j - \pi p_j.$$

Если $\min_j c_j = \bar{c}_s < 0$, то начальное решение можно улучшить путем включения в базис вектора x_s и выполнения преобразований (ведущей операции). Если имеется много столбцов (например, от нескольких тысяч до нескольких миллионов), то нахождение $\min_j \bar{c}_j$ путем вычислений каждого $\bar{c}_j$ и сравнения утомительно или практически невозможно. К счастью, в таких больших задачах множество столбцов технологической матрицы, как правило, имеет «хорошую» структуру, что в конечном счете объясняется специфическими структурными особенностями реальных систем, моделируемых в этих задачах. Если бы это было не так,

то задача сбора неупорядоченной числовой информации, необходимой для формирования всех столбцов, была бы безнадежной.

В качестве примера укажем на задачу о раскрое (§ 4.1). В этой задаче столбцы имеют неотрицательные целочисленные компоненты a_i , на которые наложены линейные ограничения

$$\sum_i a_i l_i \leq l,$$

где l_i, l — заданные числа. Другим примером является задача, возникающая вследствие применения принципа декомпозиции, где столбцы p_{ij} имеют вид $A_i x_i^j$, где A_i — i -я подматрица матрицы, составленной из коэффициентов общих связывающих ограничений, а x_i^j — j -я крайняя точка множества, определяемого i -м диагональным блоком. Таким образом, столбцы p_{ij} также определяются соответствующей системой линейных уравнений и неравенств. При общем рассмотрении будем предполагать, что все столбцы p_j выбираются из множества S , являющегося множеством всех возможных m -мерных векторов, удовлетворяющих некоторой системе равенств или неравенств. Поэтому теперь столбец (вектор), подлежащий включению в базис, может быть определен в результате решения подзадачи вида

$$\min_{p_j \in S} \{c(p_j) - \pi p_j\},$$

где $c(p_j) \equiv c_j$ — заданная функция вектора p_j . В зависимости от структуры множества S и вида функции $c(\cdot)$ выбирается наиболее эффективный метод решения приведенной выше подзадачи (например, метод линейного программирования для задачи с блочно-диагональной структурой технологической матрицы или методы динамического программирования для решения раскройной задачи). Такой прием называется *генерацией столбцов*, так как при решении подзадачи фактически используется лишь небольшое число столбцов множества, которые генерируются по мере необходимости. Таким образом, ни один из столбцов не хранится в памяти машины, что является очевидным преимуществом при решении задач большей размерности.

Очевидно, что размерность задачи определяется не только числом переменных, но и числом ограничений на эти переменные. В последнем случае весьма эффективным является преобразование исходной задачи в эквивалентную задачу линейного программирования с меньшим числом ограничений, которая, однако, содержит уже большое число переменных. Для решения эквивалентной задачи может быть использован метод генерации столбцов. Этот метод является **основой принципа разложения Данцига — Вулфа**, к изложению которого мы приступаем.

§ 2.4. ОПИСАНИЕ ПРИНЦИПА ДЕКОМПОЗИЦИИ

Рассмотрим задачу линейного программирования, матрица ограничений которой имеет блочно-диагональную структуру со связующими строками вида

$$A = \begin{bmatrix} A_1 & A_2 & \dots & A_p \\ B_1 & & & \vdots \\ & B_2 & & \vdots \\ & & \ddots & \vdots \\ & & & B_p \end{bmatrix},$$

где $p \geq 1$. Матрица любой задачи линейного программирования может быть приведена к такому виду при $p=1$ в результате соответствующего разбиения ограничений на два подмножества, поскольку стандартную задачу можно записать в виде.

минимизировать

$$z = cx \quad (1)$$

при ограничениях

$$\hat{A}_1 x = b_1 \quad (m_1 \text{ ограничений}), \quad (2)$$

$$A_2 x = b_2 \quad (m_2 \text{ ограничений}), \quad (3)$$

$$x \geq 0. \quad (4)$$

Предположим, что выпуклое многогранное множество

$$S_2 = \{x \mid A_2 x = b_2, x \geq 0\} \quad (5)$$

ограничено (это предположение, как будет показано ниже, не является обязательным). Тогда в соответствии с теоремой 2.2.2 любой элемент S_2 может быть представлен в виде

$$x = \sum_j \lambda_j x^j, \quad (6)$$

где

$$\sum_j \lambda_j = 1, \lambda_j \geq 0, \quad (7)$$

и x^j — крайние точки многогранника S_2 . Исходная задача (1) — (4) может быть трактована следующим образом: необходимо выбрать из всех решений системы (3) — (4) такое, которое удовлетворяет (2) и доставляет минимум функции z .

Подставив (6) в (2), получим

$$\sum_j (\hat{A}_1 x^j) \lambda_j = b_1. \quad (8)$$

Аналогичным образом получим новое выражение для целевой функции

$$z = \sum_i (cx^j) \lambda_j. \quad (9)$$

Обозначим

$$\hat{A}_1 x^j = p_j, \quad (10)$$

$$cx^j = f_j. \quad (11)$$

Учитывая (7) — (9), мы приходим к следующей задаче линейного программирования:

минимизировать

$$\sum_i f_i \lambda_i \quad (12)$$

при ограничениях

$$\sum_i p_i \lambda_i = b, \quad (13)$$

$$\sum_i \lambda_i = 1, \quad (14)$$

$$\lambda_i \geq 0. \quad (15)$$

Эта задача, эквивалентная исходной, называется *координирующей задачей*. Она имеет только $m_1 + 1$ строк по сравнению с $(m_1 + m_2)$ строками исходной задачи и большое число столбцов. (Если m_2 — достаточно большое число, то число столбцов, равное числу крайних точек S_2 , будет измеряться многими тысячами.) Чтобы не хранить в памяти машины все эти столбцы, будем их получать по мере необходимости, используя для этих целей метод генерации столбцов.

Рассмотрим следующую процедуру. Для каждого j вычислим характеристические разности

$$\bar{f}_j = f_j - \pi \begin{bmatrix} p_j \\ \vdots \\ 1 \end{bmatrix}. \quad (16)$$

Представим π в виде $\pi = (\pi_1, \pi_0)$, где π_1 — соответствует ограничениям (13), а скаляр π_0 — единственному ограничению (14).

Используя формулы для определения $\bar{f}_j$ и p_j , можно записать

$$\bar{f}_j = (c - \pi_1 \hat{A}_1) x^j - \pi_0. \quad (17)$$

В соответствии с обычной процедурой симплекс-метода для определения переменной, например, λ_0 , вводимой в базис,

необходимо найти

$$\min_i \bar{f}_i = \bar{f}_s = (c - \pi_1 \hat{A}_1) x^s - \pi_0. \quad (18)$$

Так как оптимальное решение задачи линейного программирования, множество ограничений которой является ограниченным множеством, достигается в крайней точке этого множества, выполнение операции (18) эквивалентно решению подзадачи вида:

$$\text{минимизировать} \quad (c - \pi_1 \hat{A}_1) x \quad (19)$$

при ограничениях

$$A_2 x = b_2, \quad x \geq 0. \quad (20)$$

Вычисление решения x^s этой задачи позволяет найти столбец, который следует ввести в базис координирующей задачи:

$$p_s = \begin{bmatrix} \hat{A}_1 x^s \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}. \quad (21)$$

Соответствующий коэффициент в целевой функции равен

$$f_s = c x^s. \quad (22)$$

Такой подход оказывается особенно эффективным, если p (число независимых диагональных блоков) больше единицы, т. е. если исходная задача может быть представлена в виде:

$$\text{минимизировать} \quad z = c_1 x_1 + c_2 x_2 + \dots + c_p x_p \quad (23)$$

при ограничениях

$$\begin{array}{rcl} A_1 x_1 + A_2 x_2 + \dots + A_p x_p & = & b_0, \\ B_1 x_1 & = & b_1, \\ B_2 x_2 & = & b_2, \\ & \vdots & \\ & B x_p & = b_p, \end{array} \quad (24)$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_p \geq 0, \quad p > 1. \quad (25)$$

Для такой задачи подзадача (19) — (20) имеет вид:

минимизировать

$$\sum_{i=1}^p (c_i - \pi_1 A_i) x_i \quad (26)$$

при ограничениях

$$B_i x_i = b_i, \quad x_i \geq 0 \quad (i = 1, \dots, p). \quad (27)$$

Из сепарабельности целевой функции (26) и независимости ограничений типа (27) немедленно следует, что решение задачи может быть получено в результате изолированного решения p задач вида:

минимизировать

$$(c_i - \pi_1 A_i) x_i \quad (28)$$

при ограничениях

$$B_i x_i = b_i x_i \geq 0. \quad (29)$$

Алгоритм разложения. Теперь можно сформулировать двух-уровневый алгоритм решения задачи (23) — (25), где на первом уровне необходимо решать подзадачи (28) — (29), а на втором — координирующую задачу.

Пусть уже имеется допустимое базисное решение задачи (12) — (15). Ему соответствует базисная матрица B и вектор оценок ограничений (π_1, π_0) .

Шаг 1. При использовании оценок π_1 решаются подзадачи (28) — (29) и находятся решения $x_i(\pi_1)$ и оптимальные значения целевых функций z_i^0 . Пусть

$$x(\pi_1) = (x_1(\pi_1), \dots, x_p(\pi_1)).$$

Шаг 2. Вычисляется $\min \bar{f}_j$, который в силу (17) равен

$$\min \bar{f}_j = \sum_{i=1}^p z_i^0 - \pi_0. \quad (30)$$

Если

$$\min \bar{f}_j \geq 0, \quad (31)$$

вычисления заканчиваются и определяется решение задачи (23) — (25):

$$x^0 = \sum_{j \text{ баз}} \lambda_j x^j, \quad (32)$$

где x^j — j -я крайняя точка множества S_2 , соответствующая базисной переменной λ_j .

Шаг 3. Если $\min \bar{f}_j < 0$, формируется столбец

$$p = \left(\sum_i A_i x_i(\pi_1) \right). \quad (33)$$

Столбец p умножается на матрицу B^{-1} и с использованием стандартной ведущей операции определяются новые обратная матрица и вектор оценок ограничений. После выполнения этих операций следует возвратиться к шагу 1 и повторить вычисления.

Если координирующая задача является невырожденной (или использован прием ε -возмущений) *), тогда на каждой итерации значение целевой функции убывает. Так как имеется только конечное число возможных базисов, и ни один из них не используется дважды, использование метода декомпозиции приведет к оптимальному решению за конечное число итераций.

Необходимо отметить следующее: решение задачи в целом x^0 , как правило, не является простым объединением решений подзадач. Из формулы (32) следует, что x^0 является выпуклой комбинацией ряда промежуточных решений. Невозможно получить оптимум системы в целом просто путем ввода в подзадачи (подсистемы) подходящих оценок, необходима определенная свобода комбинирования решений подзадач в общее решение. В этом смысле метод декомпозиции не является средством полной децентрализации процесса принятия решений. Более удачно определение Д. Данцига, который охарактеризовал принцип декомпозиции как «централизованное планирование без использования в центре полной информации» [8].

Ограниченная координирующая задача. Как было показано выше, в соответствии с принципом разложения Данцига — Вулфа, оптимизационная задача решается на первом уровне, но не на втором, где выполняется только ведущая операция. Возможна модификация этого метода, заключающаяся в решении задач линейного программирования на двух уровнях. В этом случае дополнительно решается так называемая ограниченная координирующая задача, которая получается из координирующей задачи (12) — (15) в результате «отбрасывания» всех столбцов, за исключением базисных и претендующих на включение в базис на текущей итерации. Такую ограниченную координирующую задачу можно представить в виде

минимизировать

$$f_1\lambda_1 + \dots + f_m\lambda_m + f\lambda \quad (34)$$

при ограничениях

$$p_1\lambda_1 + \dots + p_m\lambda_m + p\lambda = b_0, \quad (35)$$

$$\lambda_1 + \dots + \lambda_m + \lambda = 1, \quad (36)$$

$$\lambda_i \geq 0 \quad (i = 1, \dots, m), \quad \lambda \geq 0, \quad (37)$$

где $m = m_1 + 1$, λ_i — переменные текущего базиса и λ — переменная, вводимая в базис (для такой переменной $\bar{f} < 0$). Если текущий базис является невырожденным, то из базиса будет выводиться переменная, для которой $\bar{f} > 0$, и полученное новое решение будет оптимальным.

*) Д. ж. Данциг, *Линейное программирование, его применения и обобщения*, «Прогресс», М., 1966, стр. 22—29. (Прим. перев.)

Только в случае вырожденности базиса и если ведущий элемент — отрицательное число, потребуется более одной итерации для решения задачи (34) — (37). Таким образом, решение ограниченной задачи дает мало информации, однако, как будет показано ниже, возможны и другие более предпочтительные варианты постановки ограниченной координирующей задачи.

Варианты декомпозиции (разложения) прямой задачи. Имеется много способов разложения прямой задачи, каждый из которых приводит к своей форме координирующей и ограниченной координирующей задач.

Например, если решение системы

$$B_i x_i = b_i, \quad x_i \geq 0 \quad (38)$$

представить в виде

$$x_i = \sum_j \lambda_{ij} x_i^j, \quad (39)$$

где x_i^j , j — экстремальные точки многогранника $\{x | B_i x_i; x_i \geq 0\}$, то, как нетрудно показать, возникает следующая координирующая задача:

минимизировать

$$\sum_{i,j} f_{ij} \lambda_{ij} \quad (40)$$

при ограничениях

$$\sum_{i,j} p_{ij} \lambda_{ij} = b_0, \quad (41)$$

$$\sum_j \lambda_{ij} = 1 \quad (i = 1, \dots, p), \quad \lambda_{ij} \geq 0, \quad (42)$$

где

$$f_{ij} = c_i x_i^j, \quad (43)$$

$$p_{ij} = A_i x_i^j. \quad (44)$$

Эта задача отличается от координирующей задачи (12) — (14) в двух отношениях. Во-первых, она имеет не одно, а p ограничений вида (42), во-вторых, решение каждой подсистемы $B_i x_i = b_i$, $x_i \geq 0$ самостоятельно выражается через переменные λ_{ij} (в задаче (12) — (14) подсистемы рассматривались совместно). Последнее обстоятельство представляется важным с точки зрения вычислительных возможностей.

Чтобы убедиться в этом, применим симплекс-метод с использованием процедуры генерации столбцов для решения (40) — (42). Пусть B — базисная матрица допустимого решения размерности $(m_1 + p) \times (m_1 + p)$. Пусть $(\pi, \pi_{01}, \dots, \pi_{0p})$ — вектор

двойственных оценок для решения (π — оценка ограничений (41), π_{0i} — оценки, соответствующие ограничениям (42)). Вычислим теперь характеристические разности (оценки) для столбцов, соответствующих переменным λ_{ij} :

$$\bar{f}_{ij} = (c_i - \pi A_i) x_i^j - \pi_{0i}. \quad (45)$$

Нахождение $\min_j \bar{f}_{ij}$ для фиксированного i эквивалентно решению подзадачи вида:

минимизировать

$$(c_i - \pi A_i) x_i$$

при ограничениях

$$B_i x_i = b_i, \quad x_i \geq 0.$$

Мы пришли к подзадачам, рассмотренным выше. Обозначим минимальное значение целевой функции для таких задач через z_i^0 . Если

$$\min_i \min_j \bar{f}_{ij} = \min_i (z_i^0 - \pi_{0i}) \geq 0, \quad (46)$$

то текущее решение является оптимальным. В противном случае необходимо найти новый столбец, который должен быть введен в базис. В базис вводится столбец, на номере которого достигается

$$\min_{1 \leq i \leq p} (z_i^0 - \pi_{0i}). \quad (47)$$

Если этот минимум достигается на $i=s$ и $x_s(\pi_1)$ — решение подзадачи с номером s , то в базисе вводится столбец

$$\begin{bmatrix} A_s x_s(\pi_1) \\ \cdot \\ \cdot \\ u_s \end{bmatrix}, \quad (48)$$

где u_s — p -мерный вектор, все компоненты которого нули, за исключением s -й, равной единице. Хотя введение этого столбца в базис приводит к изменению вектора оценок (π_1, π_0) , подобная же операция генерации столбцов может быть осуществлена и на основе решений других подзадач. Некоторым из таких столбцов будут соответствовать отрицательные значения характеристических разностей (относительных оценок), полученных с применением двойственных оценок предыдущей итерации. Если хотя бы одна из этих характеристических разностей с использованием новой системы двойственных оценок остается отрицательной (что весьма вероятно, так как от итерации к итерации оценки не должны резко меняться), то возможно дальнейшее убывание

значения функции. Это наблюдение может служить достаточным основанием формирования новой ограниченной координирующей задачи, в которую включаются по одному новому столбцу из каждой подсистемы (подзадачи). Пусть $m = m_1 + p$, $\{p_{ij}\}$ — столбцы коэффициентов текущего базиса и пусть $p_1^*, \dots, p_p^*$ — столбцы, генерированные из решений соответствующих подзадач. Возникает следующая ограниченная координирующая задача:

минимизировать

$$\sum_{i=1}^p \sum_{j \in \text{баз}} f_{ij} \lambda_{ij} + \sum_i f_i^* \lambda_i \quad (49)$$

при ограничениях

$$\sum_{i=1}^p \sum_{j \in \text{баз}} p_{ij} \lambda_{ij} + \sum_i p_i^* \lambda_i = b_0, \quad (50)$$

$$\sum_{j \in \text{баз}} \lambda_{ij} + \lambda_i^* = 1, \quad i = 1, \dots, p, \quad (51)$$

$$\lambda_{ij} \geq 0, \quad \lambda_i^* \geq 0. \quad (52)$$

По сравнению с задачей (34) — (37) в этой задаче число переменных превышает число ограничений не на единицу, а на p . Можно ожидать, что здесь достижимо большее уменьшение целевой функции за один шаг. Однако при этом возрастает размерность базиса (с $(m_1 + 1)$ до $(m_1 + p)$). Возникающие в этом случае дополнительные вычислительные трудности могут быть успешно преодолены с помощью одного из вариантов симплексного метода — алгоритма учета обобщенных двухсторонних ограничений (см. § 5.4)*).

Таким образом, были представлены варианты координирующей задачи с одним и p ограничениями, соответствующие совместно и изолированному представлению диагональных блоков исходной задачи. Очевидно, что другие способы рассмотрения этих блоков повлекут появление новых вариантов координирующей задачи с любым числом ограничений k , причем $1 < k < p$. Хотя наиболее предпочтительным представляется вариант координирующей задачи с p ограничениями, при решении конкретных задач может оказаться более эффективным использование иного варианта такой задачи.

*) Более подробно преимущества такого подхода будут нами рассмотрены в § 3.2.

§ 2.5. ЧИСЛОВОЙ ПРИМЕР ИСПОЛЬЗОВАНИЯ ПРИНЦИПА ДЕКОМПОЗИЦИИ

Для иллюстрации принципа декомпозиции, рассмотрим задачу:

минимизировать

$$z = -x_1 + x_2 - 2y_1 - y_2$$

при ограничениях

$$\boxed{x_1 + 2x_2 + 2y_1 + y_2} \leq 40,$$

$$\boxed{\begin{array}{l} x_1 + 3x_2 \\ 2x_1 + x_2 \end{array}} \leq \begin{array}{l} 30, \\ 20, \end{array}$$

$$\boxed{\begin{array}{l} y_1 \\ y_2 \\ y_1 + y_2 \end{array}} \leq \begin{array}{l} 100, \\ 10, \\ 15, \end{array}$$

и кроме того, все переменные должны быть неотрицательными. Эта задача имеет одно общее связывающее ограничение и два независимых блока. Области допустимых решений неравенств обеих подсистем представлены на рис. 2.1.

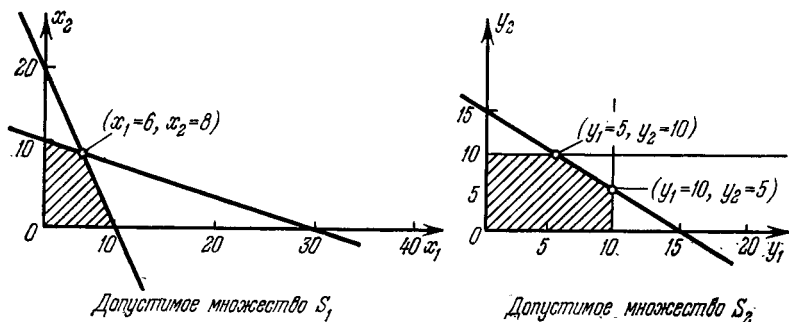


Рис. 2.1. Области допустимых решений подзадач.

Как уже было отмечено в § 2.4, разбиение (декомпозиция) условий подобной задачи может быть осуществлено разными способами и каждому способу разбиения будет соответствовать своя координирующая задача с определенным числом ограничений вида $\sum_j \lambda_{ij} = 1$. Постараемся использовать преимущества ограниченной координирующей задачи, которая в нашем случае будет иметь два ограничения. Пусть x и y — допустимые решения

неравенств блоков 1 и 2 соответственно. Можно записать

$$x = \sum_i \alpha_i x^i, \quad y = \sum_i \beta_i y^i,$$

где x^i, y^i — экстремальные точки этих блоков. В векторной форме связывающее ограничение и целевая функция могут быть представлены в виде

$$z = c_1 x + c_2 y, \quad a_1 x + a_2 y + s = 40,$$

где

$$c_1 = (-1, -1), \quad c_2 = (-2, -1), \quad a_1 = (1, 2), \quad a_2 = (2, 1), \quad s \geq 0$$

— дополнительная переменная. Запишем теперь координирующую задачу:

минимизировать

$$z = \sum_i (c_1 x^i) \alpha_i + \sum_i (c_2 y^i) \beta_i$$

при ограничениях

$$\sum_i (a_1 x^i) \alpha_i + \sum_i (a_2 y^i) \beta_i + s = 40,$$

$$\sum_i \alpha_i = 1,$$

$$\sum_i \beta_i = 1,$$

$$\alpha_i \geq 0, \quad \beta_i \geq 0.$$

В качестве начального допустимого решения можно выбрать

$$x^1 = y^1 = (0, 0), \quad \alpha_1 = \beta_1 = 1, \quad s = 40.$$

Начальная симплексная таблица имеет такой вид:

Базисные переменные	Столбцы матрицы условий координирующей задачи				
	s_1	α_1	β_1	$-z$	постоянные
s	1				40
α_1		1			1
β_1			1		1
$-z$				1	0

Итерация 1. В последней строке симплекс-таблицы находятся отрицательные двойственные оценки. Так как оценки, соответствующие найденному выше начальному допустимому

решению, равны нулю, то необходимо решить следующие подзадачи:

$$\begin{array}{ll} \text{минимизировать} & \text{минимизировать} \\ z_1 = c_1 x = -x_1 - x_2 & z_2 = c_2 y = -2y_1 - y_2 \\ \text{при ограничениях} & \text{при ограничениях} \\ x \in S_1; & y \in S_2. \end{array}$$

Оптимальные решения этих подзадач легко находятся с помощью рис. 2.1: $x^* = (6, 8)$, $z_1^0 = -14$, $y^* = (10, 5)$, $z_2^0 = -25$.

Определим минимальные оценки столбцов (характеристические разности):

$$\min \bar{f}_x = z_1^0 - \pi_{01} = -14,$$

$$\min \bar{f}_y = z_2^0 - \pi_{02} = -25.$$

Оптимальные решения подзадач образуют столбцы координирующей задачи. Столбцы, соответствующие таким решениям, имеют вид

$$\begin{pmatrix} a_1 x^2 \\ 1 \\ 0 \\ c_1 x^2 \end{pmatrix} = \begin{pmatrix} 22 \\ 1 \\ 0 \\ -14 \end{pmatrix}, \quad \begin{pmatrix} a_2 y^2 \\ 0 \\ 1 \\ c_2 y^2 \end{pmatrix} = \begin{pmatrix} 25 \\ 0 \\ 1 \\ -25 \end{pmatrix}.$$

Запишем теперь ограниченную координирующую задачу, используя текущий базис координирующей задачи и эти столбцы:

минимизировать

$$-14\alpha_2 - 25\beta_2$$

при ограничениях

$$s + 22\alpha_2 + 25\beta_2 = 40,$$

$$\alpha_1 + \alpha_2 = 1,$$

$$\beta_1 + \beta_2 = 1$$

и соблюдении условия неотрицательности всех переменных. Решая задачу модифицированным симплекс-методом, получим следующую серию таблиц:

Характеристические разности					
цикл	s	α_1	β_1	α_2	β_2
0	0	0	0	-14	-25
1	0	0	25	-14	0
2	$\frac{14}{22}$	0	$\frac{200}{22}$	0	0

Цикл 0: β_2 вводится, β_1 выводится

Базисные переменные	Столбцы симплексной таблицы координирующей задачи				свободные члены	Преобразований столбец, входящий в базис (ведущий столбец)
	s	α_1	β_1	$-z$		
s	1				40	25
α_1		1			1	0
β_1			1		1	1
$-z$				+1	0	25

Цикл 1: α_2 вводится, s — выводится

s	1		-25		15	22
α_1		1			1	1
β_2			1		1	0
$-z$			25	1	25	-14

Цикл 2: проверка оптимальности

α_2	$\frac{1}{22}$		$\frac{25}{22}$			$\frac{15}{22}$
α_1	$\frac{1}{22}$	1	$\frac{25}{22}$			$\frac{7}{22}$
β_2			1			1
$-z$	$\frac{14}{22}$		$\frac{200}{22}$	1	34	$\frac{6}{11}$

Таким образом, получено новое оптимальное решение исходной задачи

$$x = \alpha_1 x^1 + \alpha_2 x^2 = \frac{7}{22} (0, 0) + \frac{15}{22} (6, 8),$$

$$y = \beta_2 x_2 = (10, 5)$$

со значением целевой функции равным $-33 \frac{6}{11}$.

Итерация 2. В последней строке заключительной симплекс-таблицы находятся новые оценки ограничений

$$(\pi_1, \pi_{01}, \pi_{02}) = \left(-\frac{14}{22}, 0, -\frac{200}{22} \right),$$

которые используются для формирования новых целевых функций подзадач:

$$z_1 = (c_1 - \pi a_1) x = \left[(-1, -1) + \frac{14}{22} (1, 2) \right] x = -\frac{8}{22} x_1 + \frac{6}{22} x_2,$$

$$z_2 = (c_2 - \pi a_2) y = \left[(-2, -1) + \frac{14}{22} (2, 1) \right] y = -\frac{16}{22} y_1 - \frac{8}{22} y_2.$$

Минимизируя эти функции при соблюдении соответствующих ограничений, получим оптимальные решения:

$$x^3 = (10, 0) \quad z_1^0 = -80/22,$$

$$y^3 = (10, 5) \quad z_2^0 = -200/22,$$

которым соответствуют следующие минимальные характеристические разности:

$$\min \bar{f}_x = z_1^0 - \pi_{01} = -80/22,$$

$$\min \bar{f}_y = z_2^0 - \pi_{02} = 0,$$

и столбцы

$$\begin{pmatrix} a_1 x^3 \\ 1 \\ 0 \\ c_1 x^3 \end{pmatrix} = \begin{pmatrix} 10 \\ 1 \\ 0 \\ -10 \end{pmatrix}, \quad \begin{pmatrix} a_2 y^3 \\ 0 \\ 1 \\ c_2 y^3 \end{pmatrix} = \begin{pmatrix} 25 \\ 0 \\ 1 \\ -25 \end{pmatrix}.$$

Новая ограниченная координирующая задача включает все оптимальные базисные столбцы предыдущей координирующей задачи (остальные столбцы отбрасываются) и новые столбцы:

минимизировать

$$-14\alpha_2 - 25\beta_2 - 10\alpha_3 - 25\beta_3$$

при ограничениях

$$22\alpha_2 + 25\beta_2 + 10\alpha_3 + 25\beta_3 = 40,$$

$$\alpha_1 + \alpha_2 \quad + \quad \alpha_3 \quad = 1,$$

$$\beta_2 + \quad \beta_3 = 1.$$

Оптимальное решение предыдущей ограниченной задачи используется в качестве начального базисного допустимого решения этой задачи. Обратная матрица базиса приведена в последней таблице, полученной на итерации 1.

Используя теперь модифицированный симплекс-метод, получим:

Характеристические разности					
цикл	α_1	α_2	α_3	β_1	β_2
0	0	0	$-\frac{80}{22}$	0	0
1	$\frac{80}{12}$	0	0	0	0

Цикл 0: α_3 вводится, α_1 выводится из базиса

Базисные переменные	Столбцы координирующей задачи				Свободные члены	Преобразованный столбец, входящий в базис
	s	α_1	β_1	$-z$		
α_2	$1/22$		$-25/22$		$15/22$	$10/22$
α_1	$-1/22$	1	$25/22$		$7/22$	$12/22$
β_2			1		1	0
$-z$	$14/22$		$\frac{200}{22}$	1	$36 \ 6/11$	$-86/22$

Цикл 1: проверка оптимальности

α_2	$1/12$	$-10/12$	$-25/12$	$5/12$
α_3	$-1/12$	$22/12$	$25/12$	$7/12$
β_2			1	1
$-z$	$4/12$	$80/12$	$200/12$	$1 \ 110/3$

Получено допустимое решение исходной задачи:

$$x = \alpha_1 x^2 + \alpha_3 x^3 = 5/12(6,8) + 7/12(10, 0),$$

$$y = \beta_2 y^2 = (10, 5),$$

а значение целевой функции равно $z = -\frac{110}{3}$.

Итерация 3. Вектор оценок и целевые функции подзадач имеют вид

$$(\pi_1, \pi_{01}, \pi_{02}) = -(4/12, 80/12, 200/12),$$

$$z_1 = [(-1, 1) + 4/12(1, 2)]x = \frac{2}{3}x_1 - \frac{1}{3}x_2,$$

$$z_2 = [(-2, -1) + 4/12(2, 1)]y = -\frac{4}{3}y_1 - \frac{2}{3}y_2;$$

найдем оптимальные решения подзадач

$$x^4 = (6, 8) \quad \text{или} \quad x^5 = (10, 0), \quad z_1^0 = -20/3,$$

$$y^4 = (10, 5) \quad z_2^0 = -\frac{50}{3}$$

и значения минимальных характеристических разностей:

$$\min \bar{f}_x = z_1^0 - \pi_{01} = 0,$$

$$\min \bar{f}_y = z_2^0 - \pi_{02} = 0,$$

$$\min \bar{c}_s = -\pi_{01} = 80/12.$$

Так как все они неотрицательны, полученное решение является оптимальным.

§ 2.6. ЭКОНОМИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ПРИНЦИПА РАЗЛОЖЕНИЯ *)

Принцип разложения Данцига — Вулфа имеет интересную экономическую интерпретацию, основанную на рассмотрении оценок π в качестве цен ресурсов. Пусть известны базисная матрица B и соответствующее ей базисное допустимое решение

$$x_B = B^{-1}b. \quad (1)$$

Как изменится значение целевой функции z при изменении вектора ограничений на величину db при условии сохранения базиса B ? Каков вообще смысл выражения $\partial z / \partial b_i$? Ясно, что для осмысленности вопроса необходимо, чтобы новое решение

$$x_B^n = B^{-1}(b + db) \quad (2)$$

было неотрицательно для любой произвольной величины db . Это условие будет выполнено, если мы предположим положительность всех компонент вектора x_B , т. е. невырожденность решения. По определению имеем

$$\pi = c_B B^{-1}, \quad (3)$$

$$z = c_B x_B = c_B (B^{-1}b) = (c_B B^{-1})b. \quad (4)$$

Таким образом,

$$z = \pi b = \sum_i \pi_i b_i. \quad (5)$$

*) Более детальные сведения по этому вопросу читатель может обнаружить в работах Фабиана [7] и Данцига [8].

Так как мы сделали предположение о невырожденности x_b , π не изменится при малых изменениях db , поскольку базис B не изменится. Поэтому из выражения (5) получим

$$\frac{\partial z^*}{\partial b_i} = \pi_i \quad (i = 1, \dots, m). \quad (6)$$

Если, например, целевая функция измеряется в денежных единицах (долларах), то и π_i будет выражаться в этих же единицах (долларах) на единицу ресурса b_i . Оценки π_i обычно называют «теневыми ценами» ингредиента ресурса « i », так как они показывают выигрыш в величине целевой функции при малом увеличении количества ресурса при условии, что все ингредиенты преобразуются в интенсивности технологических способов согласно (1). Слово «теневые» мы заключили в кавычки, так как π_i , вообще говоря, не являются действительными рыночными ценами. Отметим также, что B не обязательно является оптимальной базисной матрицей, и тогда оценки π_i могут принимать как положительные, так и отрицательные значения.

После этих замечаний, возвращаясь к толкованию принципа разложения, рассмотрим воздействие изменений в функционировании i -й подсистемы на общую целевую функцию задачи, трактуемую как совокупные затраты системы. Если i -я подсистема выберет интенсивность работы x_i , то решение приведет к прямым затратам $c_i x_i$. При этом будут израсходованы общесистемные ресурсы в объеме $A_i x_i$. Одновременно на эту же величину уменьшится общий лимит ресурсов для других подсистем, что в свою очередь может вызвать у них увеличение затрат. Поэтому i -я подсистема при выборе вектора интенсивностей должна быть поставлена в условия, при которых она учитывала бы возможность такого неприятного, но важного вклада в затраты. Простейший способ реализации этих условий заключается в том, чтобы ввести «теневые» внутренние цены на общие ограниченные ресурсы и заставить подсистемы платить за используемые общие ресурсы. Если правильно назначить такие цены, то они будут препятствовать образованию излишков в отдельных подсистемах, способствуя их экономному использованию.

Легко понять, что принцип декомпозиции координирует функционирование подсистем именно в этом направлении. В самом деле, целевая функция i -й подзадачи в силу (24)—(28) имеет вид

$$z_i = c_i x_i - \pi A_i x_i. \quad (7)$$

Слагаемое $c_i x_i$ задает прямой вклад в общесистемные затраты при использовании технологических способов с интенсивностью x_i , в то время как $\pi A_i x_i$ определяет плату i -й подсистемы за использование ограниченных общесистемных ресурсов в объ-

еме $A_i x_i$ (π — вектор оценок этих ресурсов). Таким образом, в состав общих издержек i -й подсистемы входят как прямые, так и косвенные затраты. В правой части (7) второй член имеет отрицательный знак. Это объясняется следующим образом. Использование ингредиента j в объеме $(\{A_i x_i\}_j > 0)$ ведет к уменьшению его общего количества. Если ингредиент j является лимитирующим фактором производственного процесса, то это снижение приведет к возрастанию затрат системы, поэтому с учетом (6) $\pi_j < 0$. Из этого следует, что в качестве «цены» ресурса следует рассматривать $(-\pi_j)$ — положительную величину, тогда положительна и величина $(-\pi_j(A_i x_i)_j)$ — приращение затрат подсистемы вследствие использования ресурса j .

Аналогичным образом можно интерпретировать и критерий оптимальности

$$\min_i \bar{f}_{ij} = z_i^0 - \pi_{0i} \geq 0 \quad (8)$$

или

$$z_i^0 \geq \pi_{0i} \quad (i = 1, \dots, p), \quad (9)$$

где z_i^0 — оптимальное значение целевой функции i -й подзадачи и π_{0i} — оценка ограничения

$$\sum_j \lambda_{ij} = 1. \quad (10)$$

Из (6) следует, что π_{0i} — показатель изменения целевой функции системы при малом изменении правой части уравнения (10). Увеличение правой части этого уравнения означает повышение значений взвешивающих коэффициентов λ_{ij} , измеряющих относительную значимость решений соответствующих подсистем. Отсюда следует, что π_{0i} — предельная текущая оценка таких решений.

Как уже было показано выше, z_i^0 — величина общих затрат (прямых и косвенных) i -й подсистемы при плане x_i^0 . Поэтому критерий оптимальности (9) имеет следующий смысл: если изменение текущих планов подсистем приведет к возрастанию затрат этих подсистем, то соответствующая комбинация первоначальных планов является оптимальной.

Если условие (9) не удовлетворяется, то полученный план не является оптимальным и на основе решений подзадач находится новое решение общей задачи.

По поводу изложенной выше интерпретации метода разложения возникает следующий вопрос. Как нетрудно было заметить, схема принятия решения, изложенная выше, не является полностью децентрализованной, так как планирующий орган, реализуя координирующую задачу, выбирает окончательное

решение, взвешивая план отдельных подсистем. А нельзя ли построить такую систему цен, руководствуясь которыми подсистемы в условиях полной децентрализации выбирали бы оптимальные планы?

Любопытно отметить, что для систем, описываемых условиями линейных задач, в общем случае это требование не выполняется. Дело в том, что независимо от типа целевой функции отдельных подсистем, их планы x_i являются крайними точками множества ограничений.

Однако решение исходной задачи $x_1^*, \dots, x_p^*$, вообще говоря, уже не обязательно является таким, что x_i^* есть крайняя точка соответствующего множества ограничений, так как базисное решение блочной задачи со связывающими ограничениями (2.4) — (2.3) — (2.4) — (2.5) может иметь компоненты, которые не являются крайними точками диагональных блоков. Оптимальный вектор x_i^* может быть внутренней точкой подсистемы ограничений. Внутренние же точки могут быть получены только как комбинации различных крайних точек. Очевидно, что в случае нелинейности целевых функций подзадач небазисные решения могут быть получены и непосредственно. Таким образом, для некоторых нелинейных систем условия полной децентрализации могут быть реализованы. Эта проблема более подробно будет рассмотрена в главе VII.

§ 2.7. НИЖНЯЯ ГРАНИЦА ЗНАЧЕНИЙ ВЕЛИЧИНЫ МИНИМАЛЬНЫХ ЗАТРАТ

Рассмотрим опять координирующую задачу (2.4.40) — (2.4.42), которую представим в виде:

минимизировать

$$z = \sum_{i,j} f_{ij} \lambda_{ij} \quad (1)$$

при ограничениях

$$\sum_{i,j} p_{ij} \lambda_{ij} = b_0, \quad (2)$$

$$\sum_j \lambda_{ij} = 1 \quad (i = 1, \dots, p), \quad (3)$$

$$\lambda_{ij} \geq 0. \quad (4)$$

Умножив равенства (2) и (3) соответственно на двойственные π_1 и переменные π_{0i} , просуммировав и вычитая их из функционала (7), получим

$$z - \pi_1 b_0 - \sum_i \pi_{0i} = \sum_{i,j} \lambda_{ij} (f_{ij} - \pi_1 p_{ij} - \pi_{0i}). \quad (5)$$

Число, заключенное в круглые скобки в правой части выражения (5), является характеристической разностью $\bar{f}_{ij}$ переменной λ_{ij} . В результате решения подзадач (3.4.28) — (3.4.29) определяются $\min_i \bar{f}_{ij}$ для всех i . Так как λ_{ij} является неотрицательной переменной, то $\bar{f}_{ij}$ может быть заменена на $\min_i \bar{f}_{ij}$; при этом правая часть уравнения (5) не возрастет, поэтому

$$z - \pi_1 b_0 - \sum_i \pi_{0i} \geq \sum_{i,j} \lambda_{ij} (\min_i \bar{f}_{ij}), \quad (6)$$

$$z - \pi_1 b_0 - \sum_i \pi_{0i} \geq \sum_i (\min_j \bar{f}_{ij}). \quad (7)$$

Учитывая (3), неравенство (7) можно записать так:

$$z \geq \sum_i (\min_j \bar{f}_{ij}) + \pi_1 b_0 + \sum_i \pi_{0i}. \quad (8)$$

Так как условие (8) выполняется для всех значений z , полученных в результате решения задачи (1) — (4), оно будет выполняться и для минимального значения z ; поэтому

$$\min z \geq \sum_i (\min_j \bar{f}_{ij}) + \pi_1 b_0 + \pi_0 e_p, \quad (9)$$

где e_p — p -мерный вектор, с компонентами, равными единице.

Неравенство (9) можно записать в более компактной форме. Для этого запишем два последних члена этого неравенства в виде

$$(\pi_1, \pi_0) \begin{pmatrix} b_0 \\ e_p \end{pmatrix} = c_B B^{-1} \begin{pmatrix} b_0 \\ e_p \end{pmatrix} = c_B \lambda_B = z_B, \quad (10)$$

где z_B — значение целевой функции z , соответствующее текущему базису B . Поэтому неравенство (9) можно записать в виде

$$\min z \geq z_B + \sum_{i=1}^p (\min_j \bar{f}_{ij}). \quad (11)$$

Таким образом, найдена нижняя граница значений величины минимальных затрат. Отметим, что числа $\min_j \bar{f}_{ij}$ должны находиться в процессе реализации симплексного метода, поэтому для ее нахождения не потребуются больших дополнительных вычислений.

Далее, нетрудно видеть, что в оптимальном решении неравенство (11) превращается в равенство. Напомним, что для

оптимального плана выполняется условие

$$\min_j f_{ij} > 0. \quad (12)$$

Но j — индекс крайней точки системы ограничений подзадачи и для крайних точек, соответствующих базисным переменным x_{ij} , $\bar{f}_{ij} = 0$. Таким образом, если $z_B = \min z$, то

$$\min_j \bar{f}_{ij} = 0 \quad (13)$$

и неравенство (11) превращается в равенство. Если правая часть условия (11) вычисляется на каждой итерации, то соответствующая последовательность чисел за конечное число шагов сходится к $\min z$ (но не обязательно монотонно). Эта последовательность вместе с последовательностью получаемых значений z может быть использована для определения границ изменения показателя минимальных затрат и целесообразности прекращения вычислений в случае, например, когда ожидаемое отклонение значения целевой функции от оптимального не будет превышать α процентов.

Модификация для случая неограниченности множеств ограничений подзадач. До этого мы предполагали, что многогранник

$$S_i = \{x \mid B_i x = b_i, x \geq 0\} \quad (i = 1, \dots, p) \quad (14)$$

ограничен.

Если это не так, то теорема о том, что любая точка множеств S_i может быть представлена как выпуклая комбинация крайних точек, уже не выполняется. Но именно на этой теореме было основано описание принципа разложения. Возможна модификация этого принципа при использовании теоремы § 2.2: любая точка множества S_i может быть представлена как сумма выпуклой комбинации крайних точек S_i и неотрицательной комбинации экстремальных лучей S_i . Экстремальные лучи являются решениями однородной системы

$$B_i x = 0, \quad x \geq 0. \quad (15)$$

Число таких лучей конечно.

Неограниченное решение линейной задачи можно получить в результате движения по экстремальному лучу. Такая ситуация (если она существует) может быть обнаружена с помощью симплекс-метода. Используя этот результат, представим элемент x множества S_i в виде

$$x = \sum_i \lambda_i x^i \quad (16)$$

$$\sum_i \lambda_i \delta_i = 1, \quad \lambda_i \geq 0, \quad (17)$$

где

$$\delta_i = \begin{cases} 1, & \text{если } x^i \text{ — крайняя точка } S_i, \\ 0, & \text{если } x^i \text{ — экстремальный луч } S_i. \end{cases} \quad (18)$$

Подставив (16) в связывающие ограничения и целевую функцию исходной блочной задачи, получим задачу, для решения которой может быть использован принцип декомпозиции. Координирующая задача в этом случае будет отличаться тем, что вместо условия $\sum_i \lambda_i = 1$ появятся условия (17) и (18).

§ 2.8. ИСПОЛЬЗОВАНИЕ МЕТОДА ДЕКОМПОЗИЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ ТРАНСПОРТНОГО ТИПА

Содержание транспортной задачи Хичкока — Купманса *) таково: имеется m пунктов производства и n пунктов потребления однородной продукции. Заданы объемы производства a_i и потребления b_j . Требуется найти такой план перевозок, при котором спрос был бы полностью удовлетворен, а суммарные затраты на перевозку были бы минимальны.

Пусть x_{ij} — количество продукции, отправленной из пункта производства i в пункт потребления j ; c_{ij} — затраты на перевозку единицы груза из i в j ; возникает задача минимизировать

$$\sum_{i,j} c_{ij} x_{ij} \quad (1)$$

при ограничениях

$$\sum_j x_{ij} = a_i \quad (i = 1, \dots, m), \quad (2)$$

$$\sum_i x_{ij} = b_j \quad (j = 1, \dots, n), \quad (3)$$

$$x_{ij} \geq 0, \quad (4)$$

где a_i и b_j — заданные положительные константы и $\sum_i a_i = \sum_j b_j$, что гарантирует совместность ограничений (2) и (3).

Решение транспортной задачи обладает известным свойством: если a_i и b_j — целые числа, то и оптимальное решение задачи будет целочисленным.

Для решения транспортной задачи (1)–(4) предложено много весьма эффективных методов, большинство из которых

*) С историей транспортной задачи читатель может ознакомиться в книге Данцига «Линейное программирование, его обобщения и применения», «Прогресс», М., 1966.

комбинаторного типа. Однако при решении транспортных задач с 5000 и более ограничениями возникают серьезные трудности из-за недостатка памяти машины или больших затрат времени. Если число пунктов потребления намного превышает число пунктов производства $n \gg m$, то появляется возможность воспользоваться структурными особенностями матрицы условий задачи. Это позволило Вильямсу [3] использовать метод разложения для решения транспортных задач большой размерности. При этом множество ограничений (3) соответствует ограничениям подзадачи, и ограничения (2) рассматриваются как связывающие. В результате координирующая задача будет иметь только « m » ограничений, при этом, как будет показано ниже, условия выпуклости могут не учитываться. Так как каждая переменная появляется только в одном из уравнений (3), то подзадачи легко решаются путем просматривания соответствующих столбцов матрицы затрат (c_{ij}).

Аналогичный подход реализуется и для случая, когда $m \gg n$. Пример такой ситуации будет рассмотрен нами в следующем параграфе.

Пусть x_{ij}^k — компоненты k -й крайней точки множества ограничений (3) — (4); представим любое решение этой системы уравнений в виде

$$x_{ij} = \sum_k \lambda_k x_{ij}^k, \quad \sum_k \lambda_k = 1, \quad \lambda_k \geq 0. \quad (5)$$

Подставим выражение (5) в условия (1) и (2). В результате получим следующую координирующую задачу:

минимизировать

$$\sum_k f_k \lambda_k \quad (6)$$

при ограничениях

$$\lambda_k \geq 0, \quad \sum_k v_{ik} \lambda_k = a_i \quad (i = 1, \dots, n), \quad (7)$$

где

$$f_k = \sum_{i,j} c_{ij} x_{ij}^k, \quad (8)$$

$$v_{ik} = \sum_j x_{ij}^k. \quad (9)$$

Условие выпуклости $\sum_k \lambda_k = 1$ учитывается при этом через ограни-

чение (7), так так

$$\sum_i a_i = \sum_{i,k} \lambda_k v_{ik} = \sum_k \lambda_k \sum_i \sum_j x_{ij}^k, \quad (10)$$

$$\sum_i a_i = \sum_{i,k} \lambda_k v_{ik} = \sum_k \lambda_k \sum_i \sum_j x_{ij}^k = \sum_k \lambda_k \left(\sum_j b_j \right) \quad (11)$$

и

$$\sum_i a_i = \sum_j b_j > 0.$$

Чтобы записать подзадачу, определим характеристические разности

$$\bar{f}_k = f_k - \sum_i \pi_i v_{ik}, \quad (12)$$

$$\bar{f}_k = \sum_{i,j} (c_{ij} - \pi_i) x_{ij}^k. \quad (13)$$

Так как x_{ij}^k — крайние точки множества (3) — (4), то нахождение $\min \bar{f}_k$ равносильно решению подзадач:

минимизировать

$$\sum_{i,j} (c_{ij} - \pi_i) x_{ij} \quad (14)$$

при ограничениях

$$x_{ij} \geq 0, \quad \sum_i x_{ij} = b_j \quad (j = 1, \dots, n). \quad (15)$$

Функционал (14) — сепарабельная функция и каждая переменная находится только в одном из уравнений (15). Поэтому решение задачи (14) — (15) эквивалентно решению для каждого j задачи

$$\sum_i (c_{ij} - \pi_i) x_{ij} = \min \quad (16)$$

при ограничениях (15). Решение такой задачи заключается в нахождении минимального коэффициента в (16) и приравнивании соответствующей переменной x_{ij} числу b_j , т. е. надо найти

$$\min_i (c_{ij} - \pi_i) = c_{s_j j} - \pi_{s_j}, \quad (17)$$

$$x_{ij}^k = \begin{cases} b_j, & i = s_j, \\ 0, & i \neq s_j. \end{cases} \quad (18)$$

Учитывая (13), минимальную характеристическую разность определим по формуле

$$\min \bar{f}_k = \sum_i b_i [\min_i (c_{ij} - \pi_i)]. \quad (19)$$

Если эта разность отрицательна, то, используя (18), находим коэффициент v_{ik} и вводим новый столбец в базис.

Операция нахождения минимума (17) может быть осуществлена в результате сравнения элементов столбцов матрицы затрат, в которой все элементы i -й строки уменьшаются на величину π_i . Соответствующее решение (18) характеризуется тем, что пункты потребления i закрепляются за наиболее «дешевыми» (в смысле затрат на перевозку) пунктами производства j .

§ 2.9. ОБОБЩЕННЫЕ ТРАНСПОРТНЫЕ ЗАДАЧИ И ЗАДАЧА О РУБКЕ ЛЕСА

Обобщенная транспортная задача заключается в нахождении неотрицательных переменных y_{ij} , минимизирующих линейный функционал

$$z = \sum_{i,j} \hat{c}_{ij} y_{ij} \quad (1)$$

и удовлетворяющих ограничениям

$$\sum_i a_{ij} y_{ij} = a_i \quad (i = 1, \dots, n), \quad (2)$$

$$\sum_j b_{ij} y_{ij} = b_j \quad (j = 1, \dots, m), \quad (3)$$

где a_i и b_j неотрицательные и a_{ij} — положительные числа. Здесь, так же как и в задаче Хичкока — Купманса, каждая переменная y_{ij} может иметь только два отличных от нуля коэффициента, но не обязательно равных единице. Эта задача может быть представлена и в стандартной форме, если осуществить замену переменных

$$x_{ij} = a_{ij} y_{ij}.$$

Так как a_{ij} положительны, то $x_{ij} \geq 0 \Leftrightarrow y_{ij} \geq 0$ и задача (1) — (3) переписывается в таком виде:

минимизировать

$$z = \sum_{i,j} c_{ij} x_{ij} \quad (4)$$

при ограничениях

$$\sum x_{ij} = a_i \quad (i = 1, \dots, m), \quad (5)$$

$$\sum d_{ij} x_{ij} = b_j \quad (j = 1, \dots, n), \quad (6)$$

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

где

$$d_{ij} = \frac{b_{ij}}{a_{ij}}, \quad c_{ij} = \hat{c}_{ij}/a_{ij}.$$

Такие задачи часто встречаются в практических приложениях [4]—[5]. Примером является задача о рубке леса, сформулированная Ченгом [6]. Содержание этой задачи сводится к следующему. Имеется большое число лесных участков (более тысячи). Известны величина дохода, получаемого с участка i в зависимости от объема рубок в году j , ограничения снизу и сверху на общий объем рубки каждого года. Требуется определить такую последовательность рубок во времени, которая обеспечила бы получение максимального суммарного дохода за весь рассматриваемый период.

Введем обозначения:

r_i — площадь (в га) i -го лесного участка; $b_j^{\min}, b_j^{\max}$ — минимальная и максимальная площадь, на которой могут быть осуществлены рубки в году j ; c_{ij} — доход, который может быть получен с участка i , если рубки на нем будут осуществляться в году j ; x_{ij} — площадь i -го участка (в долях единицы), подвергнутая рубке в j -м году.

С использованием этих обозначений задачу о рубках леса можно записать в следующем виде:

максимизировать

$$z = \sum_{i,j} c_{ij} x_{ij} \quad (7)$$

при ограничениях

$$\sum_{i=1}^m r_i x_{ij} + u_j = b_j^{\max} \quad (j = 1, \dots, n), \quad (8)$$

$$\sum_{i=1}^m r_i x_{ij} - v_j = b_j^{\min} \quad (j = 1, \dots, n), \quad (9)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad (i = 1, \dots, m), \quad (10)$$

$$x_{ij} \geq 0, \quad u_j \geq 0, \quad v_j \geq 0,$$

где u_j и v_j — дополнительные переменные, смысл которых очевиден. Произведем некоторые преобразования. Вычитая (9) из (8), получим:

максимизировать

$$z = \sum_{i,j} c_{ij} x_{ij} \quad (11)$$

при ограничениях

$$\sum_{j=1}^n x_{ij} = 1 \quad (i = 1, \dots, m), \quad (12)$$

$$u_i + v_j = b_j^{\max} - b_j^{\min} \quad (j = 1, \dots, n), \quad (13)$$

$$\sum_i r_i x_{ij} + u_j = b_j^{\max} \quad (j = 1, \dots, n). \quad (14)$$

Задача (11) — (14) является обобщенной транспортной задачей с верхними ограничениями на значение дополнительных переменных u_j . Основная трудность решения задачи (11) — (14) заключается в наличии колоссального числа лесных участков, которым соответствуют ограничения (12). В отличие от рассмотренной выше задачи, здесь мы встречаемся с обобщенной транспортной задачей, в которой число пунктов производства намного превышает число «пунктов потребления», т. е. $m \gg n$.

Чтобы воспользоваться приемом, изложенным в § 2.8, будем рассматривать ограничения (13) — (14) как связывающие ограничения, а ограничения (12) пусть соответствуют условиям отдельных подзадач. Представим условия задачи в векторно-матричной форме:

максимизировать

$$z = cx \quad (11')$$

при ограничениях

$$\sum_{j=1}^n x_{ij} = 1 \quad (i = 1, \dots, n), \quad (12')$$

$$u \leq b^{\max} - b^{\min}, \quad (13')$$

$$A_1 x + u = b^{\max}. \quad (14')$$

Пусть x^k — k -я крайняя точка множества ограничений (12). Возникает следующая координирующая задача:

максимизировать

$$z = \sum_k (cx^k) \lambda_k \quad (15)$$

при условиях

$$\sum_k (A_1 x^k) \lambda_k + u = b^{\max}, \quad (16)$$

$$\sum_k \lambda_k = 1, \quad (17)$$

$$u \leq b^{\max} - b^{\min}, \quad (18)$$

$$u \geq 0, \quad \lambda_k \geq 0. \quad (19)$$

Подзадача имеет вид, рассмотренный в § 2.8. Пусть π_j — двойственная оценка j -го ограничения системы (16). Для решения подзадачи теперь необходимо только сравнить между собой элементы модифицированной матрицы затрат $(c_{ij} - a_{ij}\pi_j)$. Что касается условия (18), то оно может быть учтено путем использования метода двухсторонних ограничений или путем введения соответствующего столбца в базис координирующей задачи.

Такой прием был использован Ченгом [6] при решении задачи о рубке леса на 1166 участках в течение 24-летнего периода при $b_j^{\max} = 9700$ га и $b_j^{\min} = 8900$ га. Размеры участков колебались в пределах от 30 до 1000 га и доход от каждого участка возрастал линейно с увеличением номера года планового периода. Решение задачи было осуществлено на ЭВМ IBM 7044 с емкостью памяти 32 К с использованием двойной точности. Результаты расчетов приведены в таблице 2.9.1. Из таблицы видно, что на

Таблица 2.9.1

Номер итера- ции	Общий доход	Увеличение значения целе- вой функции за каждые 90 итераций	Общее время счета (мин.)	Номер итера- ции	Общий доход	Увеличение значения целе- вой функции за каждые 90 итераций	Общее время счета (мин.)
90	1 498 481	1 498 481	75	540	1 507 022	114	450
180	1 504 541	6 060	150	630	1 507 171	149	525
260	1 505 922	1 381	225	720	1 507 242	71	600
360	1 506 556	634	300	810	1 507 289	47	675
450	1 506 908	352	375	900	1 507 321	32	750

первых итерациях происходило довольно быстрое возрастание значения целевой функции, которое замедлилось на последующих итерациях. Такой эффект наблюдается и при решении других задач линейного программирования с большим числом столбцов. На нем мы остановимся более подробно в следующих параграфах.

Решение задач транспортного типа также может быть осуществлено и с использованием метода учета двухсторонних ограничений (см. главу V).

§ 2.10. ОПТИМАЛЬНОЕ РАСПРЕДЕЛЕНИЕ РЕСУРСОВ

2.10.1. Общая формулировка задачи

Многие задачи распределения ресурсов, включая задачи теории расписаний, могут быть сформулированы следующим образом. Имеется I производственно-технологических способов использования однородного ресурса в течение T периодов времени. Пусть x_{it} — интенсивность использования i -го способа в t -м периоде и введем в рассмотрение вектор

$$x_i = (x_{i1}, \dots, x_{iT}). \quad (1)$$

Множество допустимых значений интенсивностей использования способов определяется как требованиями полного удовлетворения спроса, так и особенностями каждого из производственно-технологических способов. Например, технические возможности оборудования ограничивают объем выпускаемой продукции. Все подобные ограничения будем называть множеством (блоком) технологических ограничений (возможностей) и обозначать через S_i так, что

$$x_i \in S_i. \quad (2)$$

При использовании технологических способов возникают затраты и потребляются ресурсы. Пусть далее $c_i(x_i)$ — затраты при использовании i -го способа с интенсивностью x_i ; $y_{it}(x_i)$ — количество ресурса, расходуемого в t -м периоде при использовании i -го способа с интенсивностью x_i ; b_t — лимит использования ресурса в t -м периоде;

$$y_i(x_i) = (y_{i1}(x_i), \dots, y_{iT}(x_i)), \quad (3)$$

$$b = (b_1, \dots, b_T). \quad (4)$$

Задача оптимального распределения ресурса заключается в определении значений переменных $x_1, x_2, \dots, x_I$, минимизирующих суммарные затраты за весь период времени

$$z = \sum_{i=1}^I c_i(x_i) = \min \quad (5)$$

при соблюдении ограничений на использование ресурса

$$\sum_{i=1}^I y_{it}(x_i) \leq b_t \quad (6)$$

и выполнении технологических условий

$$x_i \in S_i. \quad (7)$$

Ограничение (6) является связующим ограничением, что не позволяет решать задачу для каждого способа в отдельности.

Рассматриваемая задача легко может быть обобщена и на случай наличия нескольких ресурсов. Пример такой задачи будет рассмотрен в конце этого параграфа.

Задача о составлении графика выпуска I изделий также является частным случаем модели (5) — (7). Чтобы убедиться в этом, обозначим через x_{it} — количество изделий i -го вида, произведенное в t -м периоде. Ограниченными ресурсами являются рабочая сила или различные виды оборудования. Пусть $c_i(x_i)$ — функция издержек производства i -го вида продукции, r_{it} — величина спроса на i -ю продукцию в t -м периоде. Условия типа $x_i \in S_i$ соответствуют ограничениям на уровни запасов η_{it} и объемы производства, т. е.

$$S_i = \{x_i / (x_{it})_{\min} \leq x_{it} \leq (x_{it})_{\max}, \quad (\eta_{it})_{\min} \leq \eta_{it} \leq (\eta_{it})_{\max}, \\ t = 1, \dots, T\}, \quad (8)$$

где уровни запасов η_{it} определяются по формуле

$$\eta_{it} = \eta_{i,t-1} + x_{it} - r_{it}, \quad t = 1, \dots, T, \quad \eta_{i0} — \text{задано.} \quad (9)$$

Приближенное решение с помощью методов линейного программирования. Могут возникнуть значительные трудности при решении задачи (5) — (7), если I — достаточно большое число или когда x_{it} принимают дискретные значения (например, в случае возможности варьирования объемами производства только за счет разного количества используемых установок). Манн [9] показал возможность приближенного решения задачи (5) — (7) при $I \gg T$ в результате сведения ее к задаче линейного программирования большой размерности.

Предположим, что множество оптимальных значений x_i конечно. Это предположение обязательно выполняется, если S_i — ограниченное множество и x_{it} — дискретные переменные (другие возможности будут рассмотрены позже). Пусть x_i^j — j -й элемент (план) множества S_i . Введем дополнительные обозначения

$$c_{ij} \equiv c_i(x_i^j), \quad (10)$$

$$y_{ij} = y_i(x_i^j), \quad (11)$$

и рассмотрим следующую задачу целочисленного программирования:

минимизировать

$$\sum_{i,j} c_{ij} \theta_{ij} \quad (12)$$

при ограничениях

$$\sum_i \theta_{ij} = 1, \quad i = 1, \dots, I, \quad (13)$$

$$\sum_{i,j} y_{ij} \theta_{ij} \leq b, \quad (14)$$

$$\theta_{ij} \geq 0, \quad (15)$$

$$\theta_{ij} \text{ — целые.} \quad (16)$$

В соответствии с ограничениями (13), (15) и (16) для каждого i только одна переменная θ_{ij} принимает значение, равное единице (остальные — нули). Переменная $\theta_{ij}=1$ определяет план выпуска i -го вида продукции. При этом должно соблюдаться ограничение на использование ресурса (14), а суммарные затраты на производство продукции (12) должны быть минимальными.

Однако решение целочисленной задачи (12)—(15) чрезвычайно трудоемко из-за большого числа столбцов и ограничений. К счастью, если $I \gg T$, то условие целочисленности переменных (16) может быть опущено и большинство переменных в оптимальном плане линейной задачи (12)—(15) будут целочисленными. Это утверждение основано на следующей теореме:

Теорема. Если $I > T$, то любое базисное допустимое решение линейной задачи (12)—(15) обладает тем свойством, что не менее чем $I - T$ индексам « i » будет соответствовать в точности одна положительная переменная θ_{ij} (отсюда следует, что соответствующие переменные θ_{ij} будут принимать значение, равное единице).

Доказательство. Любое базисное решение имеет по крайней мере $I + T$ положительных переменных. Для выполнения I условий (13) необходимо, чтобы хотя бы одна переменная в этих ограничениях была отличной от нуля.

Каждому из I ограничений (13) должна удовлетворять хотя бы одна положительная переменная. Остальные T базисных переменных будут распределяться между не более чем T ограничениями (13). Поэтому по крайней мере в $(I - T)$ ограничениях будет находиться по одной положительной переменной.

Имеется много практических задач, для которых выполняется условие $I \gg T$. К числу таких относится задача о составлении графика выпуска сотен видов продукции при $T = 12$ (мес.). Однако в подобных ситуациях чрезвычайно сложно решать задачу (12)—(15), так как матрица условий этой задачи будет иметь большое число строк и столбцов. Дзелинский и Гомори [10] рассмотрели вариант использования метода разложения Данцига — Вулфа для решения подобной задачи. При этом возникла

координирующая задача с $T+1$ строками и с большим числом столбцов. Однако трудность, связанная с наличием большого числа столбцов, может быть успешно преодолена с помощью метода генерации столбцов.

Матрица условий задачи (12)–(15) имеет следующую структуру:

$$\begin{array}{ccc}
 \theta_{11} \dots \theta_{1j_1} & \theta_{21} \dots \theta_{2j_2} & \theta_{I1} \dots \theta_{Ij_I} \\
 \boxed{y_{11} \dots y_{1j_1}} & \boxed{y_{21} \dots y_{2j_2}} & \dots \dots \boxed{y_{I1} \dots y_{Ij_I}} \\
 \boxed{1 \quad 1 \dots 1} & & \\
 & \boxed{1 \quad 1 \dots 1} & \dots \dots \boxed{1 \quad 1 \dots 1}
 \end{array}$$

Ограничения вида $\sum_j \theta_{ij} = 1$, очевидно, выступают как ограничения соответствующих подзадач, а условие (14) — общее связывающее ограничение.

Введем обозначения:

$$e_i = (1, 1, \dots, 1), \quad j_i \text{ компонент}, \quad (17)$$

$$\theta_i = (\theta_{i1}, \dots, \theta_{ij_1}), \quad (18)$$

$$c_i = (c_{i1}, \dots, c_{ij_1}), \quad (19)$$

$$\theta = (\theta_1, \dots, \theta_I), \quad (20)$$

$$c = (c_1, \dots, c_I), \quad (21)$$

$$L_i = [y_{i1}, \dots, y_{ij_1}], \quad (22)$$

$$L = [L_1, L_2, \dots, L_I], \quad (23)$$

$$A = \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_I \end{bmatrix}, \quad (24)$$

$$e = (1, 1, \dots, 1), \quad I \text{ компонент}.$$

Используя эти обозначения, перепишем условия задачи (11)–(15) в следующей более компактной форме:

минимизировать

$$c\theta \quad (25)$$

при ограничениях

$$L\theta + s = b, \quad (26)$$

$$A\theta = e, \quad (27)$$

$$\theta \geq 0, s \geq 0. \quad (28)$$

Так как множество всех решений системы (27) — (28) ограничено, то любое из этих решений может быть представлено в виде выпуклой комбинации крайних точек d_q :

$$\theta = \sum_q \lambda_q d_q, \quad \lambda_q \geq 0, \quad \sum_q \lambda_q = 1. \quad (29)$$

Подставив (29) в (25) — (26), получим следующую координирующую задачу:

минимизировать

$$z = \sum_q (cd_q) \lambda_q \quad (30)$$

при условиях

$$\sum_q (Ld_q) \lambda_q + s = b, \quad (31)$$

$$\sum_q \lambda_q = 1, \quad \lambda_q \geq 0, \quad s \geq 0. \quad (32)$$

Эта задача имеет $T+1$ ограничений и $\Pi(j_i)$ столбцов (столбцы соответствуют крайним точкам множества (27) — (28)). Перейдем к формулировке подзадач. Пусть (π_i, π_0) — вектор двойственных оценок ограничений (31) — (32). Характеристическая разность для столбца q определяется по формуле

$$\bar{c}_q = cd_q - \pi Ld_q - \pi_0. \quad (33)$$

Для того чтобы найти $\min \bar{c}_q$, напомним, что d_q — крайняя точка системы ограничений

$$\left. \begin{aligned} e_i \theta_i &= 1, \\ \theta_i &\geq 0, \end{aligned} \right\} i = 1, 2, \dots, I. \quad (34)$$

Если осуществить разбиение d_q

$$d_q = (d_{q1}, \dots, d_{qi}), \quad (35)$$

то легко проверить, что каждая d_{qi} имеет вид

$$d_{qi} = (0, 0, \dots, 0, 1, 0, \dots, 0). \quad (36)$$

Аналогичным образом осуществим разбиение cd_q и Ld_q :

$$cd_q = \sum_i c_i d_{qi}, \quad (37)$$

$$Ld_q = [L_1, \dots, L_I] \begin{bmatrix} d_{q1} \\ \vdots \\ d_{qI} \end{bmatrix} = \sum_i L_i d_{qi}. \quad (38)$$

Найдем характеристические разности

$$\bar{c}_q = \sum_{i=1}^I (c_i - \pi L_i) d_{qi} - \pi_0. \quad (39)$$

Так как (39) — сепарабельная функция, нахождение $\min \bar{c}_q$ эквивалентно нахождению для каждого i минимума выражения

$$(c_i - \pi L_i) d_{qi}. \quad (40)$$

Решение задачи (40), очевидно, заключается в том, что компоненте, для которой достигается $\min_i (c_i - \pi L_i)$, присваивается значение 1. Нахождение этого минимума может быть осуществлено в результате решения подзадачи, учитывая (19), (10) — (11) и (22):

$$\min_i (c_i(x_i^j) - \pi Y_i(x_i^j)). \quad (41)$$

Так как $x_i^j \in S_i$, то решение подзадачи (41) равносильно решению эквивалентной задачи:

минимизировать

$$c_i(x_i) - \pi Y_i(x_i) \quad (42)$$

при ограничениях

$$x_i \in S_i. \quad (43)$$

Отметим, что задача (42) — (43) — однопродуктовая задача планирования производства и запасания, в которой πY_i — «штраф» за использование общего лимитированного ресурса. Число π_i может быть интерпретировано как цена ресурса в периоде t .

Пусть x_i^0 — решение подзадачи i . Из (39) следует

$$\min \bar{c}_q = \sum_{i=1}^I (c_i(x_i^0) - \pi Y_i(x_i^0)) - \pi_0. \quad (44)$$

Определим характеристические разности столбцов дополнительных переменных

$$\bar{c}_t = -\pi_t. \quad (45)$$

Если

$$\min \bar{c}_q < 0, \quad (46)$$

$$\min \bar{c}_q < \min_t c_t, \quad (47)$$

то формируется столбец

$$\begin{bmatrix} Ld_q \\ 1 \end{bmatrix} = \begin{bmatrix} \Sigma Y_t(x_t^0) \\ 1 \end{bmatrix},$$

который вводится в базис. Если условие (47) не выполнено, но $\min_t \bar{c}_t < 0$, то в базис вводится столбец дополнительных переменных. Если же оценки всех столбцов неотрицательны, то текущий базис является оптимальным.

2.10.2. Модификация модели: определение размера партии выпуска и распределение рабочей силы

В своей работе [10] Дзелинский и Гомори показали, что при некоторых предположениях относительно производственного процесса и функции издержек подзадача (42)—(43) принимает специальный вид. Предположим, что выпуск продукции осуществляется партиями. Переход от выпуска партии одного изделия к партии другого изделия связан с затратами на переналадку оборудования (такая ситуация характерна, например, для механических цехов металлообрабатывающего предприятия с большой номенклатурой выпускаемых изделий). Общим ограниченным ресурсом является рабочая сила. Применяются различные формы оплаты труда. Количество рабочей силы можно регулировать. При этом возникают затраты, связанные как с наймом, так и с увольнением рабочих. Так как количеством рабочей силы можно варьировать, то необходимо одновременно определить общую потребность в рабочей силе и ее распределение (загрузку). Предполагается далее, что в каждый промежуток планового периода спрос удовлетворяется полностью, а начальный и конечный запасы продукции равны нулю. Отсутствует ограничение сверху на выпуск продукции.

Таким образом, ограничения $x_i \in S_i$ имеют следующий вид:

$$\eta_{it} = \sum_{\tau=1}^t (x_{i\tau} - r_{i\tau}) \geq 0 \quad (t = 1, \dots, T-1), \quad (1)$$

$$\eta_{iT} = \sum_{\tau=1}^T (x_{i\tau} - r_{i\tau}) = 0, \quad (2)$$

$$x_{it} \geq 0 \quad \text{для всех } i, t. \quad (3)$$

Функции издержек производства $c_i(x_i)$ принимаются линейными. Их параметры соответствуют изменяющимся во времени затратам на единицу различных видов продукции

$$c_i(x_i) = \sum_{t=1}^T c_{it} x_{it}. \quad (4)$$

Вектор потребления ресурсов $Y_i(x_i)$ соответствует затратам рабочей силы (в часах) при плане x_i . Рабочая сила различается по квалификационным группам ($k=1, \dots, K$). Как отмечалось, используются различные формы оплаты труда ($r=1, \dots, R$): основное время ($r=1$), сверхурочное время ($r=2$) и т. д.

Затраты рабочего времени складываются из непосредственного времени на производство продукции и из затрат на наладку оборудования (когда $x_{it} \geq 0$). Пусть a_{ik} — количество рабочего времени, необходимого на наладку оборудования для производства i -го изделия при использовании рабочей силы k -й группы; b_{ik} — трудоемкость изготовления единицы i -го изделия при использовании рабочих k -й квалификационной группы. Тогда общие затраты рабочей силы k -й группы для производства продукции в объеме x_{it} составят

$$y_{ikt}(x_{it}) = a_{ik} \delta(x_{it}) + b_{ik} x_{it}, \quad (5)$$

где

$$\delta(x) = \begin{cases} 1, & x > 0, \\ 0 & \text{во всех остальных случаях.} \end{cases} \quad (6)$$

Пусть h_k^r — фонд времени одного рабочего k -й группы при использовании r -й формы оплаты труда (применяются повышенные расценки за работу, выполненную в сверхурочное время); w_{kt}^r — общее количество рабочих k -й группы, переведенных на r -ю форму оплаты труда и используемых в t -м подпериоде (w_{kt}^r — переменная величина). Ограничение на использование рабочей силы можно записать таким

образом:

$$\sum_i y_{ikt}(x_{it}) \leq \sum_r h_k^r w_{kt}^r \quad (t = 1, \dots, T; k = 1, \dots, K). \quad (7)$$

Если теперь ввести обозначения

$$Y_{ik}(x_i) = (y_{ik1}, y_{ik2}, \dots, y_{ikT}), \quad (8)$$

$$b_k = \left(\sum_r h_k^r w_{k1}^r, \dots, \sum_r h_k^r w_{kT}^r \right), \quad (9)$$

то ограничение (7) примет вид

$$\sum_i Y_{ik}(x_i) \leq b_k \quad (k = 1, \dots, K). \quad (10)$$

Очевидно, что эта задача является обобщением задачи (2.10.1-5) — (2.10.1-7) (по сравнению с последней в этой задаче имеется не один, а K видов различных ресурсов). В задаче необходимо также учесть изменения во времени в использовании рабочей силы. Поэтому введем дополнительные обозначения: w_{kt}^+ — увеличение рабочей силы k -й группы в подпериоде t ; w_{kt}^- — уменьшение рабочей силы k -й группы в подпериоде t . Можно записать:

$$\sum_r w_{kt}^r = \sum_r w_{kt-1}^r + w_{kt}^+ - w_{kt}^- \quad (w_{k0}^r — задано). \quad (11)$$

Понятно, что затраты, связанные с колебаниями в использовании рабочей силы, должны быть учтены в целевой функции. Поэтому последняя может быть записана в виде

$$z = \sum_i c_i(x_i) + \sum_{k,t,r} c_{kt}^r w_{kt}^r + \sum_{k,t} (c_{kt}^+ w_{kt}^+ + c_{kt}^- w_{kt}^-), \quad (12)$$

где c_{kt}^r — заработная плата одного рабочего (k, r) в подпериоде t ; c_{kt}^+ , c_{kt}^- — затраты, связанные с увеличением и уменьшением рабочей силы k -й группы в t -м подпериоде (в расчете на одного рабочего).

Общая задача составления графика выпуска продукции заключается теперь в нахождении значений переменных $x_1, \dots, x_i$, доставляющих минимум целевой функции (12) и удовлетворяющих ограничениям (10), (11) и (1) — (3). Множество S_i , определяемое ограничениями (1) — (3), не является конечным. Покажем, однако, что только конечное число элементов этого множества будет присутствовать в оптимальном плане. Но сначала запишем аппроксимирующую линейную задачу вида (2.10.1-12) — (2.10.1-15), используя некоторое конечное, но про-

извольное множество программ $x_i^j \in S_i$. Пусть, как и прежде,

$$c_{ij} \equiv c_i(x_i^j), \quad Y_{ik}(x_i^j) \equiv Y_{ijk}. \quad (13)$$

Тогда аппроксимирующую линейную задачу представим в виде: минимизировать

$$\sum_{i,j} c_{ij} \theta_{ij} + \sum_{k,t,r} c_{kt}^r \omega_{kt}^r + \sum_{k,t} (c_{kt}^+ \omega_{kt}^+ + c_{kt}^- \omega_{kt}^-) \quad (14)$$

при ограничениях

$$\sum_{i,j} Y_{ijk} \theta_{ij} \leq b_k \quad (k = 1, \dots, K), \quad (15)$$

$$\sum_r \omega_{kt}^r = \sum_r \omega_{kt-1}^r + \omega_{kt}^+ - \omega_{kt}^- \quad (k = 1, \dots, K, t = 1, \dots, T), \quad (16)$$

$$\sum_i \theta_{ij} = 1 \quad (i = 1, \dots, I). \quad (17)$$

Кроме того, все переменные должны быть неотрицательными. Как мы показали уже, если $I > 2KT$ (число ограничений (15) — (16)), то в любом базисном решении по крайней мере для $I - 2KT$ изделий будет выделяться только одна определенная программа выпуска. Перейдем к реализации метода декомпозиции для решения рассматриваемой задачи, существенно используя материал п. 2.10.1. Пусть

$$L_{ik} = [Y_{i1k}, \dots, Y_{iIk}], \quad (18)$$

$$L_k = [L_{1k}, \dots, L_{Ik}]. \quad (19)$$

Если опять d_q — крайняя точка множества ограничений (2.10.1-34), то координирующая задача примет такой вид: минимизировать

$$\sum_q (cd_q) \lambda_q + \sum_{k,t,r} c_{kt}^r \omega_{kt}^r + \sum_{k,t} (c_{kt}^+ \omega_{kt}^+ + c_{kt}^- \omega_{kt}^-) \quad (20)$$

при ограничениях

$$\sum_q (L_k d_q) \lambda_q \leq b_k \quad (k = 1, \dots, K), \quad (21)$$

$$\sum_q \lambda_q = 1, \quad (22)$$

$$\sum_r \omega_{kt}^r = \sum_r \omega_{kt-1}^r + \omega_{kt}^+ - \omega_{kt}^- \quad \text{для всех } (k, t) \quad (23)$$

(все переменные отрицательные). Напомним, что компоненты вектора b_k определяются значениями переменных w_{ki}^+ (см. (9)). В соответствии с процедурой метода декомпозиции характеристические разности должны быть определены для всех столбцов, соответствующих переменным w_{ki}^+ , w_{ki}^- , w_{ki}^0 и дополнительным переменным.

Особый интерес представляют столбцы, соответствующие переменным λ_q . Пусть B — базисная матрица координирующей задачи, π_k — вектор оценок ограничений на использование k -го вида ресурсов (21) и π_q — оценка ограничения (22). Найдем характеристическую разность столбца, соответствующего переменной λ_q

$$\bar{c}_q = cd_q - \sum_k \pi_k L_{kq} d_q - \pi_q. \quad (24)$$

Как и в п. 2.10.1, представим d_q в виде

$$d_q = (d_{q1}, \dots, d_{qi}) \quad (d_{qi} \text{ — единичный вектор}). \quad (25)$$

Поэтому (24) примет вид

$$\bar{c}_q = \sum_i \left(c_i - \sum_k \pi_k L_{ki} \right) d_{qi}. \quad (26)$$

Для минимизации $\bar{c}_q$ необходимо минимизировать каждое выражение

$$\left(c_i - \sum_k \pi_k L_{ki} \right) d_{qi}. \quad (27)$$

Так как d_{qi} — единичный вектор, то операцию нахождения минимума выражения (27) можно представить в виде

$$\min \left\{ c_i(x_i') - \sum_k \pi_k Y_{ik}(x_i') \right\}. \quad (28)$$

Учитывая, что $x_i \in S_i$, это эквивалентно

$$\min \left[\sum_i c_{ii} x_{ii} - \sum_{k,i} \pi_{ki} (a_{ik} \delta(x_{ii}) + b_{ik} x_{ii}) \right] \quad (29)$$

или, в другой форме,

$$\min \left[\sum_i \left\{ c_{ii} - \sum_k \pi_{ki} b_{ik} \right\} x_{ii} - \sum_i \left\{ \sum_k \pi_{ki} a_{ik} \right\} \delta(x_i) \right] \quad (30)$$

при ограничениях (1) — (3) для каждого издания. В случае, когда может быть гарантирована неположительность оценок π_{ki} , эта задача является однопродуктовой задачей управления запасами Вагнера — Уайтина [11], которая может быть эффек-

тивно решена методом динамического программирования. Для того чтобы гарантировать выполнение условия $\pi_{ki} \leq 0$, отметим, что характеристическая разность, соответствующая дополнительной переменной в (21), равна $\bar{c}_{ki} = -\pi_{ki}$, а симплексный критерий обеспечивает, что ни одна из дополнительных переменных не может войти в базис с уменьшением затрат, если

$$\pi_{ki} \leq 0. \quad (31)$$

Следовательно, для того, чтобы обеспечить выполнение условия (31), необходимо просто провести симплексную процедуру вплоть до момента, когда в базис войдут все дополнительные переменные, введение которых способствует снижению затрат, после чего (31) окажется выполненным и подзадачи (30) далее могут решаться как задачи Вагнера — Уайтина. Заметим лишь, что прежде чем должно приниматься окончательное решение о вводе в базис, следует вычислить и характеристические разности для столбцов, соответствующих переменным w_{kt}^+ , w_{kt}^- . Конечность множества векторов, из которых осуществляется выбор, обеспечивается в рассматриваемом случае тем, что подзадача (30) является задачей специального вида. Вагнер и Уайтин доказали, что оптимальное решение задачи обладает свойством, что из $x_{ii} > 0$ следует $\eta_{ii-1} = 0$. Таким образом, оптимальные объемы выпуска продукции в задаче являются частичными суммами объемов спроса r_{ii} , совокупность которых есть конечное множество.

2.10.3. Результаты вычислительных экспериментов

Дзелинским и Гомори было решено большое количество задач на составление графиков выпуска продукции с использованием метода декомпозиции. Координирующая задача имела вид, аналогичный задаче (20) — (23), с дополнительными ограничениями на количество рабочей силы, используемой на каждом из R вариантов использования. Для решения задачи применялся вариант модифицированного симплекс-метода с мультипликативным представлением обратной матрицы. Первое допустимое решение находилось с использованием дополнительных и искусственных переменных. Результаты решения 10 задач представлены в таблице 2.10.1.

В разделе I этой таблицы сумма (5) + (6) определяет количество столбцов, соответствующих переменным w_{kt}^+ , w_{kt}^- , w_{kt}^0 и дополнительным переменным, а (7) — количество строк, соответствующих ограничениям, кроме ограничения вида $\sum_i \theta_{ij} = 1$. Раздел II таблицы свидетельствует о том, что решение

Таблица 2.10.1

I. Задачи

	35	963	428	428	428	428	100	100	100	100
1. I — число изделий										
2. K : количество видов оборудования	2	2	2	2	2	2	1	1	1	1
3. T : число периодов	3	3	3	5	7	8	12	12	12	12
4. R : число вариантов	2	2	2	2	2	2	2	2	2	2
5. $(2KT)(R+1)$ — число рабочих столбов	36	36	36	60	84	96	72	72	72	72
6. $(KT)^3$ — число дополнительных столбов	18	18	18	30	42	48	36	36	36	36
7. $(R+2)(T)+2$ — количество общих связывающих ограничений	26	26	26	42	58	66	50	50	50	50
8. $(2T-1)I$ — количество планов выпуска изделий	140	3847	1712	6784	27392	54434	204800	204800	204800	204800

II. Размерность задач как задач линейного программирования

9. $(1) + (7)$ — число строк	61	989	454	466	486	494	150	150	150	150
10. $(5) + (6) + (8)$ — число столбов	194	3901	1766	6874	27518	54928	204908	204908	204908	204908

Таблица 2.10.1 (продолжение)

III. Размерность задач, возникающая при использовании методов декомпозиции и динамического программирования

11. (7) — число строк	27	27	27	43	59	67	51	51	51	51
12. (5)+(6) — число столбцов	54	54	54	90	126	144	108	108	108	108
13. (1) — (7(7+1)/2) — число шагов при использовании динамического программирования	210	5778	2568	6420	11 984	15 408	1800	7800	7800	7800

IV. Вычисления при использовании методов декомпозиции и динамического программирования

14. Общее время на получение оптимального решения (мин.)	3,36 ^a .b	24,5 ^a .b	5,66 ^a	9,0 ^a	10,70 ^a	17,92 ^a	5,90 ^a	8,87 ^a	8,87 ^a	9,00 ^a
15. Общее количество итераций для выполнения этапа I	48	34	22	35	52	65	45	49	46	51
16. Среднее время для нахождения одного плана	0,07	0,72 ^b	0,264	0,345	0,483	0,510	0,183	0,225	0,223	0,215
17. Общее число найденных планов	16	12	19	24	20	29	26	34	31	36

Индексы: ^a — без учета времени на ввод и вывод информации. ^b использовался вариант динамического программирования, который был затем улучшен. В результате время счета было сокращено на одну треть.

задач 1—10, и прежде всего задач 7—10, стандартными методами линейного программирования весьма затруднительно (или вообще невозможно) из-за большого числа столбцов в матрице условий этих задач. В этом случае для хранения информации потребовался бы огромный объем памяти, а на считывание данных с магнитного диска (или ленты) на каждой

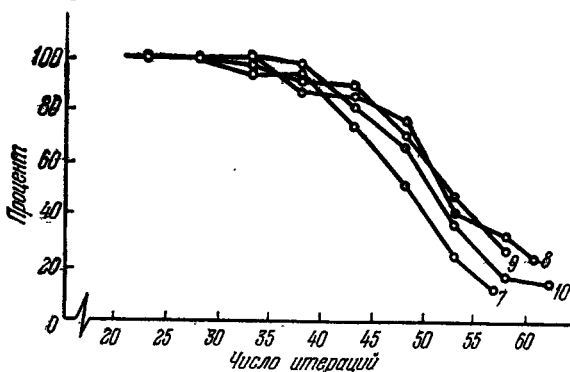


Рис. 2.2. Процент изделий.

итерации симплекс-метода ушло бы большое количество времени. Использование же метода декомпозиции или других методов, основанных на идее генерации столбцов, является более предпочтительным. При использовании этих методов число итераций в основной фазе симплексного метода сравнительно невелико и примерно равно числу ограничений задачи. Это почти наилучшая оценка по сравнению с оценкой, обычно определяемой на основе использования эмпирического правила (от m до $3m$ итерации). Учитывая опыт решения задачи о рубке леса (см. § 2.9), такой результат можно считать обнадеживающим.

Можно ожидать, что вследствие наличия большого числа столбцов в координирующей задаче оптимальному плану будет предшествовать большое число не оптимальных планов. Это наблюдение представляется очень важным по следующей причине. Использование метода декомпозиции для рассматриваемого класса задач имеет смысл только в том случае, если большинство переменных в оптимальном плане принимает целочисленные значения. Этим свойством обладают все базисные решения линейной задачи (если $I > 2KT$). Но неоптимальные базисные решения координирующей задачи уже этому условию не удовлетворяют. Поэтому любое промежуточное решение координирующей задачи может иметь произвольное число нецелочисленных переменных. На рис. 2.2 для условий четырех последних

задач табл. 2.10.1 представлена зависимость общего числа нецелочисленных переменных (в % %) от количества итераций.

Рисунок 2.2 позволяет сделать вывод, что с увеличением числа итераций количество нецелочисленных переменных довольно быстро убывает, что позволяет для некоторых задач не доводить всех вычислений до конца.

Другой возможный подход к решению задачи (2.10.1-12) — (2.10.1-15) будет рассмотрен в § 3.2. Возможность непосредственного решения задачи (2.10.1-5) — (2.10.1-7) обсуждается также и в § 7.8.3.

§ 2.11. ПРИМЕНЕНИЕ МЕТОДА ОДНОВРЕМЕННОГО РЕШЕНИЯ ПРЯМОЙ И ДВОЙСТВЕННОЙ ЗАДАЧ К КООРДИНИРУЮЩЕЙ ЗАДАЧЕ

Метод одновременного решения прямой и двойственной задач [12] является вариантом симплексного метода. Он оказался особенно эффективным для решения задач специального вида и, прежде всего, для решения транспортных задач. Сейчас наша цель состоит в применении этого метода к координирующей задаче при использовании принципа разложения. Как будет показано ниже, реализация такой идеи ведет к появлению подзадач с дробно-линейным функционалом с ограничениями в форме линейных неравенств (или равенств).

Вначале остановимся на самостоятельной проблеме решения задач дробно-линейного программирования, а затем рассмотрим новый подход к координирующей задаче.

2.11.1. Дробно-линейное программирование

Задачу дробно-линейного программирования можно представить в таком виде:

минимизировать

$$\frac{px + a}{qx + \beta} \quad (1)$$

при ограничениях

$$a_i x \leq b_i \quad (i = 1, \dots, m). \quad (2)$$

Эта задача представляет интерес в силу ряда причин [13]—[14]. Рассматривается она здесь потому, что эта задача возникает, как будет показано ниже, и при применении метода одновременного решения прямой и двойственной задачи к решению координирующей задачи.

Покажем, что задача (1)—(2) может быть решена с помощью симплексного метода. Дробно-линейная функция не является ни вогнутой, ни выпуклой. Однако поверхности уровня этой

функции, т. е. множества

$$S_b = \{x \mid px + a/qx + \beta = b\} \quad (3)$$

являются гиперплоскостями. Дорн [13] доказал, что любой локальный минимум задачи дробно-линейного программирования является в то же время глобальным, и если конечное оптимальное решение достигается в конечной точке, существует крайняя точка, которая является оптимальной. Пусть

$$S = \{x \mid a_i x \leq b_i, \quad i = 1, \dots, m\}. \quad (4)$$

Сформулированное выше утверждение выполняется, если числитель и знаменатель (1) не обращаются одновременно в нуль для любого $x \in S$. Во всех задачах, которые будут рассмотрены в следующем параграфе, имеется требование неотрицательности знаменателя (1). Так как такое условие значительно упрощает все последующие доказательства, будем предполагать во всех необходимых случаях, что оно выполняется.

Предположение 1.

$$qx + \beta > 0 \quad \text{для любого } x \in S.$$

Упростим выражение (1). Для этого увеличим число компонент вектора x на единицу

$$x = (x_1, \dots, x_n, x_{n+1}) \quad (5)$$

и введем дополнительное ограничение

$$x_{n+1} \geq 1, \quad x_{n+1} \leq 1. \quad (6)$$

Теперь функцию (1) можно записать в таком виде *):

$$f(x) = \frac{px}{qx}. \quad (7)$$

Определение 1. Непрерывная функция $f(x)$ является квазивыпуклой функцией на выпуклом множестве S , если выполняется любое из следующих эквивалентных условий:

$$(a) \quad \{x \mid f(x) \leq c, \quad x \in S\} \text{ — выпукло для всех } c, \quad (8)$$

$$(b) \quad x_1, x_2 \in S, \quad f(x_2) < f(x_1) \Rightarrow f(\lambda x_2 + (1 - \lambda)x_1) \leq f(x_1), \quad 0 \leq \lambda \leq 1. \quad (9)$$

Определение 2. Функция $f(x)$ является строго квазивыпуклой на множестве S , если условие (9) выполняется в виде строгого неравенства.

*) Для упрощения записи здесь сохранены прежние обозначения p, q для расширенных векторов (p, α) , (q, β) . (Прим. ред.)

Теорема 1 (Понстейн [17]). *Любой локальный минимум строго квазивыпуклой функции является глобальным максимумом.*

(Читатель может найти доказательство теоремы в работе [17].) Опровергающие примеры можно привести для случая, когда не учитывается требование *строгой* квазивыпуклости функции.

Значение этой теоремы для целей нашего исследования состоит в том, что при предположении 1 $f(x) = px/qx$ — строго квазивыпуклая функция на множестве S , определяемом выражением (4)*, что немедленно приводит к следующему утверждению.

Теорема 2. *Если выполнены условия предположения 1, то любой локальный минимум задачи дробно-линейного программирования (1) — (2) является глобальным.*

Чтобы доказать эту теорему, введем определение.

Определение 3. Функция $f(x)$ является квазивогнутой на выпуклом множестве S , если функция $-f(x)$ квазивыпукла.

Так как в соответствии с предположением 1 множество $\{x | -px/qx \leq c, x \in S\}$ выпукло для всех значений c , то $f(x) = px/qx$ является квазивогнутой функцией на множестве S , заданном согласно (4). Заметив f на $-f$, перепишем условие (9) для непрерывных квазивогнутых функций в виде

$$f(x_2) > f(x_1) \Rightarrow f(\lambda x_1 + (1 - \lambda)x_2) > f(x_1) \quad (10)$$

или

$$f(\lambda x_1 + (1 - \lambda)x_2) \geq \min[f(x_1), f(x_2)] \quad (11)$$

для любых $x_1, x_2 \in S$ и $0 < \lambda \leq 1$.

В дальнейшем (11) будет использовано нами в качестве определения квазивогнутости функций. Так как в следующем параграфе при рассмотрении блочной задачи со связующими ограничениями предполагается, что ограничения подзадачи образуют ограниченное множество, сформулируем и докажем для этого случая теорему.

Теорема 3. *Предположим, что множество S ограничено и выполнено условие предположения 1. Тогда $f(x) = px/qx$ достигает минимального значения на множестве S в крайней точке.*

Доказательство. Пусть $x^i, i=1, \dots, s$ — крайние точки множества S . Тогда в соответствии с теоремой 2.2.-2 любая точка $x \in S$ может быть представлена в виде

$$x = \sum_{i=1}^s \alpha_i x^i, \quad \sum_{i=1}^s \alpha_i = 1, \quad \alpha_i \geq 0. \quad (12)$$

*) В упражнении 12 это утверждение требуется доказать.

Так как f — квазивогнутая функция, то из (11) следует

$$f(\alpha x^1 + (1 - \alpha)x^2) \geq \min[f(x^1), f(x^2)], \quad (13)$$

$$0 \leq \alpha \leq 1.$$

Пусть

$$x_3 = \alpha x^1 + (1 - \alpha)x^2. \quad (14)$$

Тогда, снова используя (11), получим

$$f(x_3 + (1 - \beta)x^3) \geq \min[f(x_3), f(x^3)]. \quad (15)$$

Так как из условия (13) следует

$$f(x_3) \geq \min[f(x^1), f(x^2)],$$

то

$$\min[f(x_3), f(x^3)] \geq \min[f(x^1), f(x^2), f(x^3)]. \quad (16)$$

Можно записать также

$$\beta x_3 + (1 - \beta)x^3 = \beta \alpha x^1 + \beta(1 - \alpha)x^2 + (1 - \beta)x^3. \quad (17)$$

Коэффициенты этой линейной комбинации неотрицательны и их сумма равна единице. Таким образом, когда коэффициенты α и β принимают значения, заключенные между нулем и единицей, $\beta x_3 + (1 - \beta)x^3$ принимает все значения линейной формы $\alpha_1 x^1 + \alpha_2 x^2 + \alpha_3 x^3$, где α_i — неотрицательные числа, сумма которых равна единице.

Учитывая этот факт и принимая во внимание (16), можно записать (15) в таком виде:

$$f(\alpha_1 x^1 + \alpha_2 x^2 + \alpha_3 x^3) \geq \min[f(x^1), f(x^2), f(x^3)]. \quad (18)$$

Повторение описанной процедуры дает

$$f(x) = f\left\{\sum_{i=1}^s \alpha_i x^i\right\} \geq \min\{f(x^i)\}, \quad (19)$$

что и доказывает утверждение теоремы.

Такое доказательство остается верным для любой квазивогнутой функции. Мартос [18] получил следующий результат. Пусть L — многогранное подмножество ограниченного выпуклого многогранника S . Тогда справедливо утверждение: $f(x)$ достигает своего экстремального значения в крайней точке L тогда и только тогда, когда $f(x)$ — квазивогнутая функция на множестве S .

Используя теорему 3, можно доказать, что если задача дробно-линейного программирования с положительным знаменателем (1) имеет неограниченное решение, значение целевой функции стремится к $-\infty$ вдоль экстремального луча множества S . Это утверждение делает соответствие задач линейного и дробно-линейного программирования совершенно полным.

Методы решения задачи дробно-линейного программирования. Существует большое число таких методов (см. [13] в гл. 1 и [13—15] этой главы). Для нас наибольший интерес представляют алгоритмы Чарнса и Купера [15], Гилмора и Гомори [14]. Для использования метода Чарнса — Купера необходимо только, чтобы множество ограничений задачи было бы ограниченным. Этот метод обеспечивает получение оптимального плана при условии положительности или отрицательности значения знаменателя (1). Так как одно из этих условий присутствует в задаче, которая будет рассмотрена в следующем параграфе, этот алгоритм может быть использован для ее решения.

Что же касается метода Гилмора — Гомори, то он является непосредственным вариантом симплекс-метода. Для его использования не требуется большой переделки стандартных программ. Использование метода этих авторов предполагает, что соответствующая задача дробно-линейного программирования обладает рядом существенных свойств линейной задачи. Как уже было показано, это выполняется лишь при определенных условиях: знак знаменателя не должен меняться на множестве ограничений. В противном случае решение задачи может быть неограниченным, причем оно будет достигаться не в крайней точке и не при движении по экстремальному лучу.

Алгоритм Гилмора — Гомори с использованием метода генерации столбцов может быть применен и для задачи дробно-линейного программирования с большим числом столбцов (см. § 3.1).

Алгоритм Чарнса — Купера. Представим задачу дробно-линейного программирования в таком виде: минимизировать

$$\frac{px + a}{qx + \beta} \quad (20)$$

при ограничениях

$$Ax \leq b, \quad (21)$$

$$x > 0, \quad (22)$$

где множество

$$S = \{x | Ax \leq b, x > 0\} \quad (23)$$

предполагается непустым и ограниченным. Выполним замену

переменных

$$y = tx, \quad t \geq 0 \quad (24)$$

и рассмотрим следующую задачу линейного программирования:
минимизировать

$$py + \alpha t \quad (25)$$

при ограничениях

$$Ay + bt \leq 0, \quad (26)$$

$$qy + \beta t = \gamma, \quad \gamma \neq 0, \quad (27)$$

$$y \geq 0, \quad t \geq 0. \quad (28)$$

Эта задача, как нетрудно видеть, получена из исходной задачи в результате умножения ограничений из числителя (20) на t и приравнивания знаменателя (20) некоторому числу γ . Задача обладает следующим свойством. Для любой пары (y, t) , удовлетворяющей условиям (26)–(28), $t \geq 0$. В самом деле, если это не так, то существует некоторый вектор y , удовлетворяющий условию $Ay \leq 0$, $y \geq 0$. Из (27) следует, что y — ненулевой вектор. Тогда, если x удовлетворяет условиям (21)–(22), $x + \alpha y$ также удовлетворяет этим же условиям для всех $\alpha \geq 0$, что противоречит предположению об ограниченности множества S .

Следующая теорема характеризует соотношение между задачами линейного и дробно-линейного программирования.

Теорема 4. Если (а) γ и $qx^0 + \beta$ положительны для некоторого оптимального решения x^0 задачи (20)–(22), (б) (y^*, t^*) — решение задачи линейного программирования (25)–(28), то y^*/t^* является решением задачи дробно-линейного программирования (20)–(22).

Доказательство. Очевидно, что y^*/t^* — допустимое решение задачи дробно-линейного программирования. Предположим, что y^*/t^* не является оптимальным решением. Тогда x^0 соответствует меньшее значение целевой функции

$$\frac{px^0 + \alpha}{qx^0 + \beta} < \frac{p(y^*/t^*) + \alpha}{q(y^*/t^*) + \beta}. \quad (29)$$

Так как γ и $qx^0 + \beta$ имеют одинаковый знак, то найдется θ такое, что

$$qx^0 + \beta = \theta \gamma. \quad (30)$$

Рассмотрим пару:

$$\hat{y} = \frac{x^0}{\theta}, \quad \hat{t} = \frac{1}{\theta}. \quad (31)$$

Нетрудно проверить, что $(\hat{y}, \hat{t})$ — допустимое решение задачи (25) — (28). Разделив теперь числитель и знаменатель левой части неравенства (29) на θ , получим

$$\frac{p\hat{y} + \alpha t}{-\gamma} < \frac{py^* + \alpha t^*}{\gamma}, \quad (32)$$

что противоречит утверждению об оптимальности плана (y^*, t^*) . Таким образом, теорема доказана. Эта теорема дает возможность получить решение задачи дробно-линейного программирования в результате решения линейной задачи (25) — (28) при условии положительности в точке оптимума знаменателя (20). Если знаменатель отрицателен, то, изменив знаки (p, α) и (q, β) на обратный (умножением на -1), мы опять обеспечим условия применимости теоремы.

Таким образом, нахождение конечного минимума задачи дробно-линейного программирования сводится к решению следующей пары задач:

Положительный знаменатель	Отрицательный знаменатель
Минимизировать $py + \alpha t$ при условиях $Ay - bt \leq 0,$ $qy + \beta t = 1,$ $y \geq 0, t \geq 0.$	Минимизировать $-py - \alpha t$ при условиях $Ay - bt \leq 0,$ $-qy - \beta t = 1,$ $y \geq 0, t \geq 0.$

Понятно, что если известен знак знаменателя в точке оптимума, то решать необходимо только одну из этих задач.

Использование симплекс-метода. Опять будем предполагать, что знаменатель отношения (20) принимает положительные значения на множестве ограничений S . Тогда, как уже было показано выше, любой конечный минимум дробно-линейной функции достигается в крайней точке множества S и неограниченное решение может быть получено при движении вдоль крайнего луча. Оптимальное решение может быть найдено в результате перехода от одной крайней точки к другой, каждый раз выбирая крайнюю точку, которой соответствует меньшее значение целевой функции.

Покажем, как может быть реализован такой процесс.

Теорема 5. Пусть $f(x) = z_1(x)/z_2(x)$ — отношение двух линейных функций, определенных на множестве S , $y \in S$ и пусть s — заданное направление. Предположим, что

$$(a) \quad z_2(x) > 0 \quad \text{для всех } x \in S;$$

(b) $y + \alpha s \in S$ для некоторой области $0 \leq \alpha \leq \delta$, $\delta > 0$;

$$(c) \quad g(0) = \left. \frac{df(y + \alpha s)}{d\alpha} \right|_{\alpha=0} < 0.$$

Тогда $h(\alpha) = f(y + \alpha s)$ — монотонно убывающая функция для всех $0 < \alpha \leq \delta$.

Доказательство. Из (a) и (b) следует, что $h(\alpha)$ — непрерывная и дифференцируемая функция на интервале $[0, \delta]$. Таким образом, теорема будет доказана, если мы сможем показать, что

$$g(\alpha) = \frac{dh}{d\alpha} < 0 \quad \text{для } \alpha \in [0, \delta].$$

Дифференцируя, получим

$$\begin{aligned} g(\alpha) &= \frac{d}{d\alpha} \frac{z_1(y + \alpha s)}{z_2(y + \alpha s)} = \frac{z_1(y + \alpha s) z_2'(s) - z_1'(y + \alpha s) z_2(s)}{z_2^2(y + \alpha s)} = \\ &= \frac{(z_1(y) - \alpha z_1'(s)) z_2'(s) - (z_1'(y) + \alpha z_1''(s)) z_2(s)}{z_2^2(y + \alpha s)} = \\ &= \frac{z_1(y) z_2'(s) - z_1'(y) z_2(s)}{z_2^2(y + \alpha s)}. \end{aligned} \quad (33)$$

Легко показать, что

$$g(\alpha) = \frac{z_2^2(y)}{z_2^2(y + \alpha s)} g(0). \quad (34)$$

Так как ни $z_2(y)$, ни $z_2(y + \alpha s)$ не обращаются в нуль и $g(0) < 0$, то $g(\alpha) < 0$ для всех $\alpha \in [0, \delta]$.

Теперь ясно, что если $g(0) > 0$, то $h(\alpha)$ возрастает. Таким образом, если производная дробно-линейной функции по направлению вдоль ребра многогранника, идущего к новой крайней точке, является отрицательной, то значение целевой функции в этой точке будет меньше по сравнению со значением целевой функции в текущей точке. Если в некоторой крайней точке значения всех таких производных по направлению неотрицательны, то текущая точка является оптимальной. Остается только указать достаточно эффективный способ вычисления производных по направлению вдоль всех ребер.

Предположим, что условия задачи представлены в канонической форме и B — текущий допустимый базис. Введем два

дополнительных уравнения:

$$\sum_j c_{1j}x_j - z_1 = 0, \quad (35)$$

$$\sum_j c_{2j}x_j - z_2 = 0, \quad (36)$$

где $-z_1$ и $-z_2$ переменные, которые неизменно остаются в базисе. Если исключить базисные переменные x_j из уравнений (35) и (36), то система ограничений переписывается в таком виде:

$$\left. \begin{array}{l} x_1 \dots + \bar{a}_{1m+1}x_{m+1} + \dots + \bar{a}_{1n}x_n = b_1, \\ \vdots \\ x_m + \bar{a}_{m,m+1}x_{m+1} + \dots + \bar{a}_{mn}x_n = b_m, \\ -z_1 + \bar{c}_{1,m+1}x_{m+1} + \dots + \bar{c}_{1n}x_n = -\bar{z}_1, \\ -z_2 + \bar{c}_{2,m+1}x_{m+1} + \dots + \bar{c}_{2n}x_n = -\bar{z}_2. \end{array} \right\} \quad (37)$$

Для того чтобы определить переменную, вводимую в базис, необходимо найти производные по направлению

$$\bar{c}_j = \frac{\partial (z_1/z_2)}{\partial x_j} = \frac{\bar{z}_2 \partial z_1 / \partial x_j - \bar{z}_1 \partial z_2 / \partial x_j}{\bar{z}_2^2} \quad (j = m+1, \dots, n), \quad (38)$$

$$c_j = \frac{\partial (z_1/z_2)}{\partial x_j} = \frac{\bar{z}_2 \bar{c}_{1j} - \bar{z}_1 \bar{c}_{2j}}{\bar{z}_2^2}. \quad (39)$$

Из (39) следует, что они легко вычисляются, если известны характеристические разности $\bar{c}_{1j}$ и $\bar{c}_{2j}$. Последние в свою очередь могут быть получены в результате применения модифицированного симплексного метода с использованием информации исходной задачи и соответствующих двойственных переменных. Пусть $\hat{B}$ — матрица текущего базиса (размерности $(m+2) \times (m+2)$):

$$\hat{B} = \left[\begin{array}{c|cc} B & 0 & 0 \\ \hline c_{1B} & 1 & 0 \\ \hline c_{2B} & 0 & 1 \end{array} \right], \quad (40)$$

где c_{1B} и c_{2B} — векторы коэффициентов при базисных переменных в выражениях для z_1 и z_2 соответственно. Легко проверить, что обратная матрица к матрице $\hat{B}$ имеет вид

$$\hat{B}^{-1} = \left[\begin{array}{c|cc} B^{-1} & 0 & 0 \\ \hline -c_{1B}B^{-1} & 1 & 0 \\ \hline -c_{2B}B^{-1} & 0 & 1 \end{array} \right]. \quad (41)$$

Пусть

$$\pi_1 = c_{1B}B^{-1}, \quad \pi_2 = c_{2B}B^{-1}$$

— двойственные переменные текущего базиса, соответствующие z_1 и z_2 (см. (35) и (36)). Теперь небазисные столбцы в (37) можно представить в виде

$$\begin{bmatrix} \bar{p}_j \\ \bar{c}_{1j} \\ \bar{c}_{2j} \end{bmatrix} = \hat{B}^{-1} \begin{bmatrix} p_j \\ c_{1j} \\ c_{2j} \end{bmatrix} = \begin{bmatrix} B^{-1}p_j \\ c_{1j} - \pi_1 p_j \\ c_{2j} - \pi_2 p_j \end{bmatrix}. \quad (42)$$

Таким образом, характеристические разности, необходимые для вычислений $\bar{c}_j$ в (39), могут быть определены с использованием двойственных оценок π_1 и π_2 и данных исходной задачи по формулам

$$\bar{c}_{1j} = c_{1j} - \pi_1 p_j, \quad (43)$$

$$\bar{c}_{2j} = c_{2j} - \pi_2 p_j. \quad (44)$$

Описываемый алгоритм, по существу, является модифицированным симплекс-методом. При этом производная по направлению $\bar{c}_j$ используется в качестве характеристической разности. Дополнительные вычисления связаны с появлением в матрице B^{-1} дополнительных строк, вычислением $\bar{c}_{1j}$ и $\bar{c}_{2j}$ ((43) и (44)) и, наконец, вычислением $\bar{c}_j$ по формуле (39). На каждом шаге алгоритма значение целевой функции убывает и в предположении невырожденности получение оптимального плана гарантируется за конечное число шагов.

2.11.2. Применение метода одновременного решения прямой и двойственной задач к координирующей задаче

Задача, которую мы рассмотрим, является p -блочной задачей со связывающими отражениями:

минимизировать

$$z = \sum_i c_i x_i \quad (1)$$

при ограничениях

$$\sum_i A_{it} x_i = b_0, \quad (2)$$

$$B_t x_t = b_t, \quad x_t \geq 0 \quad (i = 1, \dots, p). \quad (3)$$

Если предположить, что множества

$$S_t = \{x_t | B_t x_t = b_t, x_t \geq 0\} \quad (4)$$

являются ограниченными, то, как было показано в § 2.4, координирующая задача будет иметь такой вид:

минимизировать

$$z = \sum_{i,j} f_{ij} \lambda_{ij} \quad (5)$$

при ограничениях

$$\sum_{i,j} p_{ij} \lambda_{ij} = b_0, \quad (6)$$

$$\sum_j x_{ij} = 1, \quad i=1, \dots, p, \quad (7)$$

где

$$f_{ij} = c_i x_i^j, \quad (8)$$

$$p_{ij} = A_i x_i^j, \quad (9)$$

а x_i^j — j -я крайняя точка множества S_i . Белл [16] применил метод одновременного решения прямой и двойственной задач к координирующей задаче. Для того чтобы начать процесс вычислений необходимо иметь допустимое решение двойственной задачи. Задача, двойственная к задаче (5) — (7), имеет такой вид: максимизировать

$$v = b'_0 \pi + \sum_{i=1}^p \pi_{0i} \quad (10)$$

при ограничениях

$$\bar{f}_{ij} = f_{ij} - \pi p_{ij} - \pi_{0i} \geq 0 \quad \text{для всех } i, j. \quad (11)$$

Учитывая (8) и (9), запишем ограничения двойственной задачи следующим образом:

$$\pi_{0i} \leq (c_i - \pi A_i) x_i^j. \quad (12)$$

Эти условия будут выполнены, если

$$\pi = 0, \quad (13)$$

$$\pi_{0i} = \min \{c_i x_i^j \mid B_i x_i = b_i, x_i \geq 0\}. \quad (14)$$

Все вычисления осуществляются в соответствии с процедурой одновременного решения прямой и двойственной задач с учетом факта, что координирующая задача имеет колоссальное количество столбцов, которые не могут быть заданы в явном виде.

Представим исходную структуру задачи для исследования метода решения прямой и двойственной задач в следующем

виде:

$$\sum_{i,j} p_{ij} \lambda_{ij} + x_1 = b_0, \quad (15)$$

$$\sum_j \lambda_{ij} + x_{2i} = 1, \quad i = 1, \dots, p, \quad (16)$$

$$\sum_{i,j} \bar{d}_{ij} \lambda_{ij} = w - w_0, \quad (17)$$

$$\sum_{i,j} \bar{f}_{ij} \lambda_{ij} = z - z_0, \quad (18)$$

где x_1 и x_{2i} — искусственные переменные и форма недопустимости w (w — форма) представлена в каноническом виде в результате вычитания (15) и (16) из w [8, стр. 244]. Запишем оценки начального базиса относительно (17):

$$\hat{\sigma} = (\sigma, \sigma_{01}, \dots, \sigma_{0p}) = (1, 1, \dots, 1), \quad (19)$$

а (13) и (14) определяют начальное двойственное допустимое решение

$$\hat{\pi} = (0, \pi_{01}, \dots, \pi_{0p}). \quad (20)$$

На первом шаге алгоритма [8, стр. 245] одновременного решения прямой и двойственной задач значения переменных λ_{ij} первой ограниченной задачи определяются так, чтобы соответствующие x_i были бы оптимальными для (14), так как тогда из (11) следует $\bar{f}_{ij} = 0$.

Этим переменным соответствуют столбцы

$$p_{ij} = \begin{pmatrix} A_i x_i^j \\ u_j \end{pmatrix}. \quad (21)$$

Они являются небазисными столбцами первой ограниченной прямой задачи и учитываются при минимизации формы недопустимости с использованием модифицированного симплекс-метода.

Единственный непосредственный способ определения окончания процесса вычислений на втором шаге алгоритма заключается в проверке условия $\bar{d}_{ij} \geq 0$ путем нахождения $\min_j \bar{d}_{ij}$ для всех i в результате решения последовательности подзадач. Однако такую проверку можно осуществить и более экономным способом, используя двойственную задачу. Если $w_0 = 0$, то процесс закончен, текущее решение является оптимальным. Если $w_0 > 0$, то или допустимого решения прямой задачи не существует, т. е. $\min w > 0$, или просто не достигнуто еще значение

$w=0$. Для получения однозначного ответа воспользуемся фактом, что недопустимость решений прямой и допустимость решения двойственной задач означает неограниченность оптимума двойственной задачи.

Таким образом, условие $\min w > 0$ означает, что на третьем шаге алгоритма мы можем встретиться с неограниченным решением задачи ($k^* = +\infty$). Переходя затем к 3-му шагу алгоритма, определим новое допустимое решение $\hat{\pi}^*$ двойственной задачи

$$\hat{\pi}^* = \hat{\pi} + k^* \sigma, \quad (22)$$

где [8, стр. 246]

$$k^* = \min_{\bar{d}_{ij} < 0} \frac{\bar{f}_{ij}}{-\bar{d}_{ij}}. \quad (23)$$

Так как общее количество коэффициентов f_{ij} и d_{ij} очень велико, то для определения минимума в (23) обычно решаются подзадачи.

По определению,

$$\bar{d}_{ij} = -\sigma p_{ij} - \sigma_{0i} = -\sigma A_i x_i^l - \sigma_{0i}, \quad (24)$$

$$\bar{f}_{ij} = f_{ij} - \pi p_{ij} - \pi_{0i} = (c_i - \pi A_i) x_i^l - \pi_{0i}. \quad (25)$$

Поэтому (23) можно представить в виде

$$k^* = \min_{-\sigma A_i x_i^l - \sigma_{0i} < 0} \frac{(c_i - \pi A_i) x_i^l - \pi_{0i}}{\sigma A_i x_i^l - \sigma_{0i}}, \quad (26)$$

x_i^l — крайние точки выпуклого ограниченного множества S_i и, как это следует из теоремы 2.11.1-3, минимум дробно-линейной функции, определенной на таком множестве, достигается в крайней точке. Таким образом, для каждого i минимум функции (26) может быть найден в результате решения следующей задачи дробно-линейного программирования:

минимизировать

$$\frac{(c_i - \pi A_i) x_i - \pi_{0i}}{\sigma A_i x_i + \sigma_{0i}} \quad (27)$$

при ограничениях

$$B_i x_i = b_i, \quad x_i \geq 0, \quad (28)$$

$$\sigma A_i x_i + \sigma_{0i} > 0. \quad (29)$$

Для каждого i решается задача (27)–(29) и выбирается наименьшее значение целевой функции k^* . Если ни для одного i не существует точек x_i^l , для которых $\bar{d}_{ij} < 0$ (таким образом,

речь идет о точках, удовлетворяющих условию (29)), то, как уже отмечалось, $k^* = +\infty$ и координирующая задача не имеет допустимого решения.

Затем строится новая ограниченная координирующая задача и процесс вычислений в соответствии с алгоритмом одновременного решения прямой и двойственной задач повторяется. Новая задача получается в результате использования всех базисных и некоторых небазисных столбцов, для которых $\bar{f}_i = 0$. Эти столбцы имеют вид (21), где теперь x_i^j — решение любой из подзадач (27) — (29), для которых значение целевой функции равно числу k^* .

Заметим, что при этом можно использовать и другие варианты оптимальных решений подзадач. Если, например, один из них будет опущен (для него $\bar{f}_{ij}^* = 0$), то на следующем цикле, если соответствующий множитель $\bar{d}_{ij}$ отрицателен, новое число k^* будет равно нулю. Такая итерация, вообще говоря, является излишней в том смысле, что в результате ее осуществления значение целевой функции двойственной задачи не возрастает. Для того чтобы не просматривать всех вариантов оптимального решения подзадачи, Белл предложил решать вначале те подзадачи, значения целевых функций которых на предыдущем цикле были равны k^* . Если новые оптимальные значения функции равны нулю, то не требуется решать новые подзадачи, и новая ограниченная координирующая задача находится на основе полученного решения.

Алгоритмы Чарнса — Купера (см. п. 2.11.1) можно использовать для решения подзадач. Необходимо решить только одну линейную задачу, имеющую положительный знаменатель. Если не существует точек в которых знаменатель (27) положителен, то линейная задача не имеет допустимого решения, так как при этом не существует точек, удовлетворяющих (3), для которых выполняются условия $qy_i + \beta t = t(qx_i + \beta) = 1$, $t \geq 0$. Как мы уже отмечали выше, это означает, что в результате решения такой подзадачи не появляется ограничение сверху на значение k .

2.11.3. Числовой пример использования метода одновременного решения прямой и двойственной задач

Вернемся к задаче, рассмотренной в § 2.5, координирующая задача которой имеет вид:

минимизировать

$$z = \sum_i (c_1 x^i) + \sum (c_2 y^i) \beta_i$$

при ограничениях

$$\sum_i (a_1 x^i) \alpha_i + \sum_i (a_2 y^i) \beta_i + s = 40,$$

$$\sum_i \alpha_i = 1,$$

$$\sum_i \beta_i = 1,$$

$$\alpha_i, \beta_i, s \geq 0,$$

где $c_1 = (-1, -1)$, $c_2 = (-2, -1)$, $a_1 = (1, 2)$, $a_2 = (2, 1)$, x^i и y^i — крайние точки множеств S_1 и S_2 (см. § 3.5):

$$x^1 = (6, 8), \quad y^1 = (10, 5),$$

$$x^2 = (10, 0), \quad y^2 = (5, 10),$$

$$x^3 = (0, 10), \quad y^3 = (10, 0),$$

$$x^4 = (10, 0), \quad y^4 = (0, 10),$$

$$y^5 = (0, 0).$$

Для того чтобы начать вычисления в соответствии с процедурой одновременного решения прямой и двойственной задач, необходимо иметь начальное допустимое решение задачи, двойственной к координирующей задаче. Такая двойственная задача имеет следующие ограничения:

$$\left. \begin{aligned} (a_1 x^i) \pi_1 + \pi_{01} &\leq c_1 x^i, \quad i = 1, \dots, 4, \\ (a_2 y^i) \pi_1 + \pi_{02} &\leq c_2 y^i, \quad i = 1, \dots, 5. \end{aligned} \right\} \quad (30)$$

Этим ограничениям удовлетворяют следующие значения переменных:

$$\pi_1 = 0,$$

$$\pi_{01} = \min \{c_1 x^i \mid x^i \in S_1\} = -14, \text{ достигается при } x^1 = (6, 8), \quad (31)$$

$$\pi_{02} = \min \{c_2 y^i \mid y^i \in S_2\} = -25, \text{ достигается при } y^1 = (10, 5). \quad (32)$$

Заметим, что большинство выражений $(c_1 x^i)$ и $(c_2 y^i)$ являются отрицательными, тогда как в соответствии с методом одновременного решения прямой и двойственной задач для того, чтобы обеспечить допустимость плана двойственной задачи, они должны быть неотрицательными. Неотрицательность этих скалярных произведений можно обеспечить в результате умножения ограничений координирующей задачи на $(\pi_2, \pi_{01}, \pi_{02})$ и последующего их вычитания из z . Затем, добавляя искусственные

переменные и w -форму, получим

$$\left. \begin{aligned} \sum_i (a_1 x^i) \alpha_i + \sum_i (a_2 y^i) \beta_i + s + x_a^2 &= 40, \\ \sum_i \alpha_i &+ x_a^2 = 1, \\ \sum_i \beta_i &+ x_a^2 = 1, \\ \sum_i (\bar{c}_i^x) \alpha_i + \sum_i (\bar{c}_i^y) \beta_i - \pi_1 s - z &= 39, \\ x_a^1 + x_a^2 + x_a^3 - w &= 0, \end{aligned} \right\} \quad (33)$$

где $\bar{c}_i^x = c_i x^i - (a_1 x^i) \pi_1 - \pi_{01}$, $\bar{c}_i^y = c_i y^i - (a_2 y^i) \pi_1 - \pi_{02}$ — дополнительные переменные в ограничениях (30) двойственной задачи.

Итерация 1

Исходная ограниченная прямая задача. Исходная ограниченная прямая задача имеет столбцы, соответствующие переменным x_a^1 , x_a^2 , x_a^3 , $-z$, $-w$ системы (33), а также столбцы, соответствующие переменным, входящим в целевую функцию z с нулевыми коэффициентами. К таким столбцам относятся столбцы, соответствующие крайним точкам, в которых достигается минимум в (31) и (32), а также дополнительным переменным:

$$\begin{bmatrix} a_1 x^1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 22 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} a_2 y^1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 25 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Так как все столбцы исходной ограниченной прямой задачи имеют нулевые коэффициенты в z -уравнении, это уравнение не влияет на решение и поэтому может быть опущено.

Начальная матрица базиса ограниченной прямой задачи и матрица обратная к ней имеют следующую структуру:

$$\begin{matrix} x_a^1 & x_a^2 & x_a^3 & -w \end{matrix}$$

$$B = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad B^{-1} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ -1 & -1 & -1 & 1 \end{bmatrix}.$$

Применяя модифицированный симплекс-метод, получим решение ограниченной прямой задачи:

Характеристические разности			
цикл	α_1	β_1	z
0	-23	-26	0
1	-23	0	-1
2	0	0	$\frac{1}{22}$
			$\frac{22}{22}$

Цикл 0: β_1 вводится, x_a^3 выводится

Базисные переменные	Столбцы матрицы условий координирующей задачи				Свободные члены	Преобразованный столбец, вводимый в базис
x_a^1	1				40	25
x_a^2		1			1	0
x_a^3			1		1	1
$-w$	-1	-1	-1	1	-42	-26

Цикл 1: α_1 вводится, x_a^1 выводится

x_a^1	1			-25		15	22
x_a^2		1				1	1
β_1			1	1		1	0
$-w$	-1		-1	25	1	-16	-23

Цикл 2: оптимальный

α_1	$\frac{1}{22}$			$-\frac{25}{22}$		$\frac{15}{22}$	
x_a^2	$-\frac{1}{22}$		1	$\frac{25}{22}$		$\frac{7}{22}$	
β_1				1		1	
$-w$	$\frac{1}{22}$		-1	$-\frac{25}{22}$	1	$-\frac{7}{22}$	

Проверка оптимальности. Так как $w_0 = 7/22 > 0$, текущее решение не является оптимальным. Для того чтобы установить, существует ли допустимое решение прямой задачи, вместо проверки условия $\bar{d}_i \geq 0$ перейдем к следующему шагу. Если

решение двойственной задачи неограниченно (т. е. $k^* \rightarrow \infty$), то прямая задача не имеет допустимого решения. В этом случае все подзадачи не имеют допустимых решений.

Новое двойственное решение. Из последней таблицы получим двойственные переменные относительно формы недопустимости:

$$\sigma = (\sigma_1, \sigma_{01}, \sigma_{02}) = \left(-\frac{1}{22}, 1, \frac{25}{22}\right).$$

Новое двойственное решение имеет вид:

$$(\pi)_{\text{нов}} = (\pi)_{\text{стар}} + k\sigma = (0, -14, -25) + k\left(-\frac{1}{22}, 1, \frac{25}{22}\right),$$

где $k=k^*$ определяется по формуле (23). Для этого необходимо найти решения подзадач, которые являются задачами дробно-линейного программирования типа (27) — (29). Запишем эти подзадачи.

Подзадача 1.

Минимизировать

$$\frac{(c_1 - \pi_1 a_1)x - \pi_{01}}{\sigma_1 a_1 x - \sigma_{01}} = 22 \left[\frac{-x_1 - x_2 + 14}{-x_1 - 2x_2 + 22} \right]$$

при ограничениях

$$\begin{aligned} (x_1, x_2) &\in S_1, \\ -x_1 - 2x_2 + 22 &> 0. \end{aligned}$$

Подзадача 2.

Минимизировать

$$\frac{(c_2 - \pi_2 a_2)y - \pi_{02}}{\sigma_2 a_2 y - \sigma_{02}} = 22 \left[\frac{-2y_1 - y_2 + 25}{-2y_1 - y_2 + 25} \right] = 22$$

при ограничениях

$$(y_1, y_2) \in S_2, \quad -2y_1 - y_2 + 25 > 0.$$

Решения подзадач. $x=x^*=(10,0)$, значение целевой функции $=22/3$, $y=y^*$ или y^* или y^* или y^* , значение целевой функции $=22$, k^* равно наименьшему значению целевых функций:

$$k^* = \min\left(\frac{22}{3}, 22\right) = \frac{22}{3}.$$

Таким образом, получено новое двойственное решение:

$$(\pi)_{\text{нов}} = (0, -14, -25) + \frac{22}{3} \left(-\frac{1}{22}, 1, \frac{25}{22}\right) = \left(-\frac{1}{3}, -\frac{20}{3}, -\frac{50}{3}\right).$$

Сравнив его с заключительной таблицей ограниченной координирующей задачи, убедимся, что полученное двойственное решение является оптимальным.

Итерация 2

Вторая ограниченная прямая задача. Столбцами новой ограниченной прямой задачи являются столбцы оптимального базиса первой ограниченной прямой задачи, включая и столбец, соответствующий оптимальному решению подзадачи 1:

$$\begin{bmatrix} a_1 x^2 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 1 \\ 0 \\ 0 \end{bmatrix}.$$

Ниже приведено решение задач с использованием модифицированного симплекс-метода:

Характеристическая разность	
цикл	σ_2
0	12 -22
1	0

Цикл 0: α_2 вводится, x_2^2 выводится

Базисные переменные	Столбцы матрицы условий координирующей задачи				Свободные члены	Преобразован- ный столбец, зводимый в базис
α_1	$\frac{1}{22}$		$\frac{25}{-22}$		$\frac{15}{22}$	$\frac{10}{22}$
x_2^2	$\frac{-1}{22}$	1	$\frac{25}{22}$		$\frac{7}{22}$	$\frac{12}{22}$
β_1			$\frac{1}{22}$		$\frac{1}{22}$	$\frac{0}{22}$
$-w$	$\frac{1}{22}$	-1	$\frac{25}{-22}$	1	$\frac{-7}{22}$	$\frac{-12}{22}$

Цикл 1: оптимальный

α_1	$\frac{1}{12}$	$\frac{-10}{12}$	$\frac{-25}{12}$		$\frac{5}{12}$
α_2	$\frac{-1}{12}$	$\frac{22}{12}$	$\frac{25}{12}$		$\frac{7}{12}$
β_1			1		$\frac{1}{12}$
$-w$				1	0

Так как $w_0 = 0$, то текущее решение является оптимальным. Теперь легко получить соответствующее решение исходной задачи:

$$x = \alpha_1 x^1 + \alpha_2 x^2 = \frac{5}{12}(6,8) + \frac{7}{12}(10,0),$$

$$y = \beta_1 y^1 = (10,5).$$

Оно совпадает с решением, полученным ранее (см. § 3.5).

§ 2.12. СРАВНЕНИЕ ТРЕХ АЛГОРИТМОВ РЕШЕНИЯ КООРДИНИРУЮЩЕЙ ЗАДАЧИ *)

В главах I и II для решения координирующей задачи Данцига — Вулфа были использованы три наиболее известных алгоритма решения задач линейного программирования: прямой симплекс-метод (§ 2.4), двойственный симплекс-метод (§ 1.6) и, наконец, метод одновременного решения прямой и двойственной задач (§ 2.11.2). Хотя об эффективности применения этих методов к координирующей задаче можно судить лишь на основе вычислительных экспериментов, попытаемся все-таки дать их сравнительную оценку. При этом будем учитывать следующие основные факторы:

1. Условия обеспечения допустимости прямой задачи на каждом цикле вычислений.
2. Сложность генерирования необходимой информации на основе решения подзадач.
3. Частота обращений к решению подзадач и сложность их решения.

Из всех перечисленных выше алгоритмов только прямой симплекс-метод гарантирует допустимость плана задачи с самого начала вычислений. Это обстоятельство является очень важным, так как позволяет не доводить всех вычислений до конца и ограничиться получением субоптимального решения. При этом целесообразно использовать формулу для определения нижней границы значений целевой функции (см. § 2.7). Использование этой формулы, как показали вычислительные эксперименты Джекобсена, позволяет в значительной мере преодолеть основной недостаток метода декомпозиции Данцига — Вулфа — плохую сходимость прямого алгоритма вблизи точки оптимума. Последнее обстоятельство в значительной мере способствовало поиску других методов решения задачи.

С точки зрения условий 2 и 3 подзадачи предпочтительнее решать симплекс-методом, для реализации которого разработа-

*) В этом разделе использован материал работы Джекобсена [19].

ны эффективные стандартные программы для ЭВМ. Применение двойственных методов связано с использованием методов решения задач дробно-линейного программирования. Программы решения таких задач, как правило, менее доступны. Так как множества ограничений подзадач не изменяются от итерации к итерации, предыдущий оптимальный базис может быть использован для нового решения. Конечно, если матрицы условий подзадач имеют специальную структуру, то для решения подзадач целесообразно применять специальные алгоритмы (например, методы решения транспортных задач или задач о максимальных потоках в сетях). Кроме того, любые крайние точки, определяемые в результате решения подзадачи, которым соответствуют отрицательные характеристические разности, могут быть использованы в ограниченной координирующей задаче (см. § 2.4). Таким образом, на каждом цикле не обязательно решать все подзадачи (хотя без их реализации нельзя найти нижнюю границу значений функционала) или осуществлять их решения до получения оптимального плана. (Хотя, как будет показано в § 3.1, в этом случае число итераций может возрасти.) Таким образом, процедура является достаточно гибкой и позволяет наиболее эффективно использовать информацию, получаемую в ходе решения.

Основной недостаток всех двойственных методов, так же как и алгоритма одновременного решения прямой и двойственной задачи, состоит в том, что они вплоть до получения оптимального плана не обеспечивают допустимость исходной задачи. Это обстоятельство препятствует прекращению вычислений на какой-либо промежуточной итерации. Что касается подзадач с дробно-линейным функционалом, то сложность их решения зависит от особенностей применяемого алгоритма.

Метод Чарнса — Купера обеспечивает получение решения путем решения одной задачи линейного программирования (см. п. 2.11.1). Эта задача формируется в результате добавления одной переменной (t в 2.11.1—25) и одного ограничения (2.11.1—27) к ограничениям подзадачи $B_i x_i = b_i$, $x_i \geq 0$; так как это ограничение является знаменателем дроби и зависит от двойственной переменной σ координирующей задачи, то оно будет изменяться на каждом цикле, что и не позволяет учитывать особенность структуры подзадачи и затрудняет ее решение.

Поэтому Джекобсен рекомендует вместо алгоритма Чарнса — Купера использовать метод Абади — Вильямса (см. [13] в главе 6). В этом случае решение задачи дробно-линейного программирования заменяется решением последовательности линейных задач с ограничениями вида $B_i x_i = b_i$, $x_i \geq 0$, но с разными целевыми функциями. Метод позволяет учесть специфику структуры матриц B_i и использовать «старый» оптимальный базис для

нахождения нового решения. Недостаток его очевиден: необходимо решать большое количество задач линейного программирования.

Далее. При использовании любого из этих методов все подзадачи должны быть решены полностью, так как в противном случае не может быть гарантирована допустимость двойственного решения.

С точки зрения условия 2 двойственные методы уступают прямым методам. Это прежде всего относится к двойственному симплекс-методу, для которого отсутствуют процедуры, основанные на использовании неоптимальных решений. Метод одновременного решения прямой и двойственной задач эффективнее в том смысле, что допускает использование различных вариантов оптимальных решений подзадач для генерации столбцов новой ограниченной координирующей задачи. Однако использование околооптимальных решений здесь не дает преимущества, в отличие от прямых методов. Кроме того, как уже отмечалось, если не используются все варианты оптимальных решений, значение целевой функции двойственной задачи может на следующей итерации не возрасти. Таким образом, теряется гибкость, присущая прямым методам.

В заключение сделаем несколько дополнительных замечаний относительно случая неограниченности множества ограничений подзадач. При использовании прямых методов возникающие при этом некоторые трудности легко преодолеваются, чего нельзя сказать относительно всех двойственных методов. Чтобы убедиться в этом, напомним, что начальное двойственное решение (2.11.2—13) и (2.11.2—14) может быть использовано, если каждая целевая функция $c_i x_i$ имеет конечный минимум на множестве ограничений $B_i x_i = b_i$, $x_i \geq 0$. В противном случае вводится дополнительное ограничение вида $\sum_i x_i \leq m$, где m — достаточно большое положительное число. Однако если это ограничение является жестким в точке оптимума, необходимо осуществить проверку, является ли решение координирующей задачи неограниченным. При использовании прямого симплекс-метода ответ на этот вопрос мы получаем в процессе вычислений.

Чтобы наши суждения относительно прямых методов не были слишком односторонними, сделаем несколько предостережений. Дело в том, что прямые методы в силу ряда исторических условий были предметом более детального анализа. Двойственные методы, несмотря на отмеченные недостатки, требуют дальнейших исследований. Во многих случаях использование их сокращает количество циклов вычислений и экономит время. Однако только вычислительные эксперименты могут внести окончательную ясность в обсуждаемую проблему.

Задачи и упражнения

1. Доказать теорему 2.2—3.
2. Решить пример из § 2.5 с использованием координирующей задачи с одним ограничением и не используя ограниченную координирующую задачу.
3. Рассмотреть задачу из [20] (стр. 407):
минимизировать

$$z = -x_1 - 8x_2 - 5x_3 - 6x_4$$

при ограничениях

$$x_1 + 4x_2 + 5x_3 + 2x_4 \leq 7,$$

$$2x_1 + 3x_2 \leq 6,$$

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

$$3x_3 + 4x_4 \geq 12,$$

$$x_3 \leq 4,$$

$$x_4 \leq 3,$$

$$x_i \geq 0, \quad i = 1, \dots, 4.$$

Решить задачу методом декомпозиции Данцига — Вулфа, используя координирующую задачу с двумя ограничениями и ограниченную координирующую задачу с двумя новыми столбцами на каждом цикле. Сравнить полученное решение с решением в работе [20].

4. Пусть дана блочная линейная задача со связывающими ограничениями. Количество блоков равно пяти. Объединив вместе два первых и три последних блока, применить метод декомпозиции Данцига — Вулфа с координирующей задачей с двумя ограничениями. Записать подзадачи и признак оптимальности. Кратко описать итеративную процедуру решения.

5. Рассмотреть задачу линейного программирования с ограничениями сверху на значения некоторых переменных. Использовать метод декомпозиции и построить координирующую задачу без учета ограничений сверху. Записать в явном виде решение подзадач и описать итеративную процедуру решения.

6. Дана многошаговая задача линейного программирования (так называемая ступенчатая система). Применить для ее решения метод декомпозиции, рассматривая ограничения каждой последующей «ступени» в качестве связывающих ограничений, а оставшиеся ограничения — в качестве условий подзадач. Описать метод, основанный на такой идее разложения исходной задачи (сравнить со способом построения подобных процедур решения, предложенным Данцигом [8]).

7. Дать детальное описание метода декомпозиции Данцига — Вулфа для решения блочной задачи с неограниченными множествами ограничений подсистем. Обосновать каждый шаг алгоритма. Показать способ построения экстремального луча для подзадачи в случае, когда использование симплекс-метода указывает на неограниченность множества решений этой подзадачи.

8. Рассмотреть блочную задачу со связывающими переменными (2.5—10) — (2.5—11). Показать, что задача, двойственная к ней, может быть представлена в таком виде:

максимизировать

$$\sum_{i=1}^n p_i b'_i \pi_i$$

при ограничениях

$$\sum_{i=1}^n p_i A' \pi_i \leq c, \quad B' \pi_i \leq d, \quad i = 1, \dots, n.$$

Применить для ее решения метод Данцига — Вулфа с использованием координирующей задачи с одним ограничением (строк). Описать итеративный метод решения. Объяснить, как при этом наилучшим образом можно учесть многократное использование матриц A и B в процессе вычислений. Показать, что оптимальный вектор оценок ограничений (строк) координирующей задачи, которые не являются выпуклыми, является оптимальным вектором (решением) исходной задачи со связывающими переменными.

9. Используя результаты § 2.7, получить нижнюю границу значений целевой функции для транспортной и обобщенной транспортной задач. Назвать наиболее общий класс задач линейного программирования, для которого такая граница может быть определена. Будет ли в точке оптимума соответствующее условие выполняться в форме строгого неравенства? Если да, то обеспечивает ли монотонная сходимость к оптимальному значению?

10. Построить транспортную задачу с четырьмя пунктами производства и двумя пунктами потребления. Решить задачу, применяя к ней метод декомпозиции и симплекс-метод. Сравнить результаты решения.

11. Показать, что для решения задачи из § 2.9 необходимо только просмотреть элементы матрицы $(c_{ij} - a_i \pi_j)$.

12. Доказать, что при выполнении предположения 1 (§ 2.11.1) $f(x) = -px/qx$ — строго квазивыпуклая функция на множестве S .

13. Доказать, что любой локальный минимум строго квазивыпуклой функции является в то же время глобальным.

14. Придумать пример, опровергающий предыдущее утверждение, если в нем опущено слово «строго».

15. Доказать теорему 2.11.1—2 без предположения 1, сохраняя условия, что px и qx не обращаются одновременно в нуль на множество S (см. [13], учитывая, что доказательство в этом источнике приведено с ошибкой).

16. Верно ли следующее утверждение: функция $f(x)$ достигает своего минимального значения на выпуклом многогранном множестве S в крайней точке S , если $f(x)$ — квазивогнутая функция на множестве S . Привести пример.

17. Задана функция

$$f(x) = \frac{2x_1 + x_2 - 3}{x_1 - x_2 + 1}.$$

Изобразить на плоскости x_1, x_2 множество

$$S = \{(x_1, x_2) \mid f(x) \leq 1\}.$$

Изобразить также множества

$$S^+ = S \cap \{(x_1, x_2) \mid x_1 - x_2 + 1 > 0\},$$

$$S^- = S \cap \{(x_1, x_2) \mid x_1 - x_2 + 1 < 0\}.$$

Заметим, что множество S не является выпуклым, поэтому $f(x)$ не может быть выпуклой функцией. S^+ и S^- — выпуклые множества, как и должно быть в случае, когда $f(x)$ является как квазивыпуклой, так и квазивогнутой в области, в которой знаменатель функции $f(x)$ положителен.

18. Решить координирующую задачу из упражнения 3, используя метод одновременного решения прямой и двойственной задач.

19. Применить метод декомпозиции с использованием метода одновременного решения прямой и двойственных задач (см. § 2.11.2) к блочной задаче с неограниченными множествами ограничений подсистем (подзадач).

ЛИТЕРАТУРА

1. G. B. Dantzig and P. Wolfe, The Decomposition Algorithm for Linear Programming, *Econometrica*, 9, No. 4, 1961; also *Operations Res.*, 8, Jan., Feb., Oct. 1960.
2. S. Karlin, *Mathematical Methods and Theory in Games, Programming and Economics*, Vol. 1, Addison—Wesley Publishing Company, Inc., Reading Mass, 1967.—Русский перевод: С. Карлин, Математические методы в теории игр, программировании и экономике, «Мир», М., 1964.]
3. A. C. Williams, A Treatment of Transportation Problems by Decomposition, *J. Soc. Ind. Appl. Math.*, 10, No. 1, 1962, pp. 35—48.
4. W. W. Carvin, *Introduction to Linear Programming*, McGraw—Hill, Inc., New York, 1960.
5. S. I. Gass, *Linear Programming, Methods and Applications*, McGraw—Hill, Inc., New York, 2nd ed., 1964.
6. Tse—Hao Tcheng, Scheduling of a Large Forestry—Cutting Problem by Linear Programming Decomposition, Ph. D. Thesis, University of Iowa, 1966.
7. W. J. Baumol and T. Fabian, Decomposition, Pricing for Decentralization, and External Economies, *Management Sci.*, 11, No. 1, 1964, pp. 1—32.
8. G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, N. J., 1963, chap. 23. [Русский перевод, см. стр. 55.]
9. A. S. Manne, Programming of Economic Lot Sizes, *Management Sci.*, 14, 1958, pp. 115—135.
10. B. P. Dzielinski and R. E. Gomory, Optimal Programming of Lot Sizes Inventory and Labor Allocations, *Management Sci.*, 11, No. 9, 1965, pp. 874—890.
11. H. M. Wagner and T. M. Whitin, A Dynamic Version of the Economic Lot Size Model, *Management Sci.*, 5, 1958, pp. 89—96.
12. E. R. Ford and D. R. Fulkerson, A Primal-Dual Algorithm for the Capacitated Hitchcock Problem, *Naval Res. Logistics Quart.*, 4, No. 1, 1957, pp. 47—54.
13. W. S. Dorn, Linear Fractional Programming, IBM, Res. Rept. RC—830, Nov. 1962.
14. P. C. Gilmore and R. E. Gomory, A Linear Programming Approach to the Cutting Stock Problem, Part II, *Operations Res.*, 11, 1963, pp. 863—888.
15. A. Charnes and W. W. Cooper, Programming with Linear Fractional Functionals, *Naval Res. Logistics Quart.*, 9, Sept—Dec. 1962, pp. 181—186.
16. E. J. Bell, Primal-Dual Decomposition Programming, Ph. D. Thesis, Operations Research Center, University of California at Berkeley, 1965; Rept. ORC 65—23.
17. J. Ponstein, Seven Kinds of Convexity, *SIAM Rev.*, 9, No. 1, 1967.
18. B. Martos, The direct Power of Adjacent Vertex Programming Methods, *Management Sci.*, 12, No. 3, 1965, pp. 241—252.
19. S. Jacobsen, Comparison of the Primal, Dual, and Primal—Dual Decomposition Algorithms (Notes on Operations Research 6), Operation Research Center, University of California at Berkeley, 1967, Rept. ORC ()—17.
20. G. Hadley, *Linear Programming*, Addison—Wesley Publishing Company, Inc., Reading, Mass, 1962.

ГЛАВА III

РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С БОЛЬШИМ ЧИСЛОМ СТОЛБЦОВ С ИСПОЛЬЗОВАНИЕМ ПРОЦЕДУРЫ ГЕНЕРАЦИИ СТОЛБЦОВ

Использование метода декомпозиции Данцига — Вулфа состоит из двух различных этапов. Вначале задача со многими строками преобразуется в эквивалентную ей координирующую задачу с небольшим числом строк и большим числом столбцов. Затем для решения координирующей задачи применяется метод генерации столбцов.

Вместе с тем вторая фаза (генерация столбцов) может быть применена и для решения иных проблем, не связанных с построением координирующей задачи. Такие проблемы могут возникнуть в различных ситуациях и обычно не поддаются решению без использования специальных приемов, поскольку число переменных в них доходит до многих миллионов. Примером является задача о раскрое (§ 3.1). В данной главе рассматривается ряд подобных проблем.

Особое внимание уделяется свойствам различных возникающих подзадач и методам, используемым для их решения. Среди них динамическое программирование (§ 3.1, § 3.2), линейное (§ 3.3) и нелинейное (§ 4.4) программирование, а также алгоритмы расчета потоков в сетях (§ 3.5).

§ 3.1. ЗАДАЧА О РАСКРОЕ

Пусть необходимо составить план раскроя (резки) неограниченного числа рулонов бумаги, при котором потребность в кусках различной длины n_i ($i=1, \dots, I$) (в дальнейшем будем их условно называть *заготовками*) удовлетворялась бы за счет использования наименьшего числа рулонов (так как при раскрое неизбежны отходы, то требование минимизации числа использованных рулонов эквивалентно требованию минимизации отходов). Упростим задачу, сделав предположение, что бумага во всех рулонах имеет одинаковую длину l .

Очевидно, что если $l_i \leq l$ ($i=1, \dots, I$), где l_i — длина i -й заготовки, то задача имеет решение. Из каждого рулона можно получать заготовки различной длины. Комбинируя количеством таких заготовок, будем получать каждый раз новый способ раскроя. Таким образом, способ раскроя однозначно определяется вектором $a = (a_1, a_2, \dots, a_I)$, где a_i — количество заготовок i -го вида (в данном случае заготовки различаются только длиной). Имеется конечное число таких способов.

Введем обозначения: a_{ij} — число заготовок i -го вида, получаемых при использовании j -го способа с единичной интенсивностью (т. е. при раскroe одного рулона); $p_j = (a_{1j}, \dots, a_{Ij})$ — вектор, описывающий j -й способ раскроя; x_j — интенсивность использования j -го способа раскроя (количество рулонов, раскрываемых по j -му способу); $n = (n_1, \dots, n_I)$ — вектор спроса на заготовки.

Теперь условия задачи можно записать следующим образом: минимизировать

$$z = \sum_j x_j \quad (1)$$

при ограничениях

$$\sum_j p_j x_j \geq n, \quad (2)$$

$$x_j \geq 0, \quad x_j \text{ — целые числа.} \quad (3)$$

Условие целочисленности переменных (3) значительно усложняет решение задачи. Поэтому во многих случаях целесообразно вместо задачи (1) — (3) решать непрерывную задачу, опуская ограничение (3). Этот прием полностью себя оправдывает в задачах, когда показатели спроса — достаточно большие числа. Последнее облегчает процесс округления дробных значений переменных x_j . Однако даже решение линейного варианта задачи (1) — (3) связано с преодолением ряда вычислительных трудностей из-за колоссального числа векторов-столбцов p_j . Так, например, при раскroe материала длиной в 200 дюймов ($l=200$) на заготовки 40 видов длиной от 20 до 80 дюймов необходимо просмотреть от 10 до 100 миллионов вариантов (способов) раскроя [2]. Однако трудности, связанные с большим количеством столбцов, могут быть преодолены при использовании принципа декомпозиции.

Определим характеристические разности для небазисных переменных x_j ,

$$\bar{c}_j = 1 - \sum_i \pi_i a_{ij}, \quad (4)$$

где π_i — двойственные переменные, соответствующие ограничениям (2). Очевидно, что компоненты вектора a должны

удовлетворять условиям:

$$a_i = \text{неотрицательные целые числа}, \quad (5)$$

$$\sum_{i=1}^I l_i a_i \leq l. \quad (6)$$

Таким образом, нахождение $\min \bar{c}_j$ на множестве всех столбцов эквивалентно решению задачи:

максимизировать

$$\sum_i \pi_i a_i \quad (7)$$

при ограничениях (5) и (6).

Задача такого типа получила название *задачи о ранце* из-за следующей возможной интерпретации ее условий. Отправляясь в поход, необходимо заполнить ранец различными предметами a_i весом l_i , полезность которых измеряется числами π_i . Требуется найти вариант загрузки, максимизирующий функцию полезности набора предметов (7) при ограничении на емкость (6). Задача о ранце эффективно может быть решена, например, методами динамического программирования [3].

Гилмор и Гомори предложили другой метод решения этой задачи, который по сравнению с методами динамического программирования обеспечивает значительную экономию времени счета (в 4—5 раз!). Пусть v — оптимальное значение целевой функции (7). Тогда условия возможности улучшения текущего плана $\min \bar{c}_j < 0$, учитывая (4), можно записать в виде

$$v > 1. \quad (8)$$

Если (8) выполнено, то находится вектор, вводимый в базис. Если условие (8) не соблюдается и если одновременно для дополнительных переменных выполнено

$$\bar{c}_s = \pi_s \geq 0,$$

то текущее решение является оптимальным. Такой подход успешно был использован Гилмором и Гомори при решении большого числа практических задач применительно к условиям бумажной промышленности. Они реализовали четыре серии экспериментальных расчетов, решая задачу о раскрое при различных исходных данных (варьируя длиной бумаги в рулонах и показателями потребности при неизменности видов заготовок). В таблице 3.1.1 представлены условия и результаты решения одной из таких задач. Дополнительно было введено специальное ограничение на число разрезов $\sum_i a_i$ (зависит от количества ножей, применяемых на установке для резки бумаги). Это ограничение легко учитывается при решении задачи о ранце [2]. В по-

Таблица 3.1.1

Стандартная проверочная задача

Длина бумаги в рулоне 218,00. Спец. ограничение равно 5.

Длина заготовки	Потребность	Длина заготовки	Потребность	Длина заготовки	Потребность
81,00	4415	56,25	459	44,50	94
70,00	291	56,00	343	41,25	393
68,25	4765	52,50	766	38,50	47
67,50	4827	52,00	58	38,00	95
66,75	90	51,75	2736	35,00	411
66,00	691	51,00	212	33,50	36
64,00	263	50,00	720	33,00	273
63,75	141	49,50	133	32,00	56
63,00	133	46,50	529	31,50	171
60,00	390	45,50	185	21,50	140
Показатели интенсивности использования способов раскроя (количество рулонов)		Количество заготовок, получаемых при единичной интенсивности использования способа		Длина заготовок	
2382,50000		1		81,00	
		2		68,25	
302,55552		2		81,00	
36,00000		1		56,00	
		1		81,00	
		2		51,75	
		1		33,50	
1194,38860		1		81,00	
		2		67,50	
94,99999		1		67,50	
		1		60,00	
		1		52,50	
		1		38,00	
141,00000		1		67,50	
		1		63,75	
		1		51,75	
		1		35,00	
86,40738		1		67,50	
		1		66,00	
		1		63,00	
		1		21,50	
77,48144		1		60,00	
		3		52,50	
26,40740		2		67,50	
		1		51,00	
		1		31,50	
529,00000		1		67,50	
		2		51,75	
		1		46,50	
133,00000		1		67,50	
		1		66,00	
		1		49,50	
		1		35,00	

Т а б л и ц а 3.1.1 (продолжение)

Показатели интенсивности использования способов раскроя (количество рулонов)	Количество заготовок, получаемых при единичной интенсивности использования способа	Длина заготовок
19,59259	1	63,00
	2	51,75
	1	51,00
56,00000	2	67,50
	1	51,00
	1	32,00
144,24070	2	56,25
	2	52,50
	1	67,50
145,25922	2	51,75
	1	45,50
	1	67,50
5,50000	2	52,00
	1	45,50
	2	67,50
272,99999	1	50,00
	1	33,00
	1	70,00
94,00000	2	51,75
	1	44,50
	3	51,75
53,59262	1	41,25
	1	21,50
	1	67,50
34,24078	2	52,50
	1	45,50
	1	60,00
170,51850	1	56,25
	1	51,75
	1	50,00
169,70368	1	67,50
	2	41,25
	2	66,00
109,99999	1	51,00
	1	35,00
	1	67,50
27,00001	1	63,00
	1	52,50
	1	35,00
90,00000	1	67,50
	1	66,75
	1	57,75
13,48148	1	31,50
	3	56,00
	1	50,00
263,00000	1	64,00
	2	51,75
	1	50,00

Т а б л и ц а 3.1.1 (продолжение)

Показатели интенсивности использования способов раскроя (количество рулонов)	Количество заготовок, получаемых при идентичной интенсивности использования способа	Длина заготовок
54,59259]	1	67,50
	1	66,00
	1	52,50
	1	31,50
197,000000	1	81,00
	1	70,00
	1	66,00
	1	67,50
47,00001	1	60,00
	1	52,00
	1	38,50
	1	

Приложение к табл. 3.1.1

Данные о проценте отходов для задач,
решение которых не было закончено

Серия задач	В	С	Д
A4—1	0,3918	0,3921	0,9355
2		0,4353	1,0387
3			0,9747
A5—1	0,5459	0,4927	

добных задачах оценки различных ограничений (1) часто совпадают. Этот факт позволяет значительно упростить решение, так как, если, например, оценки π_1 и π_2 совпадают, а $l_2 > l_1$, то существует оптимальное решение в (5)—(7), при котором $a_2 = 0$.

Это происходит по следующей причине. Если бы a_2 было положительным, то на такую величину можно было бы увеличить a_1 , не нарушая при этом условия (6) и не изменяя значения целевой функции (7). Таким образом, l_2 и соответствующая переменная a_2 могут быть опущены, что приводит к задаче меньшей размерности.

Рисунок 3.1. дает представление о размерности задач о ранце и о времени, необходимом для их практической реализации. Использование совпадающих одинаковых оценок ограничений позволило уменьшить среднее число переменных с 30 до 18 и тем самым значительно сократить время счета.

Заслуживает внимания предложенный Гилмором и Гомори прием сокращения объема вычислений, называемый «методом медианы». Этот метод связан с решением следующей проблемы. При использовании процедуры симплекс-метода введение в

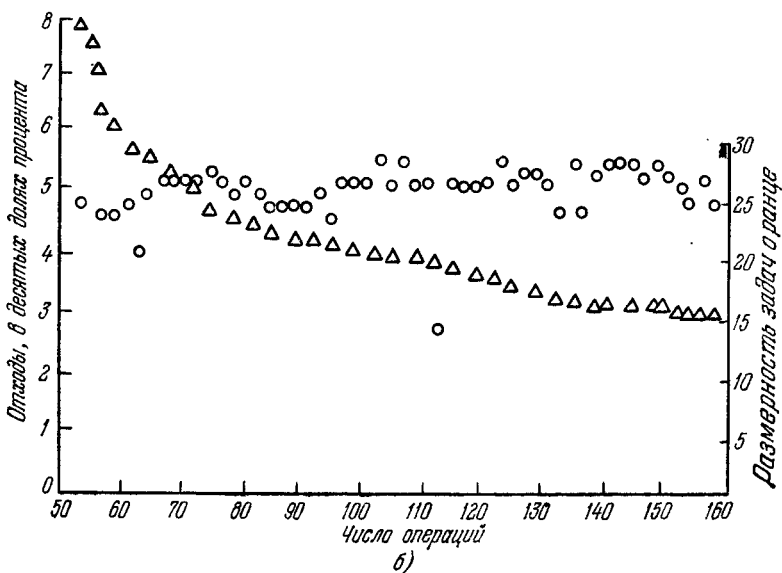
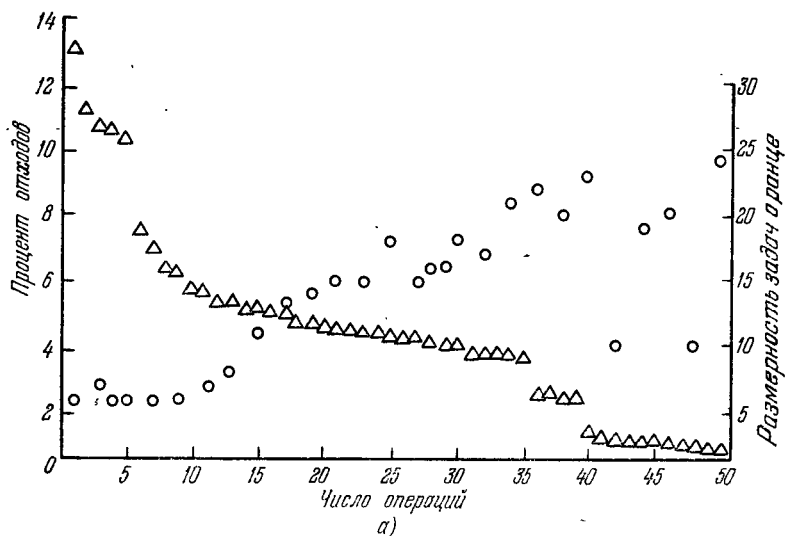


Рис. 3.1. Зависимость процента отходов и размерности задачи о ранце от числа ведущих операций. Следует обратить внимание на разные масштабы на рис. (а) и (б). При этом треугольники соответствуют данным о проценте отходов, а кружки — данным о размерности задачи о ранце.

базис столбца, которому соответствует $\min \bar{c}_j$, не обязательно гарантирует максимальное убывание значения целевой функции z . Это объясняется тем, что значение целевой функции при этом уменьшается на величину $\bar{c}_j x_j$, которая зависит и от значения переменной x_j . Выбор переменной, которой соответствует не наиболее отрицательная характеристическая разность, может дать больший эффект. Для выбора, обеспечивающего наибольшее приращение, вообще говоря, требуется слишком много вычислений, но в данной задаче кое-что удастся сделать.

Из таблицы 3.1.1 видно, что спрос на заготовки (бумагу различного формата) изменяется в достаточно широких пределах. Рассмотрим способ раскроя, при котором получают заготовки с низким и высоким спросом. Естественно, что интенсивность использования такого способа ограничена заготовками с низким спросом. Поэтому для того, чтобы обеспечить получение достаточно больших значений переменных x_j , гарантирующих соответствующее убывание значения z , все заготовки целесообразно разделить на группы заготовок с пониженным и повышенным спросом (т. е. группы заготовок, спрос на которые превосходит и не превосходит значение медианы спроса). В ходе проведения расчетов при осуществлении ведущей операции (т. е. решении задачи о ранце) переменные a_i для заготовок группы с пониженным спросом приравнивались нулю, что в силу отмеченных выше обстоятельств оказалось весьма эффективным. Некоторые результаты таких вычислений приведены в таблице 3.1.2. Все перечисленные в этой таблице задачи решались с использованием метода решения задачи о ранце [2] и с учетом одинаковых двойственных оценок. Вычисления были осуществлены на ЭВМ IBM-7094. Сравнение столбцов таблицы А и В позволяет судить об эффективности «метода медианы», особенно при решении серии задач А4. В столбце С содержится информация относительно случая, когда решение задачи о ранце не доводилось до конца (вычисления прекращались, если $\bar{c}_j < 0$). При этом преследовалась цель выяснить целесообразность введения в базис столбцов, которым соответствовал $\min \bar{c}_j$. Хотя этот вопрос был предметом многих исследований [4], в данном случае процедура нахождения $\min \bar{c}_j$ оказалась более эффективной. Использование этой процедуры привело к сокращению числа ведущих операций и, что более важно при решении задач большой размерности, — уменьшению времени счета. Вычисления также проводились и при случайном выборе столбца, вводимого в базис. Результаты таких вычислений показаны в столбце D. Результаты эти довольно противоречивы. Так, например, среднее число необходимых ведущих операций для серии задач было равно $20m$ (m — количество строк) (при использовании стандартного подхода это число равняется $2,5m$).

Таблица 3.1.2

Результаты экспериментальной проверки различных методов нахождения вводимого в базис вектора

Серия задач	Длины заготовок	% отходов	А		В		С		D	
			Решение задачи о ранце с использованием метода медианы	число ведущих операций	время счета (мин.)	Решение задачи о ранце без использования метода медианы	число ведущих операций	время счета (мин.)	Без решения задачи о ранце, без использования метода медианы	число ведущих операций
A1—1	40	3,0474	148	1,34	2,64	194	333	1,47	541	2,07
	35	0,0184	233	4,50	9,24	347	560	6,30	1104	11,03
	30	0,0616	161	6,32	6,75	171	418	7,19	586	7,84
	25	0,6227	101	0,47	0,93	135	177	0,62	545	1,51
	5	0,0539	90	1,32	2,23	111	214	2,25	370	3,08
A2—1	40	4,7232	110	0,52	0,62	167	122	0,50	316	1,06
	2	7,8219	90	0,31	0,28	76	93	0,30	126	0,40
	3	8,6921	87	0,25	0,22	71	87	0,24	111	0,25
	4	9,6407	47	0,14	0,15	73	64	0,14	76	0,17
	5	5,1748	21	0,09	0,06	49	28	0,07	28	0,08
A3—1	40	5,0845	57	0,33	0,32	73	66	0,33	80	0,52
	35	0,4825	141	3,22	2,34	157	348	3,35	450	3,83
	30	0,2114	83	0,59	0,78	124	371	1,97	518	1,69
	4	0,1484	90	0,54	0,58	95	259	0,81	380	0,92
	5	2,7297	24	0,12	0,07	27	173	0,30	476	0,93
A4—1	40	0,3869	138	5,53	41,58*)	240	637	10,65*	727	15,87*)
	2	0,4326	125	4,20	7,79	208	648	4,58*	878	17,75*)
	3	0,4494	92	1,24	4,52	140	489	3,62	817	19,23*)
	4	0,1415	98	2,51	6,30	141	415	3,66	878	9,19
	5	0,5490	67	1,74	1,57	77	536	5,13	786	18,10*)
A5—1	50	0,4862	291	14,51	16,97*)	540	980	12,51*	2123	19,87

*) Решение задачи не было закончено.

*) Решение задачи не было закончено.

В то же время при решении задачи серии А2 среднее число ведущих операций равнялось $4,4m$, а при решении задачи серии А4 — $27m$. В последнем случае только для одной из пяти задач было получено оптимальное решение при таком числе ведущих операций. Такой результат, вообще говоря, является типичным для многих процедур в математическом программировании (например, при использовании метода отсечений в целочисленном программировании), когда выбор строки или столбца осуществляется случайным образом.

Обобщение на случай дробно-линейной целевой функции. В работе [2] было отмечено, что зачастую число заготовок может задаваться нежестко, например, в виде $n'_i \leq l_i \leq n''_i$ ($n''_i > n'_i$). По данным Гилмора и Гомори [2, стр. 881] во многих случаях можно принять, что $n''_i = 1, 1 n'_i$.

В такой ситуации число рулонов уже не является подходящим критерием, поскольку оно обычно оказывается наименьшим, если производится наименьшее допустимое число заготовок n'_i . В то же время достижение истинной цели — уменьшение процента отходов — может потребовать большего выпуска.

Рассмотрим соответствующую модель. Определим отходы при использовании j -го способа раскроя с единичной интенсивностью

$$w_j = l - \sum_i l_i a_{ij}. \quad (9)$$

Если j -й способ используется с интенсивностью x_j , то общий процент отходов будет равен

$$\frac{z_1}{z_2} = 100 \left(\frac{\sum_j w_j x_j}{\sum_j x_j} \right). \quad (10)$$

Таким образом, возникает следующая задача дробно-линейного программирования:

минимизировать

$$\frac{\sum_j w_j x_j}{\sum_j x_j} \quad (11)$$

при ограничениях

$$\sum_j p_j x_j \geq n', \quad (12)$$

$$\sum_j p_j x_j \leq n'', \quad (13)$$

$$x_j \geq 0. \quad (14)$$

Введем дополнительную переменную s в ограничения (12) и вычтя (12) из (13), получим следующую систему:

$$\sum_i p_i x_i - s = n', \quad (15)$$

$$0 \leq s \leq (n'' - n'). \quad (16)$$

Для решения задачи дробно-линейного программирования (11), (15) — (16) применим метод Гилмора — Гомори (§ 2.11.1). Так как этот метод является модификацией симплекс-метода, то ограничение (16) легко можно учесть без увеличения размеров базиса, используя специальную процедуру (см. § 5.3).

Единственная трудность, возникающая при этом, заключается в выборе столбца из множества столбцов p_j . Если при этом следовать стандартной процедуре симплекс-метода, то такому столбцу должен соответствовать столбец с наименьшим значением производной целевой функции по направлению. Учитывая (2.11.1-39), получим следующие производные целевой функции задачи о раскрое:

$$\bar{c}_j = \frac{\bar{z}_2 \bar{c}_{1j} - \bar{z}_1 \bar{c}_{2j}}{\bar{z}_2^2}, \quad (17)$$

где $\bar{z}_1/\bar{z}_2$ — значение целевой функции в текущей точке.

Из (2.11.1-43) и (2.11.1-44) следует, что

$$\bar{c}_{1j} = c_{1j} - \pi_1 p_j = w_j - \pi_1 p_j, \quad (18)$$

$$\bar{c}_{2j} = c_{2j} - \pi_2 p_j = 1 - \pi_2 p_j. \quad (19)$$

Подставляя (18) и (19) в (17), получим

$$\bar{c}_j = \frac{\bar{z}_2 (w_j - \pi_1 p_j) - \bar{z}_1 (1 - \pi_2 p_j)}{\bar{z}_2^2}$$

или, используя (9),

$$\bar{c}_j = \frac{l \bar{z}_2 - \bar{z}_1 + \sum_i (\bar{z}_1 \pi_{2i} - \bar{z}_2 (\pi_{1i} + l_i)) a_{ij}}{\bar{z}_2^2}. \quad (20)$$

Пусть

$$k = l \bar{z}_2 - \bar{z}_1, \quad (21)$$

$$\bar{\pi}_i = \bar{z}_1 \pi_{2i} - \bar{z}_2 (\pi_{1i} + l_i). \quad (22)$$

Теперь понятно, что для нахождения столбца, вводимого в базис, необходимо вычислить

$$\min \bar{c}_i = \frac{\min \left(k - \sum_i \bar{\pi}_i a_{ij} \right)}{\bar{z}_2^2}, \quad (23)$$

что эквивалентно решению следующей подзадачи:
минимизировать

$$\sum_i \bar{\pi}_i a_i$$

при ограничениях

$$\sum_i l_i a_i \leq 1,$$

где a_i — целевые неотрицательные числа. В результате мы опять пришли к задаче о ранце. Основное отличие от исходной задачи заключается в том, что теперь коэффициенты целевой функции $\bar{\pi}_i$ являются функциями оценок π_1 и π_2 . После решения подзадачи дальнейшие вычисления осуществляются в соответствии с описанной в § 2.11.1 процедурой. Дальнейшее развитие этих идей применительно к задаче о раскрое осуществлено в работе [5].

§ 3.2. ИСПОЛЬЗОВАНИЕ МЕТОДА ГЕНЕРАЦИИ СТОЛБЦОВ ПРИ СОСТАВЛЕНИИ ПРОИЗВОДСТВЕННЫХ ГРАФИКОВ

В § 2.10 был предложен приближенный метод решения задачи, частным случаем которой является задача составления графика выпуска изделий. Этот метод основан на сведении этой задачи к линейной задаче большой размерности.

Для решения последней предлагался метод декомпозиции Данцига — Вулфа. Хотя такой подход вполне оправдан, но при его практической реализации возникают некоторые трудности. Основная трудность и недостаток метода состоит в том, что решение линейной задачи в этом случае является лишь приближенным решением целочисленной задачи, которая, вообще говоря, нас только и интересует. При таком подходе достаточно «хорошее» приближение можно получить только тогда, когда количество изделий намного превосходит число временных промежутков планового периода. Именно в этом случае решение линейной задачи обладает свойством, что большинство переменных будет принимать целочисленные значения. Однако при использовании метода декомпозиции этим свойством не обладает координирующая задача, если вычисления не доведены до получения оптимального плана. Поэтому промежуточные планы могут не иметь смысла и использование нижней оценки (границы) значения целевой функции становится малоэффективным.

Но, как подчеркивалось в [7], задачи со многими столбцами имеют громадное число околооптимальных планов. Поэтому отмеченные недостатки носят весьма серьезный характер.

Ниже мы остановимся на рассмотрении метода решения задачи о производственной программе, сформулированной Манном. Этот метод предложен Лэсдоном и Макки [8]. Особенность метода состоит в том, что исходная задача решается непосредственно без применения координирующей задачи. Трудности, связанные с наличием большого числа столбцов и строк преодолеваются за счет использования методов генерации столбцов и учета двухсторонних ограничений (см. § 5.4). Подзадачи имеют аналогичную структуру, как и при использовании принципа декомпозиции. Базисная матрица имеет меньшую размерность по сравнению с размерностью базисной матрицы координирующей задачи (хотя соответствующие вычисления сложнее).

Поскольку, как правило, на каждом шаге мы имеем дело с базисным решением исходной задачи, то большинство переменных принимают целочисленные значения. Для определения возможности окончания вычислений, до получения точного оптимума, может быть использована нижняя граница значений целевой функции. Возможны и другие приемы, повышающие эффективность алгоритма, что, как показывают экспериментальные расчеты, является одним из основных преимуществ этого метода. Эти же расчеты показали, что по сравнению с методом декомпозиции получение оптимального плана с использованием нового метода гарантируется за гораздо меньшее число итераций (примерно в два раза). При этом трудоемкость подготовительной работы не возрастает.

Процедура решения. Итак, необходимо решить задачу (2.10.1-12) — (2.10.1-15), которую здесь запишем следующим образом:

минимизировать

$$\sum_{i,j} c_{ij} \theta_{ij} \quad (1)$$

при ограничениях

$$\sum_i \theta_{ij} = 1 \quad (j = 1, \dots, J), \quad (2)$$

$$\sum_{i,j} Y_{ij} \theta_{ij} \leq b, \quad (3)$$

$$\theta_{ij} \geq 0, \quad (4)$$

где

$$c_{ij} = c_i(x_i^j), \quad Y_{ij} = Y_i(x_i^j), \quad (5)$$

x_i^j — элементы конечного множества V_i . В практических ситуациях эта линейная задача, как правило, имеет колоссальное число столбцов и большое число строк (сотни или тысячи). Как отмечалось, трудность, связанная с наличием большого числа столбцов, довольно легко преодолевается. В самом деле, пусть B — базисная матрица задачи и $\pi = (\pi_1, \pi_2)$ — вектор двойственных переменных, соответствующих этому базису, где π_1 — оценки ограничений (2) и π_2 — оценки ограничений (3).

Используя эти оценки, определим характеристические разности для небазисных столбцов (векторов)

$$\bar{c}_{ij} = c_{ij} - \pi_2 Y_{ij} - \pi_{1i}. \quad (6)$$

Для того чтобы определить столбец, вводимый в базис, необходимо найти

$$\min_{i,j} \bar{c}_{ij} = \min_i \min_j \bar{c}_{ij}. \quad (7)$$

Нахождение $\min_j \bar{c}_{ij}$ эквивалентно решению подзадачи

$$\min_j [c_{ij} - \pi_2 Y_{ij}] \quad (8)$$

или, используя (5) и учитывая, что x_i^j — элементы V_i , минимизировать

$$c_i(x_i) - \pi_2 Y_i(x_i) \quad (9)$$

при ограничениях

$$x_i \in V_i. \quad (10)$$

Нетрудно понять, что задача (9) — (10) аналогична задаче (2.10.1-42) — (2.10.1-43). Так как V_i — конечное множество, а c_i и Y_i , по предположению, ограничены снизу на множестве V_i , подзадачи (9) — (10) — имеют оптимальные конечные решения для всех значений π_2 . Чтобы найти вводимый в базис столбец, необходимо вычесть π_{1i} из значения целевой функции подзадачи (9) — (10) и затем среди найденных таким образом чисел найти минимальное.

Однако все еще остается трудность, связанная с наличием большого числа ограничений. Преодоление этой трудности связано с учетом специфики ограничений (2). В работах, посвященных методу учета (обобщенных) двухсторонних ограничений (см. также [9]), Данциг и Ван-Слайк доказали, что задачи с ограничениями типа (2) могут быть решены с использованием рабочего базиса, размерность которого равна числу оставшихся ограничений (3), т. е. в нашем случае $(T \times T)$. Для любого базиса в задаче (1) — (4) легко находятся двойственные переменные,

значения базисных переменных и вектор $\bar{P}_s = B^{-1}P_s$ при использовании текущего базиса.

Хотя процедура изменения базиса на каждой итерации алгоритма усложняется по сравнению со стандартным симплекс-методом, в целом она является более эффективной [9] в особенности, если I намного превосходит T .

Процедура, позволяющая учесть наличие в задаче (1) — (4) большого числа ограничений, будет рассмотрена в § 5.4.

Нижняя граница. Так же как и в случае метода декомпозиции, при решении рассматриваемой задачи строится нижняя граница значения целевой функции.

Пусть $\pi = (\pi_1, \pi_2)$ — вектор оценок для базиса B задачи (1) — (4), которую представим в следующем виде:

минимизировать

$$\sum_{i,j} c_{ij} \theta_{ij} = z \quad (11)$$

при ограничениях

$$\sum_j \theta_{ij} = 1, \quad i = 1, \dots, I, \quad (12)$$

$$\sum_{i,j} Y_{ij} \theta_{ij} + s = b, \quad (13)$$

$$\theta_{ij} \geq 0 \text{ для всех } i, j, s_t \geq 0, t = 1, \dots, T, \quad (14)$$

где s_t — дополнительные переменные. Умножив (12) на π_1 и (13) на π_2 и вычитая из (11), получим

$$\sum_{i,j} (c_{ij} - \pi_1 Y_{ij} - \pi_2 s_t) \theta_{ij} - \pi_2 b = z - \pi_1 b - \sum_i \pi_{1i}. \quad (15)$$

Выражение, стоящее в скобках (15), как это следует из (6), — характеристическая разность $\bar{c}_{ij}$. Заменив $\bar{c}_{ij}$ на $\min \bar{c}_{ij}$, получим

$$\sum_i (\min_j \bar{c}_{ij}) \sum_j \theta_{ij} - \pi_2 s \leq z - \pi_2 b - \sum_i \pi_{1i} \quad (16)$$

или, учитывая (12),

$$z \geq \sum_i (\min_j \bar{c}_{ij}) - \pi_2 s + \left(\pi_2 b + \sum_i \pi_{1i} \right). \quad (17)$$

Так как

$$(\pi_1, \pi_2) = c_B B^{-1} \quad (18)$$

и если e_A — I -мерный вектор с компонентами, равными единице (+1),

$$z_B = c_B x_B = c_B B^{-1} \begin{bmatrix} e_I \\ b \end{bmatrix} = (\pi_1, \pi_2) \begin{bmatrix} e_I \\ b \end{bmatrix}, \quad (19)$$

то

$$z_B = \pi_2 b + \sum_i \pi_{2i}. \quad (20)$$

Неравенство (17) выполняется для всех значений z задачи (11) — (14). Поэтому

$$\min z \geq z_B + \sum_i (\min \bar{c}_{ij}) - \pi_2 s. \quad (21)$$

Таким образом, получена нижняя граница значений целевой функции z . Изменение этой оценки в процессе вычислений будет представлено в конце этого раздела.

Одновременное введение в базис нескольких векторов. При использовании метода декомпозиции Данцига — Вульфа для решения задачи (1) — (4) строится координирующая задача, имеющая от одного до I выпуклых ограничений, в зависимости от способа разбиения множества ограничений (2) на соответствующие подмножества. Решение любого такого подмножества ограничений может быть представлено в виде выпуклой комбинации крайних точек соответствующих многогранников. В методе, описанном в § 2.10, координирующая задача имеет одно ограничение, в то время как в изложенном выше методе используется I ограничений (таким образом «воспроизводится» исходная задача). В обоих случаях решаются аналогичные подзадачи, решение которых $x_i(\pi)$, $i=1, \dots, I$ зависит от значения компонент вектора двойственных переменных π . Если координирующая задача имеет p условий выпуклости $1 \leq p \leq I$, то на каждой итерации в результате решения подзадач формируется p новых столбцов (векторов) и только один из них, для которого характеристическая разность равна $\min_i \bar{c}_{ij}$, вводится в базис. Затем

выполняется ведущая операция и определяется новый вектор двойственных переменных $\hat{\pi}$. При этом нередко вектор $\hat{\pi}$ совпадает или незначительно отличается от вектора π , и поэтому столбцы, соответствующие $x_i(\pi)$, опять будут иметь отрицательные характеристические разности, и из них будет выбираться столбец, вводимый в базис. Такой процесс вычислений должен повторяться до тех пор, пока характеристические разности станут неотрицательными. Подобная процедура соответствует частичному решению сокращенной координирующей задачи в § 2.4, когда отбрасываются все столбцы, выводимые из базиса,

и полностью используются все столбцы, определенные из подзадач.

Можно ожидать, что с вычислительной точки зрения нахождение характеристических разностей столбцов (т. е. скалярного произведения двух векторов) будет на порядок менее трудоемкой операцией по сравнению с решением подзадачи (9) — (10). Понятно, что чем больше ограничений используется, тем больше формируется столбцов на каждом цикле, из которых затем выбирается столбец, вводимый в базис. А это в свою очередь повышает вероятность увеличения скорости убывания значения целевой функции исходной задачи. Но увеличение числа учитываемых ограничений в то же время создает дополнительные вычислительные трудности при реализации используемого при таком подходе модифицированного симплекс-метода. Однако эти трудности могут быть преодолены в результате применения метода учета обобщенных двухсторонних ограничений (метод GUB, § 5.4).

Заметим, что подобный подход применим и в случае, когда координирующая задача имеет только одно ограничение. В этом случае генерируются k столбцов ($k \geq 1$) с отрицательными характеристическими разностями, с использованием которых формируется ограничение координирующей задачи. Опыт показывает, что такой прием улучшает сходимость алгоритма. Описанный выше прием может быть применен и при использовании общего метода декомпозиции Данцига — Вулфа. Для матрицы условий исходной задачи со структурой

$$\begin{bmatrix} A_1 & A_2 & \dots & A_n \\ B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_n \end{bmatrix}$$

можно сформулировать координирующую задачу с $1 \leq p \leq n$ ограничениями (см. § 2.4). Вычислительные эксперименты показали, что использование на каждом шаге координирующей задачи $p = n$ приводит к улучшению сходимости метода Данцига — Вулфа.

Задача определения оптимального размера партии изделий с «непрерывной» функцией затрат на переналадку. Проверка эффективности изложенного выше подхода осуществлялась на примере ряда задач. Была решена серия задач по составлению графиков выпуска изделий. Содержание общей задачи этой серии состоит в следующем. Имеется I видов продукции, спрос на которую для каждого временного промежутка t ($t = 1, \dots, T$) задан. При производстве продукции используется взаимоза-
ме-

няемое оборудование. Переход от выпуска изделий одного вида к выпуску изделий другого вида на одном и том же оборудовании связан с затратами на переналадку. Требуется составить график выпуска изделий, при котором спрос удовлетворялся бы полностью, а суммарные затраты на переналадку оборудования и хранение продукции были бы минимальными.

Задача может быть сформулирована в терминах более общей задачи о распределении ресурсов (2.10.1-5) — (2.10.1-7).

Введем обозначения:

m_{it} — количество оборудования (машин), используемого для производства i -го вида продукции в t -м периоде; $m_i = (m_{i1}, m_{i2}, \dots, m_{iT})$; b_t — количество оборудования (машин), которое может быть использовано в t -м периоде; $b = (b_1, \dots, b_T)$; d_{it} — величина спроса на i -й вид продукции в t -м периоде; k_i — количество продукции i -го вида, которое может быть произведено одной единицей оборудования; y_{it} — запас продукции i -го вида в конце периода t , $y_{it} = y_{it-1} + k_i m_{it} - d_{it}$, $t = 1, \dots, T$, y_{i0} — задано; p_s — затраты на одну переналадку; $(y_{it})_{\max}$, $(y_{it})_{\min}$ — верхняя и нижняя границы значений переменной y_{it} ; n_i — максимальное количество оборудования, которое одновременно может быть использовано для производства продукции i -го вида.

Поскольку затраты на переналадку предполагаются не зависящими от вида продукции, общие затраты на переналадку могут быть представлены в виде

$$c_s = p_s \sum_{i=1}^I s_i(m_i), \quad (22)$$

где

$$s_i(m_i) = \sum_{t=1}^T (m_{it} - m_{it-1})_+, \quad m_{i0} \text{ — задано.} \quad (23)$$

Здесь

$$(x)_+ = \begin{cases} x, & \text{если } x \geq 0, \\ 0, & \text{если } x < 0. \end{cases} \quad (24)$$

Суммарные затраты, связанные с хранением (для $y_{it} > 0$) и убытками от дефицита (для $y_{it} < 0$) i -го вида продукции определяются по формуле

$$\gamma_i(m_i) = \sum_{t=1}^T \gamma_{it}(y_{it}), \quad (25)$$

где функции γ_{it} могут иметь любой подходящий вид. Пусть

$$c_i(m_i) = p_s s_i(m_i) + \gamma_i(m_i), \quad (26)$$

Тогда задача приобретает следующий вид:
минимизировать

$$c = \sum_{i=1}^I c_i(m_i) \quad (27)$$

при ограничениях

$$(y_{it})_{\min} \leq y_{it} \leq (y_{it})_{\max} \text{ для всех } i, t, \quad (28)$$

$$m_{it} \leq n_i \text{ для всех } i, t, \quad (29)$$

$$m_{it} \text{ — целые неотрицательные числа для всех } i, t, \quad (30)$$

$$\sum_t m_{it} \leq b_i \quad (i = 1, \dots, T). \quad (31)$$

Пусть

$$V_i = \{m_i \mid m_i, \text{ удовлетворяющие условиям (28) — (30)}\}. \quad (32)$$

Так как n_i — конечные числа, то V_i — конечное множество, которое будет предполагаться непустым. Пусть, далее, m_i^j — j -й элемент множества V_i и

$$c_i(m_i^j) \equiv c_{ij}. \quad (33)$$

Теперь задачу (27) — (31) можно привести к стандартному виду задачи о распределении ресурсов:
минимизировать

$$\sum_{i,j} c_{ij} \theta_{ij} \quad (34)$$

при ограничениях

$$\sum_i \theta_{ij} = 1, \quad i = 1, \dots, I, \quad (35)$$

$$\sum_j m_i^j \theta_{ij} \leq b_i, \quad (36)$$

$$\theta_{ij} \geq 0 \text{ для всех } i, j. \quad (37)$$

Запишем условия i -й подзадачи:
минимизировать

$$p_i s_i(m_i) + \gamma_i(m_i) - \pi_i m_i \quad (38)$$

при ограничениях

$$(y_{it})_{\min} \leq y_{it} \leq (y_{it})_{\max}, \quad t = 1, \dots, T, \quad (39)$$

$$m_{it} \leq n_i, \quad t = 1, \dots, T, \quad (40)$$

$$m_{it} \text{ — целые, неотрицательные числа, } t = 1, \dots, T. \quad (41)$$

Каждая подзадача может быть решена методами динамического программирования с двумя переменными состояния m_{it} и y_{it} .

Результаты вычислительного эксперимента. Ранее было обнаружено, что нахождение первого допустимого плана задачи с использованием полного множества искусственных переменных во многих случаях не эффективно. Поэтому вначале решались подзадачи (38) — (41) при $\pi_2=0$. Если полученные решения удовлетворяли условию (31), то они являлись оптимальными. В противном случае строился допустимый план с использованием переменных θ_{ij} (по одной для каждой подзадачи), а также T дополнительных и искусственных переменных. Это позволяло осуществлять операцию обращения базисной матрицы аналитически и получать хорошее приближение для фазы 2 симплекс-метода. Были решены четыре задачи с использованием трех методов: 1) метода декомпозиции, 2) нового метода, описанного выше, 3) метода с одновременным оцениванием нескольких столбцов.

Задачи решались для шести подпериодов при числе видов продукции равном 6, 10, 10 и 15. В задачах 2 и 3 предполагалось, что на каждом подпериоде общее число агрегатов ограничено 22 и 24 единицами соответственно. Другие параметры, например, спрос на продукцию, задавались таким образом, чтобы исключить возможность получения тривиального решения. Заметим, что уже при незначительном количестве видов продукции число столбцов задачи (34) — (37) становится достаточно большим. Например, если для какого-либо изделия отсутствует ограничение типа (39), то каждому виду изделий будет соответствовать $(n_i+1)^T$ столбцов. При $T=6$ и $n_i=7$ это число равно 3 262 124. В каждой задаче имелся такой вид изделия, а при решении задачи 4 учитывались два изделия с $n_i=7$, одно изделие с $n_i=5$ и два изделия с $n_i=4$. Для сокращения времени счета была сделана попытка введения верхних и нижних границ, но фактически это оказалось бесполезным, поскольку ограничения не достигались ни на одном цикле.

На рисунках 3.2 — 3.5 показано изменение функционала (затрат) в зависимости от числа итераций, выполненных на 2-й фазе симплекс-метода (после получения допустимого плана). Хотя начальные затраты несколько различаются (ввиду использования различных способов построения допустимого решения на фазе 1), эти различия исчезают после нескольких первых итераций фазы 2. Для использования принципа декомпозиции характерно «затягивание», медленная сходимость вблизи оптимального решения, по сравнению с предлагаемым методом. Вероятно, это происходит из-за включения меньшего числа столбцов.

Использование приема оценивания и ввода одновременно нескольких столбцов дает очень существенное уменьшение числа итераций (от 2 до 9 раз). Поскольку для решавшихся задач время

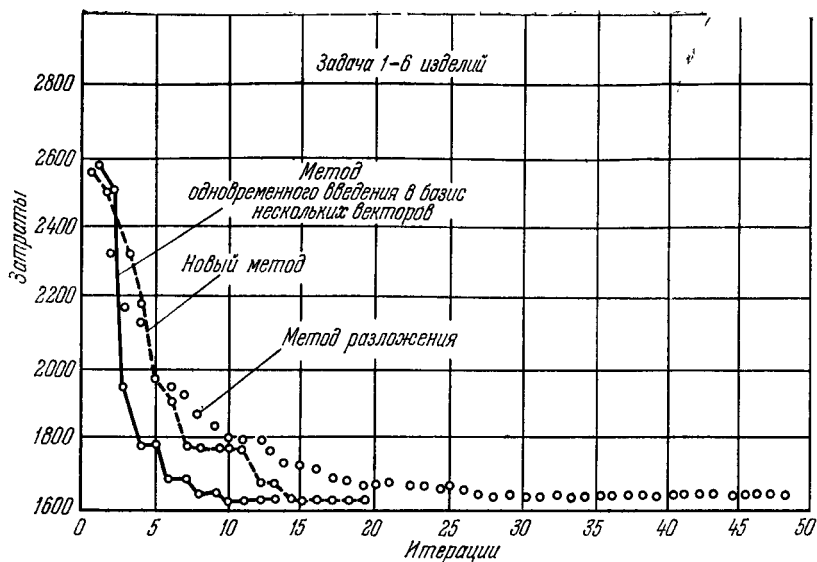


Рис. 3.2. Зависимость затрат от числа итераций в фазе 2.

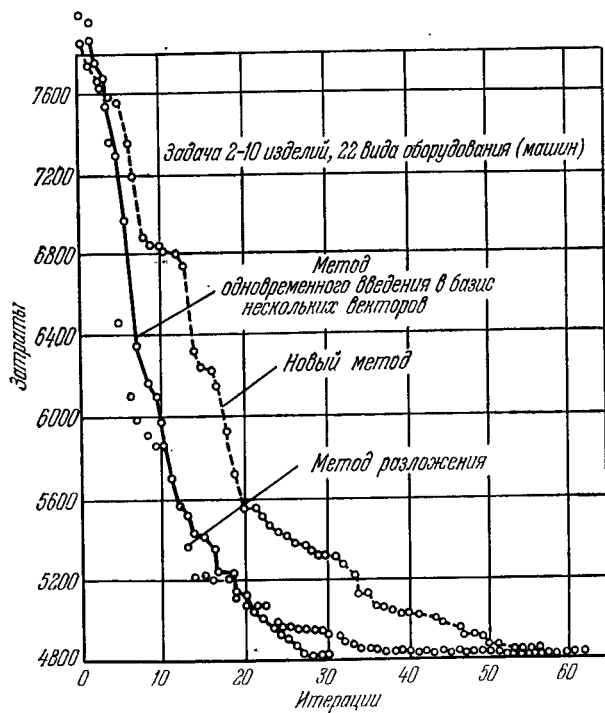


Рис. 3.3. Зависимость затрат от числа итераций в фазе 2.

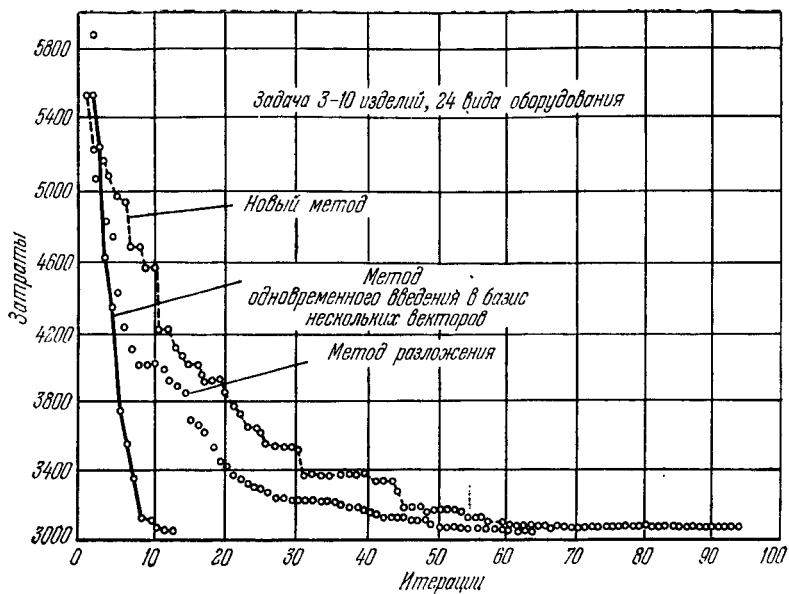


Рис. 3.4. Зависимость затрат от числа итераций в фазе 2.

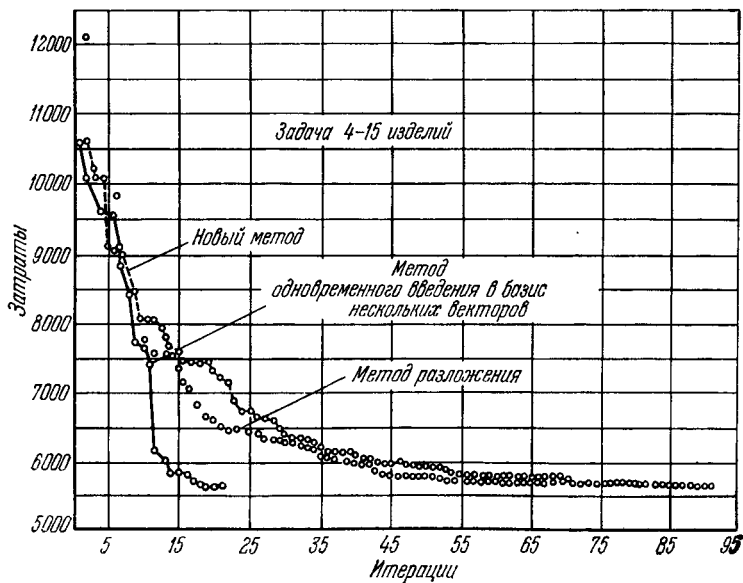


Рис. 3.5. Зависимость затрат от числа итераций в фазе 2.

расчета оценки было на два порядка меньше, чем время решения самих подзадач, было принято, что в методе 3 число итераций увеличивалось на единицу только когда производилось «перерешивание» подзадач. С учетом этого обстоятельства, можно считать, что для решавшихся здесь проблем время вычислений приблизительно пропорционально числу итераций во всех трех методах.

Из таблицы 3.2.1 видно, что при применении приема одновременного введения нескольких векторов решения подзадач, генерированные динамическими программами, использовались многократно. Рис. 3.6 и 3.7 показывают поведение оценки снизу в зависимости от числа проведенных итераций метода 2. Если бы было принято правило остановки после получения оценки, отличающейся от найденных затрат, менее чем на 5%, то решение задач 1—4 по методу 3 закончилось за 16, 16, 36 и 26 итераций соответственно, а для получения строго оптимального решения требовалось 20, 18, 43 и 31 итерация.

Таблица 3.2.1

Использования метода для решения задачи 3—4

Решение подзадач на 2-м этапе		Общее число ведущих операций, выполненных с использованием оценок текущего решения переменных	
Задача 3	Задача 4	Задача 3	Задача 4
1	2	1	5
2	4	10	11
3	6	12	18
4	8	15	21
5	10	19	33
6	12	24	40
7	14	25	44
8	16	26	47
9	18	28	47
10		30	
11		30	

Использовался и ряд других идей для сокращения времени вычислений. Одна из них заключалась в решении подзадач с помощью упрощенного алгоритма динамического программирования, в котором допускалось изменение только одной переменной состояния, а вторая (переменная m_{ii}) фиксировалась на начальном уровне (m_{i0}). Этот прием ускорял процесс более чем вдвое и зачастую приводил к оптимальному решению либо к хорошему приближению. Он использовался в трех задачах на всех начальных итерациях вплоть до остановки из-за выполнения

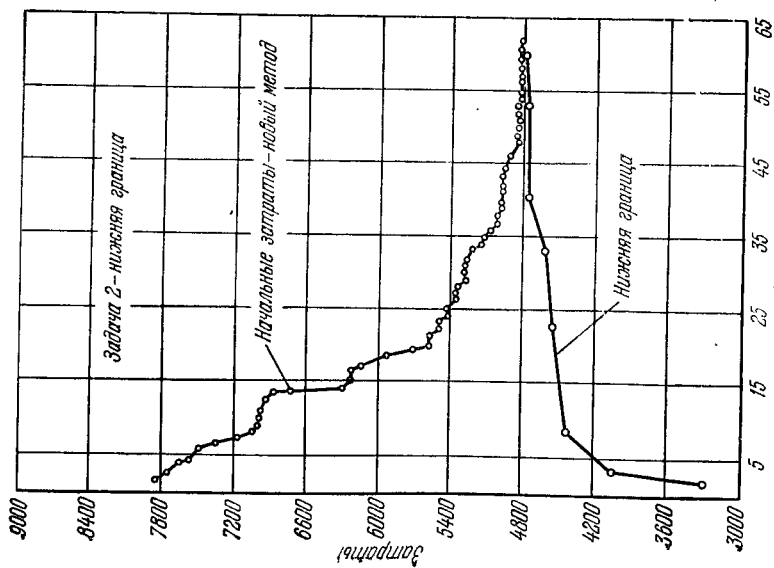


Рис. 3.6. Нижняя граница.

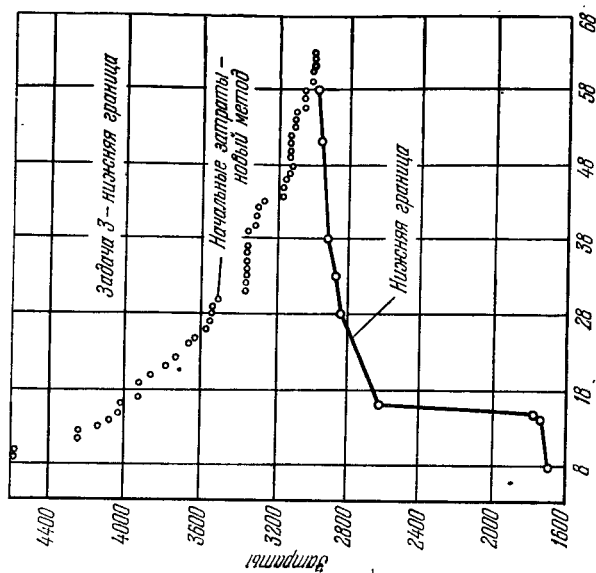


Рис. 3.7. Нижняя граница.

критерия оптимальности. Только после этого производился переход на использование точного динамического алгоритма (с двумя переменными состояниями). Затраты времени в целом сокращались примерно наполовину.

§ 3.3. ЗАДАЧА ОБОБЩЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Вулф (см. [10]) определил обобщенную линейную задачу как задачу линейного программирования, в которой допускается изменение не только показателей интенсивности использования технологических способов x_i , но и соответствующих им столбцов технологической матрицы a_i , а также коэффициентов целевой функции c_i . Эту задачу можно сформулировать так: найти неотрицательные переменные x_i и векторы (a_i, c_i) , обеспечивающие минимум целевой функции

$$z = \sum_i c_i x_i \quad (1)$$

при ограничениях

$$\sum_i a_i x_i = b, \quad x_i \geq 0, \quad (2)$$

$$(a_i, c_i) \in S_i, \quad (3)$$

где S_i предполагается выпуклым ограниченным многогранником. Задача (1) — (3) не является задачей линейного программирования, так как она включает произведения переменных. Однако с помощью принципа разложения Данцига — Вулфа она может быть приведена к линейной задаче.

Так как S_i — ограниченные множества, векторы $(a_i, c_i) \in S_i$ могут быть представлены в виде

$$(a_i, c_i) = \sum_k \lambda_{ik} (a_i^k, c_i^k), \quad \sum_k \lambda_{ik} = 1, \quad \lambda_{ik} \geq 0, \quad (4)$$

где (a_i^k, c_i^k) — крайние точки S_i . Подставив (4) в (1) и (2), получим задачу:

минимизировать

$$z = \sum_{i,k} c_{ik} \lambda_{ik} x_i \quad (5)$$

при ограничениях

$$\sum_{i,k} a_{ik} \lambda_{ik} x_i = b. \quad (6)$$

Пусть

$$u_{ik} = \lambda_{ik} x_i \quad (7)$$

Учитывая (1), запишем задачу (5) — (6) следующим образом:
минимизировать

$$\sum_{i,k} c_{ik} u_{ik} \quad (8)$$

при ограничениях

$$\sum_{i,k} a_i^k u_{ik} = b, \quad u_{ik} \geq 0. \quad (9)$$

Пусть теперь $\{u_{ik}^0\}$ — оптимальное решение задачи (8) — (9). Если для заданных i все $u_{ik}^0 = 0$, то из (7) следует, что $x_i = 0$ и (a_i, c_i) могут принимать любые значения. Если для заданных i по крайней мере одна переменная u_{ik}^0 положительна, то

$$\lambda_{ik} = \frac{u_{ik}^0}{x_i^0}, \quad (10)$$

$$\sum_k \lambda_{ik}^0 = \frac{\sum_k u_{ik}^0}{x_i^0}, \quad (11)$$

$$x_i^0 = \sum_k u_{ik}^0, \quad (12)$$

$$(a_i^0, c_i^0) = \sum_k \lambda_{ik}^0 (a_i^k, c_{ik}). \quad (13)$$

Решая задачу (8) — (9), нет необходимости знать заранее все крайние точки (a_i^k, c_{ik}) . Как и раньше, можно воспользоваться методом генерации столбцов.

Определим характеристические разности столбцов матрицы условий задачи (8) — (9):

$$\bar{c}_{ik} = c_{ik} - \pi a_i^k.$$

В базис будет вводиться столбец с наименьшей характеристической разностью. Нахождение такого столбца, очевидно, эквивалентно решению следующей задачи линейного программирования:

минимизировать

$$c_i - \pi a_i$$

при условии

$$(a_i, c_i) \in S_i.$$

Последующие вычисления выполняются в соответствии с процедурой модифицированного симплекс-метода, и приводят, если

не считаться с возможностью заиклинивания, к решению за конечное число итераций. Изложенный подход применим и для более общего случая, когда S_i — выпуклые множества (не обязательно многогранники), что в свою очередь, как доказал Данциг [10], позволяет представить задачу выпуклого программирования как задачу обобщенного линейного программирования.

Рассмотрим задачу:

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

где $g_i(x)$ — выпуклые функции. Введя новые переменные $(y_0, y_1, \dots, y_m)$, получим следующую эквивалентную ей задачу: минимизировать y_0 при ограничениях

$$f(x) \leq y_0, \quad g_i(x) \leq y_i \leq 0, \quad i = 1, \dots, m.$$

В свою очередь эта задача может быть преобразована в задачу обобщенного линейного программирования с единственным столбцом $(y_0, y) = (y_0, y_1, \dots, y_m)$ и переменным $\lambda, \lambda_1, \dots, \lambda_m$: минимизировать z при ограничениях

$$\left. \begin{array}{rcl} \lambda & & = 1, \\ y_1 \lambda + \lambda_1 & & = 0, \\ y_2 \lambda + & \lambda_2 & = 0 \\ \vdots & & \vdots \\ y_m \lambda + & & + \lambda_m = 0, \\ y_0 \lambda & & = z, \end{array} \right\} \quad (14)$$

$$\lambda_i \geq 0, \quad i = 1, \dots, m,$$

где (y_0, y_i) принадлежит множеству

$$R = \{(y_0, y) \mid y_0 \geq f(x), y \geq g(x) \text{ для некоторых } x\}. \quad (15)$$

Это множество можно представить как совокупность всех точек пространства E^{m+1} , принадлежащих графику или расположенных над графиком функции

$$\omega(y) = \min \{f(x) \mid g(x) \leq y\}.$$

Нетрудно показать (см. упражнение 12), что если f и g_i — выпуклые функции, то R — выпуклое множество и ω — выпуклая функция. Пусть выбраны точки *) $(f(x_i), g(x_i)) \in R$.

Так как R — выпуклое множество, то выпуклая оболочка этих точек принадлежит R . Сведение рассматриваемой задачи к обобщенной задаче линейного программирования может быть

*) Метод нахождения таких точек будет рассмотрен в § 3.4.

осуществлено путем аппроксимации R его так называемой внутренней линейризацией $IL(R)$, представляющей собой совокупность всех точек внутри выпуклой оболочки, а также выше и справа от нее

$$IL(R) = \left\{ (y_0, y) \mid (y_0, y) \geq \sum_i \lambda_i (f(x_i), g(x_i)), \sum_i \lambda_i = 1, \lambda_i \geq 0 \right\}, \quad (16)$$

что иллюстрируется рис. 3.8.

Так как y_0 — целевая функция, то, решая (14) относительно $(y_0, y) \in IL(R)$, получим оптимальное значение y_0 , равное $\sum_i \lambda_i f(x_i)$. Теперь целевая функция будет содержать только переменные λ_i . Поэтому задачу (14) можно представить в виде следующей эквивалентной ей задачи линейного программирования:

минимизировать

$$\sum_i \lambda_i f(x_i)$$

при ограничениях

$$\sum_i \lambda_i g_i(x_i) \leq 0, \quad i = 1, \dots, m,$$

$$\sum_i \lambda_i = 1, \quad \lambda_i \geq 0.$$

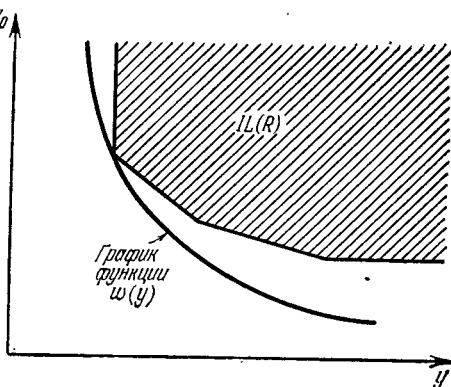


Рис. 3.8. Внутренняя линейризация.

Основная трудность, возникающая при таком подходе, заключается в том,

что, независимо от выбора точек $\{x_i\}$, в общем случае $IL(R)$ не совпадает с множеством R (совпадение возможно только в случае, когда R — выпуклый многогранник). Вследствие этого решение задачи выпуклого многогранника путем сведения последней к обобщенной задаче линейного программирования, как правило, бесконечный, но сходящийся процесс. Этот вопрос более подробно будет проанализирован в § 3.4. Различные способы линейризации множества R рассматриваются также в главах VII и VIII.

Задачи принятия решений и цепь Маркова. Задачи последовательного принятия решений часто возникают при анализе системы, которая может находиться в одном из конечного числа состояний и переход из одного состояния в другое осуществляется случайным образом (стохастически). Примером подобной

ситуации может служить однопродуктовая задача управления запасами в форме, рассмотренной Манном [11]. Предположим, что спрос на продукцию в каждый период времени является случайной величиной, одинаково распределенной для всех периодов времени. Предполагается, далее, независимость объемов спроса для различных периодов. Объем спроса может принимать конечное число значений $(0, 1, \dots, D)$. Количество заказываемой и выпускаемой продукции в каждом периоде является также дискретной величиной, принимающей значения $(0, 1, \dots, M)$.

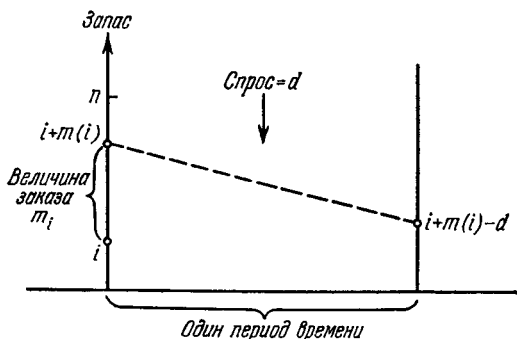


Рис. 3.9. Изменение запаса во времени.

Предположим также, что неудовлетворенный спрос не задерживается и учитывается в потере объема реализации соответствующего периода, а величина запаса для каждого периода ограничена сверху. Поэтому если начальный запас — неотрицательное целое число, не превосходящее или равное n , то величина запаса продукции для каждого периода будет принимать дискретные значения $(0, 1, 2, \dots, n)$. Правило принятия решения или стационарная политика запаса является функцией $m(i)$, определяющей величину заказа на поставку продукции, когда величина запаса продукции в начале периода равна n единицам. Предполагается, что поставка заказанной продукции осуществляется мгновенно. Поэтому изменения величины запаса для каждого периода времени можно представить в виде графика рис. 3.9.

Правило принятия решения выбирается так, чтобы для любого i

$$i + m(i) \leq n. \quad (17)$$

Рисунок 3.9 показывает, что запас продукции в конце периода будет равен величине

$$j = \begin{cases} i + m(i) - d, & 0 \leq i + m(i) - d \leq n, \\ 0, & i + m(i) - d \leq 0. \end{cases} \quad (18)$$

Очевидно, что условная вероятность того, что конечный запас продукции будет равен j единицам, зависит от истории процесса только через величину начального запаса i . Таким образом, рассматриваемая нами задача запасаания обладает свойствами конечной цепи Маркова, в которой различным состояниям процесса соответствуют уровни запасов $(0, 1, \dots, n)$.

Переходные вероятности для этого процесса определяются следующим образом:

$$p_{ij} = \text{Pr} [\text{конечный запас} = j \mid \text{начальный запас} = i]. \quad (19)$$

Для заданного правила принятия решения $m(i)$ эти вероятности могут быть определены с использованием функции распределения величины спроса

$$p_{ij} = \begin{cases} \text{Pr} [d = i + m(i) - j], & 0 < j \leq n, \\ \text{Pr} [d \geq i + m(i)], & j = 0. \end{cases} \quad (20)$$

Будем предполагать, что изучаемый процесс достигает стационарного режима («статистического равновесия»), т. е. если система функционирует достаточно долго, то вероятность оказаться в состоянии i не зависит от времени и начальных условий. Обозначим эту вероятность через x_i :

$$\lim_{t \rightarrow \infty} \text{Pr} [\text{состояние} = i] = x_i. \quad (21)$$

Будем далее учитывать три вида затрат за период: затраты на хранение (если запас положительная величина) $c_h(i)$, затраты на осуществление заказа $c_o(m(i))$, и наконец, штраф за неудовлетворенный спрос $c_p(d - i - m(i))$. Функция штрафа принимает положительное значение при положительном значении аргумента, а во всех остальных случаях она равна нулю.

Используя стационарные вероятности x_i , запишем выражение для ожидаемых суммарных затрат за период в стационарном режиме:

$$E[c] = \sum_{i=1}^n x_i \left\{ c_h(i) + c_o(m(i)) + \sum_r \text{Pr} [d = r] c_p(r - i - m(i)) \right\}. \quad (22)$$

Теперь задачу принятия решения можно представить как задачу нахождения правила принятия решения $m(i)$, обеспечивающего минимум функции (22). Такую задачу легко истолковать и как задачу выбора конечного множества возможных переходных вероятностей для цепи Маркова. Действительно, из (20) следует, что если задана функция распределения величины спроса, то каждый выбор правила принятия решения $m(0)$ определяет переходные вероятности p_{ij} .

В качестве примера рассмотрим процесс со следующим распределением вероятностей спроса:

$$\Pr[d=r] = \frac{1}{3}, \quad r=0, 1, 2. \quad (23)$$

Размер заказа равен или 0 или 1, т. е.

$$m(i) \in [0, 1] \quad (24)$$

для всех i . Верхняя граница величины запаса равна 2 и, таким образом,

$$i \in [0, 1, 2]. \quad (25)$$

Случай 1: $i=0$. Если $m(0)=0$ и $i=0$, то независимо от величины спроса конечный запас будет равен нулю. Переходные вероятности для $i=0$ при использовании правила принятия решения $m(0)=0$ равны

$$p_{00}^0 = 1, \quad p_{01}^0 = 0, \quad p_{02}^0 = 0. \quad (26)$$

Эти вероятности удобно представить в виде вектора

$$p_0^0 = (1, 0, 0). \quad (27)$$

Если $m(0)=1$, то конечный запас равен нулю при спросе, равном одной или двум единицам продукции, и равен единице, если $d=0$.

Таким образом,

$$p_0^1 = \left(\frac{2}{3}, \frac{1}{3}, 0 \right). \quad (28)$$

Из этого следует, что для состояния $i=0$ существует конечное множество векторов переходных вероятностей, по одному для каждого значения функции $m(0)$:

$$S_0 = \left\{ \overbrace{(1, 0, 0)}^{m(0)=0}, \overbrace{\left(\frac{2}{3}, \frac{1}{3}, 0 \right)}^{m(0)=1} \right\}. \quad (29)$$

Случай 2, 3: $i=1, 2$. Предоставим читателю возможность проверить утверждение, что множества векторов переходных вероятностей можно записать так:

$$S_1 = \left\{ \overbrace{\left(\frac{2}{3}, \frac{1}{3}, 0 \right)}^{m(0)=0}, \overbrace{\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right)}^{m(1)=1} \right\}, \quad (30)$$

$$S_2 = \left\{ \overbrace{\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right)}^{m(2)=0} \right\}. \quad (31)$$

Выбрав по одному вектору из множеств S_0 , S_1 и S_2 , можно, как будет показано ниже, найти предельные стационарные вероятности и определить значение функции (22). Если осуществлен выбор векторов переходных вероятностей, обеспечивающий минимум функции ожидаемых затрат, то оптимальное правило принятия решений $m(\cdot)$ определяется в результате нахождения значений $m(i)$, порождающих выбранные векторы. Конечно, задачи такого типа могут возникнуть не только в системах управления запасами. Данциг и Вулф [12] формулируют общую задачу следующим образом.

Пусть p_i — вектор переходных вероятностей из состояния i

$$p_i = (p_{i1}, p_{i2}, \dots, p_{in}), \quad i = 1, 2, \dots, n, \quad (32)$$

где

$$p_{ij} \geq 0, \quad \sum_{j=1}^n p_{ij} = 1. \quad (33)$$

Вектор p_i может выбираться из множества S_i : $p_i \in S_i$. Если выбран вектор p_i , то ему соответствуют затраты $c_i(p_i)$. Предполагается, что выполняется только одно из следующих двух условий:

(а) S_i — конечное множество и c_i — произвольная функция, определенная на множестве S_i ;

(б) S_i — выпуклый ограниченный многогранник, определяемый конечным множеством линейных равенств и неравенств. Каждый элемент многогранника является распределением. Элементу $p_i \in S_i$ соответствуют затраты $c_i(p)$, являющиеся линейной функцией, заданной на S_i .

Для лучшего понимания связи между условиями (а) и (б) введем понятие *смешанных стратегий*. Смешанной стратегией будет называть множество распределений вероятностей (по одному для каждого состояния), где k -й элемент в распределении i — вероятность того, что решение k осуществляется тогда, когда текущим состоянием является состояние i . Таким образом, смешанная стратегия состоит в выборе из чистых стратегий с заданными вероятностями.

Обозначим эти вероятности λ_{ik} :

$$\lambda_{ik} = \text{Pr} [\text{решение} = k \mid \text{состояние} = i], \quad (34)$$

где

$$\sum_k \lambda_{ik} = 1, \quad \lambda_{ik} \geq 0. \quad (35)$$

Теперь рассмотрим множества S_i , фигурирующие в условии (а). Пусть p_{ij}^k — вероятность перехода $i \rightarrow j$, когда принято решение k , а p_j^k — вектор с компонентами p_{ij}^k ($j = 1, \dots, n$).

При использовании смешанных стратегий могут быть получены векторы переходных вероятностей, отличные от векторов $p_i^k \in S_i$. Предположим, что рассматриваемая система находится в состоянии i . Переходная вероятность p_{ij} равна

$$p_{ij} = \sum_k \lambda_{ik} p_{ij}^k, \quad (36)$$

или, в векторной форме,

$$p_i = \sum_k \lambda_{ik} p_i^k, \quad p_i^k \in S_i. \quad (37)$$

В качестве элемента p_i , задаваемого в формуле (37), может быть любой вектор, принадлежащий выпуклой оболочке $C(S_i)$ множества S_i , фигурирующего в случае (а), причем как и в случае (б) $C(S_i)$ — выпуклый ограниченный многогранник. Таким образом, допуская использование смешанных стратегий, можем получить любое распределение из множеств $C(S_i)$. Хотя условия (а) и (б) во многом совпадают, методы обобщенного линейного программирования непосредственно могут быть использованы лишь в случае (б). Однако будет просто показано, что по этой схеме можно рассматривать оба случая, поскольку задача с условием (б), трактуемая как обобщенная задача линейного программирования, будет иметь оптимальное решение, в котором каждому i соответствует единственный элемент множества S_i , являющийся крайней точкой. Это означает, что если даже допускается использование смешанных стратегий, необходимо учитывать только чистые стратегии (правила принятия решения).

Однако продолжим формулировку общей задачи. Пусть x — стационарное распределение для цепи Маркова

$$x = (x_1, \dots, x_n), \quad (38)$$

где x_i — предельная вероятность для состояния i . Чтобы получить соотношения, задающие это распределение, заметим следующее: вероятность сложного события, состоящего в том, что система находится (с вероятностью x_j) в состоянии j и переходит в состояние i (с вероятностью p_{ji}) равна p_{ji} . Так как j может соответствовать любому состоянию, то x_i является суммой таких вероятностей:

$$x_i = \sum_j x_j p_{ji} \quad (39)$$

или, в векторной форме

$$x = \sum_j x_j p_j. \quad (40)$$

Определим ожидаемые затраты одного периода для стационарных условий:

$$\sum_i c_i(p_i) x_i. \quad (41)$$

Теперь можно сформулировать общую задачу: найти

$$p_i \in S_i \quad (i = 1, \dots, n),$$

минимизирующие (41) при условиях (40) и

$$\sum_i x_i = 1, \quad x_i \geq 0. \quad (42)$$

В уравнении (40) неизвестные содержатся в левой и правой частях. Перенеся в этом уравнении неизвестные в одну сторону, представим сформулированную задачу в таком виде:

минимизировать

$$z = \sum_i \hat{c}_i(v_i) x_i \quad (43)$$

при ограничениях

$$\sum_i x_i = 1, \quad (44)$$

$$\sum_i v_i x_i = 0, \quad (45)$$

$$x_i \geq 0; \quad v_i \in T_i, \quad (46)$$

где T_i — множество всех векторов

$$v_i = (p_{i1}, p_{i2}, \dots, p_{i,i-1}, \dots, p_{in}), \quad (47)$$

таких, что

$$(p_{i1}, \dots, p_{in}) \in S_i$$

и

$$\hat{c}_i(v_i) = c_i(p_i). \quad (48)$$

Задача (43) — (46) является обобщенной задачей линейного программирования, если выполнено условие (b). (Задача с учетом условия (a) будет рассмотрена ниже.)

Заметим, что T_i — многогранник, если многогранником является S_i . При этом крайние точки S_i соответствуют крайним точкам T_i в силу (47). Любую точку $v_i \in T_i$ можно представить в виде выпуклой комбинации крайних точек v_i^k :

$$v_i = \sum_k \lambda_{ik} v_i^k, \quad \lambda_{ik} \geq 0, \quad \sum_k \lambda_{ik} = 1. \quad (49)$$

Подставив (49) в (43), получим

$$z = \sum_i \hat{c}_i \left(\sum_k \lambda_{ik} v_i^k \right) x_i. \quad (50)$$

Так как для случая (b) c_i — линейная функция, то

$$\hat{c}_i \left(\sum_k \lambda_{ik} v_i^k \right) = \sum_k \lambda_{ik} \hat{c}_i(v_i^k). \quad (51)$$

Поэтому (50) можно записать так:

$$z = \sum_{i,k} \hat{c}_i(v_i^k) \lambda_{ik} x_i. \quad (52)$$

Умножив (44) на $\sum_k \lambda_{ik}$, получим

$$\sum_{i,k} \lambda_{ik} x_i = 1. \quad (53)$$

Подставив теперь (49) в (45) и объединив полученный результат с условиями (52) и (53), получим следующую задачу линейного программирования:

минимизировать

$$z = \sum_{i,k} c_{ik} u_{ik}, \quad (54)$$

$$\sum_{i,k} u_{ik} = 1, \quad (55)$$

$$\sum_{i,k} v_{ik} u_{ik} = 0, \quad (56)$$

$$u_{ik} \geq 0, \quad (57)$$

где

$$u_{ik} = \lambda_{ik} x_i, \quad c_{ik} = \hat{c}_i(v_i^k). \quad (58)$$

Как показал Манн [11], эта задача также имеет интерпретацию в терминах смешанных стратегий. Если λ_{ik} интерпретировать согласно (34), то

$$u_{ik} = \text{Pr} [\text{решение} = k | \text{состояние} = i] \text{Pr} [\text{состояние} = i], \quad (59)$$

$$u_{ik} = \text{Pr} [\text{состояние} = i, \text{ решение} = k], \quad (60)$$

т. е. u_{ik} — совместная вероятность того, что система находится в

состоянии i и принимается решение k . По определению, эти вероятности должны удовлетворять условиям (55) и (57). Условие (56) — уравнение стационарного состояния относительно переменных u_{ik} . Действительно, требование стационарности заключается в том, что вероятности пребывания в состоянии j должны совпадать в начале и конце периода. В начале периода эта вероятность равна $\sum_k u_{jk}$; в конце периода она равна $\sum_{i,k} u_{ik} p_{ij}^k$, где p_{ij}^k — j -я компонента вектора p_i^k . Из равенства этих вероятностей следует

$$\sum_{i,k} u_{ik} p_{ij}^k = \sum_k u_{jk} \text{ для всех } j, \quad (61)$$

что эквивалентно условию (56). Таким образом, задачу (54) — (57) можно сформулировать и непосредственно, допуская возможность использования смешанных стратегий.

Для того чтобы сформулировать задачу в виде (54) — (57) для случая, когда S_i и, следовательно, T_i — конечные множества (условие (а)), будем рассматривать векторы v_i^k в качестве элементов множества T_i . В этом случае, чтобы показать эквивалентность условий (54) — (57) и (43) — (46), необходимо доказать существование оптимального решения задачи (54) — (57), обладающего свойством, что для каждого i не более чем одна переменная u_{ik} будет положительна (12). Если такое решение существует, то оптимальный план задачи (43) — (46) определяется следующим образом:

$$\begin{aligned} x_i &= u_{ik}, \quad v_i = v_i^k, \\ x_i &= 0 \quad v_i \end{aligned}$$

если $u_{ik} > 0$ для некоторых k ; принимает произвольное значение на множестве S_i , если $u_{ik} = 0$ для всех k .

Если учитывается условие (b), то ситуация, когда не больше чем одна точка v_i^k выбирается для каждого i , соответствует оптимальному решению задачи (54) — (57) с использованием только чистых стратегий.

Теорема. *Существует оптимальное решение задачи (54) — (57) такое, что для каждого i не больше чем одна переменная u_{ik} будет положительной.*

Доказательство. Любая задача линейного программирования с конечным оптимальным решением имеет базисное допустимое решение, которое является оптимальным. Пусть такое решение для задачи (54) — (57) имеет $r \leq n+1$ положительных компонент и пусть $\bar{B}$ — подматрица размерности $((n+1) \times r)$ базисной матрицы, соответствующая этим компонентам. Так как

базисная матрица является неособенной, то ранг матрицы B равен r . Матрица условий задачи (55) — (56) показана в таблице 3.3.1.

Пусть s — число строк матрицы $\bar{B}$, содержащих по крайней мере один элемент вида $p_{ii}^k - 1$. За исключением первой строки, $n-s$ оставшихся строк имеют неотрицательные элементы, и этим строкам соответствуют нули в правых частях уравнений. Поэтому элементы этих строк заведомо должны быть равны нулю, и в $\bar{B}$ содержится не более $s+1$ ненулевых строк. Не более s из них являются линейно независимыми (так как сумма всех строк,

Таблица 3.3.1

$u_{11}, \dots, u_{1m_1}$	$u_{21}, \dots, u_{2m_2}$	...	$u_{n1}, \dots, u_{nm_n}$	Вектор констант
1...1	1...1		1...1	1
$p_{11}^1 - 1 \dots p_{11}^{m_1} - 1$	$p_{21}^1 \dots p_{21}^{m_2}$	...	$p_{n1}^1 \dots p_{n1}^{m_n}$	0
$p_{12}^1 \dots p_{12}^{m_1}$	$p_{21}^1 - 1 \dots p_{22}^{m_2} - 1$			0
.	.		.	.
.	.		.	.
.	.		.	.
$p_{1n}^1 \dots p_{1n}^{m_1}$	$p_{2n}^1 \dots p_{2n}^{m_2}$	...	$p_{nn}^1 - 1 \dots p_{nn}^{m_n} - 1$	0

исключая первую, равна нулю), поэтому ранг матрицы B равен или не превосходит число s :

$$\text{rank } (\bar{B}) = r \leq s. \quad (62)$$

Рассмотрим теперь структуру столбцов таблицы 3.3.1. Каждый из них содержит по одному элементу $p_{ii}^k - 1$, так что в r столбцах невозможно разместить $s > r$ таких элементов, следовательно, $s = r$. Поэтому элементы столбцов, равные $p_{ii}^k - 1$, должны находиться в разных строках таблицы 3.3.1. Из этих рассуждений следует, что столбцы матрицы B соответствуют переменным u_{ik} с различными индексами i . Таким образом, теорема доказана.

Чтобы решить задачу (54) — (57) с использованием метода генерации столбцов, предположим, что найдено допустимое базисное решение этой задачи. Пусть (π_0, π) — двойственные переменные, соответствующие найденному решению, где π_0 — оценка ограничения (55) и π — оценка ограничения (56). Найдем

теперь с помощью этих оценок характеристические разности столбцов и запишем подзадачу:

минимизировать

$$c_i(p_i) - \pi p_i \quad (63)$$

при ограничениях

$$p_i \in S_i. \quad (64)$$

Если S_i — конечное множество и c_i — произвольные функции, то задача (63) — (64) является задачей дискретного программирования. Когда S_i имеет большое число элементов, эту задачу можно решить методом ветвей и границ. Если область определения c_i может быть расширена до выпуклой оболочки $C_i(S_i)$ точек S_i и если при таком распространении c_i вогнуты на $C(S_i)$, то функция (63) будет достигать своего минимального значения в крайней точке $C_i(S_i)$. Поэтому все оставшиеся точки могут не рассматриваться.

Если S_i — выпуклый многогранник и c_i — линейные функции, то задача может быть решена методами линейного программирования. В этом случае в соответствии с теоремой оптимальному плану будут соответствовать крайние точки S_i — по одной точке для каждого из множеств S_i .

§ 3.4 ЛИНЕАРИЗАЦИЯ НА СЕТКЕ И НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

3.4.1. Общее изложение

Поскольку линейное программирование является хорошо развитым и эффективным средством решения задач, естественно попытаться использовать его методы в нелинейных задачах с помощью линеаризации. Одна из возможностей выполнения этого состоит в том, что Вулф назвал линеаризацией на сетке. Пусть нелинейная задача имеет вид

$$\min \{f(x)/g_i(x) \leq 0, i = 1, \dots, m\}. \quad (1)$$

Выберем множество узлов сетки $x_1, \dots, x_T$. Выпуклая оболочка этого множества имеет элементы вида

$$x = \sum_i \lambda_i x_i, \quad \lambda_i \geq 0, \quad \sum_i \lambda_i = 1. \quad (2)$$

Предположим сначала, что x — скаляр. Тогда функция f может быть линеаризована на сетке проведением ее кусочно-линейной аппроксимации по ординатам узловых точек так, как показано на рис. 3.10. Любая точка f на жирных линиях может

быть представлена в виде

$$f = \sum_i \lambda_i f(x_i). \quad (3)$$

К несчастью, другие точки так же представимы в виде (3), а именно, все точки из выпуклой оболочки множества $\{f(x_i), x_i\}$, заштрихованные на рис. 3.10.

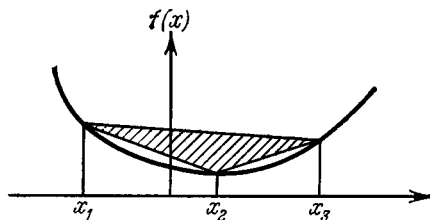


Рис. 3.10. Кусочно-линейная аппроксимация.

Однако только точки на жирных линиях составляют полезную интерполяцию между узловыми точками. Для того чтобы получить только их, необходимо ограничить выбор λ_i так, чтобы в решении самое большее два из них были положительными, и чтобы положительные λ_i имели соседние индексы. (Установление этого

критерия для многомерного случая рассматривается ниже.) Будет показано, что если f и g_i выпуклы, предлагаемый метод всегда приводит к правильной интерполяции.

Замена f и g_i их линейризацией на сетке приводит к задаче линейного программирования относительно переменных λ_i :

$$\min \left\{ z = \sum_{i=1}^T \lambda_i f(x_i) \mid \sum_{i=1}^T \lambda_i g_i(x_i) \leq 0, i = 1, \dots, m; \sum_{i=1}^T \lambda_i = 1; \lambda_i \geq 0 \right\}. \quad (4)$$

Вообще говоря, для получения точной аппроксимации нелинейных функций f, g_i требуется много узловых точек. Каждая узловая точка соответствует столбцу в (4), и было бы расточительно выбирать эти точки заранее и вычислять коэффициенты соответствующих столбцов. Желательно было бы уточнять аппроксимацию только в окрестности оптимума, и это может быть выполнено путем использования самого симплекс-метода для выбора новых узлов сетки при решении соответствующих вспомогательных задач.

Предположим, что в начале итерации k введена сетка с узловыми точками $x_1, \dots, x_T$. Задача (4) тогда может быть решена и дает оптимальное значение $\{\lambda_i^k\}$ и двойственных переменных $\{\pi^k, \pi_0^k\}$. Текущая аппроксимация оптимального решения исходной задачи (1) такова:

$$x^k = \sum_i \lambda_i^k x_i.$$

Из всех новых узловых точек, которые могли бы быть введены для улучшения этой аппроксимации, симплекс-метод естественно выбрал бы такую, которой соответствует столбец с наименьшей характеристической разностью, т. е. точку x_i , которая минимизирует

$$\bar{c}_i = f(x_i) - \sum_{i=1}^m \pi_i^k g_i(x_i) - \pi_0^k.$$

Вычисление такой точки сводится к решению нелинейной вспомогательной задачи без ограничений:

$$\min \left\{ f(x) - \sum_i \pi_i^k g_i(x) \right\}. \quad (5)$$

Эта задача может быть решена любым соответствующим методом. Если

$$\min \left\{ f(x) - \sum_i \pi_i^k g_i(x) \right\} - \pi_0^k < 0, \quad (6)$$

то, за исключением случаев вырожденности, целевая функция в (4) может быть уменьшена при введении нового столбца в базис. Если указанный минимум достигается в точке $\hat{x}^k$, то задача (4) дополняется новым столбцом

$$p^k = (g_1(\hat{x}^k), \dots, g_m(\hat{x}^k), 1)$$

и в целевую функцию вводится слагаемое $f(\hat{x}^k)\lambda_{T+1}$. После этого задача (4) решается заново, и начинается новый цикл. Заметим, что (4) играет роль сокращенной координирующей задачи в методе декомпозиции Данцига — Вулфа (см. § 2.4). Если целевая функция и ограничения в (1) линейны и определены на выпуклом многограннике, то любая точка внутри него может быть представлена в виде выпуклой комбинации крайних точек, образующих конечную сетку, и аппроксимация типа (3) может быть сделана точной. Процесс линеаризации на сетке тогда превращается в алгоритм декомпозиции Данцига — Вулфа для задачи линейного программирования. В этом случае сохранение всех ранее генерированных столбцов в координирующей задаче необязательно. В нелинейном же случае доказательства сходимости, которое в настоящее время существует [16], требует, чтобы сохранялись все столбцы.

Дадим разъясняющую геометрическую интерпретацию этой процедуры в терминах множества R , задаваемого уравнением (3.3-15). Пусть λ^k и $u^k = -(\pi_0^k, \pi^k)$ — решения задачи (4) и ей двойственной на итерации k . Вектор (λ^k, u^k) является тогда

седловой точкой функции Лагранжа, связанной с (4) (см. гл. VII). Как показано в теореме 8.3.3, π^k является субградиентом *) функции

$$\tilde{w}(y) = \min \left\{ \sum_i \lambda_i f(x_i) / \sum_i \lambda_i g(x_i) \leq y, \sum_i \lambda_i = 1, \lambda_i \geq 0 \right\}$$

в точке $\bar{y} = \sum_i \lambda_i^k g(x_i)$. Иначе говоря, π^k является вектором, составленным из коэффициентов опорной гиперплоскости к графику $\tilde{w}$ в $\bar{y}$. Так как этот график является границей множества $IL(R)$ в (3.3-16), то ситуация соответствует показанной на рис. 3.11. В § 7.4 будет показано, что если $\hat{x}^k$ — решение задачи (5), то $(f(\hat{x}^k), g(\hat{x}^k))$ является точкой контакта множества R с опорной гиперплоскостью с коэффициентами π^k . Вектор $f(\hat{x}^k)$, $g(\hat{x}^k)$, таким образом, лежит на границе R . Добавление нового

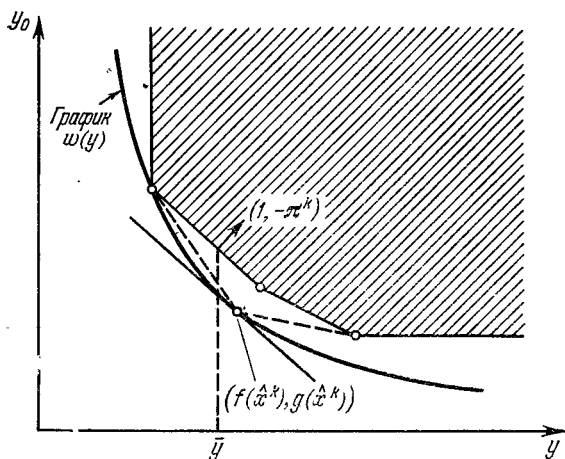


Рис. 3.11. Улучшенная линейзация R .

столбца эквивалентно образованию улучшенной внутренней линейзации, как показано пунктирными линиями на рис. 3.11. Если текущее решение близко к оптимальному, то при использовании π^k в качестве коэффициентов наклона гиперплоскости, получаем аппроксимацию R , улучшенную именно там, где это наиболее необходимо, в окрестности оптимума.

Процедура имеет ряд желательных свойств, если f , g являются выпуклыми. Во-первых, точки x^k являются допустимыми,

*) См. Приложение 2.

так как

$$g_i(x^k) \leq \sum_i \lambda_i^k g_i(x_i) \leq 0.$$

Во-вторых, если $\pi_i \leq 0$, $i=1, \dots, m$, то функция цели в (5) является выпуклой, и любой локальный минимум вспомогательной задачи является и глобальным. Условие $\pi_i \leq 0$ может быть гарантировано при минимизации z только по дополнительным переменным в (4), прежде чем найдена новая точка сетки. Это имеет место, поскольку соответствующие дополнительным переменным характеристические разности равны $-\pi_i$, а условие оптимальности приводит к тому, что эти величины неотрицательны. Наконец, могут быть оценены верхние и нижние границы для минимума f . Пусть

$$S = \{x/g_i(x) \leq 0, i = 1, \dots, m\}.$$

Так как f выпукла и x^* допустимы,

$$\min_{x \in S} f(x) \leq f(x^k) \leq \sum_i \lambda_i^k f(x_i) = z^k, \quad (7)$$

что и дает верхнюю оценку. Если $\pi_i \leq 0$, $i=1, \dots, m$, то минимум, достигаемый в (6), всегда является нижней границей для минимума f . Чтобы убедиться в этом, положим $u_i = -\pi_i$, $i=1, \dots, m$, и введем

$$L(x, u) = f(x) + \sum_i u_i g_i(x), u_i \geq 0. \quad (8)$$

По определению,

$$\min_{x \in E^n} L(x, u) \leq L(x, u), \quad (9)$$

что справедливо для всех x . Однако для всех x , удовлетворяющих $g_i \leq 0$, сумма в (8) неотрицательна и (9) приобретает вид

$$\min_{x \in E^n} L(x, u) \leq f(x), \text{ для } x \in S. \quad (10)$$

Таким образом,

$$\min_{x \in E^n} L(x, u) \leq \min_{x \in S} f(x), \quad (11)$$

что и дает нижнюю границу. Заметим, что оценка (11) справедлива даже, если функции f и g_i не являются выпуклыми. Функция Лагранжа $L(x, u)$ играет центральную роль в математическом программировании. Ее использование в нелинейной процедуре декомпозиции обсуждается в главе VII. Там показано, что минимальное значение в (10), которое рассматривается как функция от u , является целевой функцией в задаче, двойствен-

ной к исходной нелинейной задаче (1). В действительности, условие оптимальности, приводимое ниже, является просто условием того, что оптимальные значения прямой и двойственной задач совпадают.

Если условие (6) не выполняется, и все $\pi_i \leq 0$, то не существует узловой точки или дополнительной переменной, введение которой могло бы привести к уменьшению целевой функции. Текущая аппроксимация является тогда оптимальной для задачи (1), что и показано в следующей теореме.

Теорема 1 (Данциг [16]). Пусть (π^k, π_0^k) являются оценками ограничений, соответствующими оптимальному решению координирующей задачи (4) на итерации k . Если

$$\min \left\{ f(x) - \sum_{i=1}^m \pi_i^k g_i(x) \right\} - \pi_0^k \geq 0,$$

то точка $x^k = \sum_i \lambda_i^k x_i$ является оптимальной, т. е. $\min_{x \in S} f(x) = f(x^k)$.

Доказательство. По определению, характеристические разности для базисных переменных равны нулю:

$$\bar{c}_i^k = f(x_i) + \sum_i u_i^k g_i(x_i) + u_0^k = 0,$$

где $u_i^k = -\pi_i^k \geq 0$. Умножая эти равенства на λ_i^k и суммируя, получаем

$$\sum_i \lambda_i^k f(x_i) + \sum_i u_i^k \sum_i \lambda_i^k g_i(x_i) + u_0^k = 0.$$

По условиям дополняющей нежесткости имеем

$$\sum_i u_i^k \sum_i \lambda_i^k g_i(x_i) = 0.$$

Таким образом, $u_0^k = -\sum_i \lambda_i^k f(x_i) = -z^k$. Добавим теперь величины

$u_0^k + z^k$, которые равны нулю, к правым частям (11). Комбинируя результат с (7), получаем

$$\min_{x \in E^n} \{L(x, u^k) + u_0^k\} + z^k \leq \min_{x \in S} f(x^k) \leq z^k. \quad (12)$$

По условиям теоремы

$$\min_{x \in E^n} \{L(x, u^k) + u_0^k\} \geq 0,$$

так что указанная величина может быть изъята из левой части

неравенства (12):

$$z^k \leq \min_{x \in S} f(x) \leq f(x^k) \leq z^k.$$

Таким образом, имеем

$$\min_{x \in S} f(x) = f(x^k) = z^k$$

и теорема доказана.

Данциг [16] доказывает, что для выпуклой задачи этот алгоритм сходится к оптимальному решению при стремлении числа итераций к бесконечности. Доказательство вновь требует, чтобы все столбцы, генерированные в прошлых итерациях, сохранялись в каждой новой ведущей задаче.

Сепарабельные задачи. Если f , g_i являются аддитивно-сепарабельными, т. е. если

$$f = \sum_j f_j(x_j), \quad g_i = \sum_j g_{ij}(x_j),$$

то появляются значительные преимущества. Тогда множество узловых точек $\{x_{jt}\}$ может быть определено для каждой переменной x_j и каждая из функций f_j и g_{ij} линеаризована отдельно, так что возникает задача

$$\left. \begin{aligned} \min \sum_{i,t} \lambda_{it} f_i(x_{it}), \\ \sum_{i,t} \lambda_{it} g_{ij}(x_{it}) \leq 0, i = 1, \dots, m; \\ \sum_i \lambda_{it} = 1, j = 1, \dots, n; \lambda_{it} \geq 0. \end{aligned} \right\} \quad (13)$$

Вспомогательные задачи (6) разделяются в этом случае на независимые одномерные задачи

$$\min f_j(x_j) - \sum_i \pi_i g_{ij}(x_j).$$

Поскольку в них нет ограничений, их решение может быть легко получено.

Допустимые интерполяции. Возвращаясь к вопросу о том, как обеспечить, чтобы возникали только истинные точки интерполяции, можно рассмотреть трехмерный случай. Это рассмотрение убедит читателя, что желаемое действительно достигается, а именно, оптимальное решение ведущей задачи (4) должно быть таким, что выпуклая оболочка тех узловых точек x_i , для которых $\lambda_i > 0$, не содержит никаких других узловых точек.

При некоторых дополнительных предположениях это выполняется автоматически, что показывает следующая теорема, принадлежащая Вульф [14].

Теорема 2. Если f и все g_i являются выпуклыми и либо а) f строго выпукла, либо б) ведущая задача (4) имеет единственное решение, то любое оптимальное решение (4) обладает тем свойством, что выпуклая оболочка тех точек x_i , для которых $\lambda_i > 0$, не содержит никаких других узловых точек.

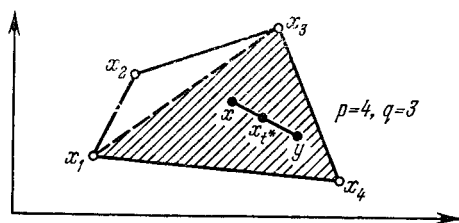


Рис. 3.12. Недопустимая интерполяция (множество H заштриховано).

Доказательство. Пусть x_i , для которых $\lambda_i > 0$, обозначены $x_1, \dots, x_p$. Предположим, что имеется другая узловая точка x_{i^*} , которая содержится в выпуклой оболочке $x_1, \dots, x_p$. Тогда точка x_{i^*} содержится в относительной внутренней *) выпуклой оболочки H некоторого подмножества этих точек, скажем, $x_1, \dots, x_q$, $q \leq p$ (рис. 3.12), т. е.

$$x_{i^*} = \sum_{i=1}^q s_i x_i, \quad s_i > 0, \quad \sum_{i=1}^q s_i = 1. \quad (14)$$

Пусть (π, π_0) являются двойственными переменными, связанными с оптимальным решением задачи (4). Тогда условие оптимальности для этой задачи записывается в виде

$$\bar{c}_i = F(x_i) = f(x_i) - \sum_i \pi_i g_i(x_i) - \pi_0 \geq 0, \quad \text{для всех } i, \quad (15)$$

$$\bar{c}_i = 0, \quad \text{если } \lambda_i > 0, \quad (16)$$

$$\pi_i \leq 0. \quad (17)$$

В силу (15) и (17) $F(x)$ выпукла, так что, преобразуя (16), имеем

$$F\left(\sum_{i=1}^q \beta_i x_i\right) \leq \sum_{i=1}^q \beta_i F(x_i) = 0. \quad (18)$$

Таким образом, $F(x)$ неположительна для всех $x \in H$.

В действительности мы можем показать, что $F(x) = 0$ на H . Предположим, что это не так. Тогда $(F(x) < 0, x \in H)$

*) Понятие относительной внутренней множества (символ $r \text{ int}$) определено в Приложении 1. (Прим. ред.)

и $(x_{t*} \in r \text{ int } H)$ приводит к тому, что существует $y \in H$ такое, что

$$x_{t*} = \alpha x + (1 - \alpha) y, \quad 0 < \alpha < 1, \quad (19)$$

как показано на рис. 3.12. Далее получаем

$$F(x_{t*}) \leq \alpha F(x) + (1 - \alpha) F(y) < 0, \quad (20)$$

что противоречит (15). Таким образом, полагая $F(x)$ нулем для всех $x \in H$, получим

$$f(x) = \sum_i \pi_i g_i(x) + \pi_0, \quad (21)$$

но функция в левой части (21) выпукла, и правая часть вогнута, так как $\pi_i \leq 0$. Единственная возможность состоит в том, что обе они являются линейными на H . Однако, так как H состоит из более чем одной точки, строгая выпуклость f , предполагаемая условиями теоремы, означает, что это невозможно, так что при условии (а) теорема доказана.

Для того чтобы доказать (б), мы покажем, что из существования x_{t*} следует, что может быть сконструировано иное решение задачи (4), $\{\bar{\lambda}_t\}$, что и приведет к противоречию. Пусть

$$\bar{\lambda}_t = \lambda_t - \theta s_t, \quad t = 1, \dots, q, \quad (22)$$

$$\bar{\lambda}_{t*} = \lambda_{t*} + \theta, \quad (23)$$

$$\bar{\lambda}_t = \lambda_t \text{ для всех других } t. \quad (24)$$

Если $\theta > 0$ и достаточно мало, то все $\bar{\lambda}_t \geq 0$ и

$$\sum_{\forall t} \bar{\lambda}_t = \sum_{\forall t} \lambda_t - \theta \sum_{t=1}^q s_t + \theta = 1. \quad (25)$$

Подставляя $\bar{\lambda}_t$ в (4), получаем

$$\sum_i \bar{\lambda}_i f(x_i) = \sum_i \lambda_i f(x_i) + \theta \left(f(x_{t*}) - \sum_{t=1}^q s_t f(x_t) \right), \quad (26)$$

$$\sum_i \bar{\lambda}_i g_i(x_i) = \sum_i \lambda_i g_i(x_i) + \theta \left(g_i(x_{t*}) - \sum_{t=1}^q s_t g_i(x_t) \right). \quad (27)$$

Так как $f(x)$ линейна на H и $x_{t*} \in H$, то

$$f(x_{t*}) = \sum_{t=1}^q s_t f(x_t), \quad (28)$$

так что по функционалу решения $\{\lambda_i\}$ и $\{\bar{\lambda}_i\}$ дают одинаковые результаты. Для того чтобы показать, что $\bar{\lambda}_i$ удовлетворяют ограничениям (4), т. е. делают выражение в (27) неположительным, заметим, что первый член в правой части (27) неположителен, так как λ_i — оптимальное решение задачи (4), и это же справедливо для слагаемого, имеющего множителем θ , в силу выпуклости g_i . Таким образом, (27) неположительная для всех $\theta \geq 0$. Отсюда следует, что для достаточно малых $\theta > 0$, $\bar{\lambda}_i$ образуют другое оптимальное решение задачи (4), а это противоречит (b).

Миллер [15] обобщил эти идеи на случай сепарабельных функций, возможно, не являющихся выпуклыми. Здесь требование, чтобы все интерполяции были правильными, обеспечивается тем, что разрешается переменной λ_{ji} в (13) входить в базис, только если базисными уже являются $\lambda_{j, i-1}$, $\lambda_{j, i+1}$ или никакие другие λ_{ji} не входят в базис. Это ограничение означает, что нельзя обеспечить достижение глобального оптимума даже в линейной задаче (13), однако практически метод позволяет находить локальные решения, являющиеся вполне приемлемыми.

3.4.2. Нелинейный вариант принципа декомпозиции Данцига — Вулфа

Используя линеаризацию на сетке, Секайн [17] обобщил принцип декомпозиции Данцига — Вулфа на случай выпуклых нелинейных подсистем. Решаемая задача такова:

$$\min \{f(x) = \sum_{i=1}^p f_i(x_i)\}, \quad (1)$$

где каждая f_i является выпуклой функцией, причем задан ряд линейных связывающих ограничений

$$\sum_{i=1}^p A_i x_i = b \quad (2)$$

и независимо ряд, возможно, нелинейных подсистемных ограничений относительно каждого x_i :

$$x_i \in S_i, \quad i = 1, \dots, p, \quad (3)$$

где S_i является выпуклыми множествами, необязательно многогранными. Эта формулировка включает линейные задачи в качестве частного случая.

Так как ограничения на каждое x_i независимы, мы можем выбрать отдельную сетку узловых точек для каждого. Пусть i -я

узловая точка для x_i есть

$$x_i^t \in S_i. \quad (4)$$

Каждая функция $f_i(x_i)$ заменяется ее линеаризацией на сетке $\{x_i^t\}$:

$$f_i(x_i) = \sum_t \lambda_i^t f_i(x_i^t), \quad (5)$$

где

$$\sum_t \lambda_i^t = 1, \quad \lambda_i^t \geq 0. \quad (6)$$

Связанные с ней значения x_i таковы:

$$x_i^a = \sum_t \lambda_i^t x_i^t. \quad (7)$$

Для того чтобы получить аппроксимирующую линейную задачу для (1) — (3), каждая f_i в (1) заменяется ее линеаризацией и (7) подставляется в (2), что дает

$$\min \left\{ z = \sum_{i,t} \lambda_i^t f_i(x_i^t) \right\} \quad (8)$$

при ограничениях

$$\sum_{i,t} (A^t x_i^t) \lambda_i^t = b, \quad (9)$$

$$\sum_t \lambda_i^t = 1, \quad i = 1, \dots, p, \quad (10)$$

$$\lambda_i^t \geq 0. \quad (11)$$

Так как $x_i^t \in S_i$ и S_i выпукло, точка x_i^a в (7) также является элементом S_i . Если λ_i^t удовлетворяет (9), то x_i^a также удовлетворяет (2) и, следовательно, является допустим решением исходной задачи. Как и ранее, нежелательно выбирать заранее много узловых точек. Этого вновь можно избежать, используя симплекс-метод для генерации их по мере необходимости. Столбец, связанный с λ_i^t , есть

$$p_i^t = \{f_i(x_i^t) : A_i x_i^t : u_i\}', \quad (12)$$

где u_i — i -й единичный вектор. Предположим, что допустимый базис в (8) — (11) имеется, и пусть (π_i, π_2) являются двойственными переменными, связанными с этим базисом. Тогда характе-

ристическая разность, соответствующая столбцу p_i^t , такова:

$$\bar{c}_i^t = f_i(x_i^t) - \pi_1 A_i x_i^t - \pi_{2i}. \quad (13)$$

Задача определения точки $x_i^t \in S_i$, минимизирующей это выражение, становится i -й подзадачей

$$\min \{f_i(x_i) - \pi_1 A_i x_i\}, \quad (14)$$

$$x_i \in S_i. \quad (15)$$

Это выпуклые задачи, и мы предполагаем, что они имеют решение для любого вектора π_1 . Если f_i линейна, а S_i многогранно, эта подзадача совпадает с полученной в главе II.

Пусть $\hat{x}_i$ — решение подзадачи i . Предположим, что по крайней мере одна из характеристических разностей отрицательна:

$$f_i(\hat{x}_i) - \pi_1 A_i \hat{x}_i - \pi_{2i} < 0 \text{ для некоторого } i. \quad (16)$$

Образуем столбцы

$$\hat{p}_i = \{f_i(\hat{x}_i); A_i \hat{x}_i; u_i\}', \quad i = 1, \dots, p. \quad (17)$$

Они могут быть использованы для построения сокращенной координирующей задачи, аналогичной той, которая была введена в § 2.4. Эта задача имеет в качестве переменных множество текущих базисных $\{\lambda_i^t\}$ плюс p новых переменных $\hat{\lambda}_i$, по одной для каждого столбца $\hat{p}_i$:

$$\min \{z = \sum_i \sum_{t \text{ баз}} \lambda_i^t f_i(x_i^t) + \sum_i \hat{\lambda}_i f_i(\hat{x}_i)\}$$

при ограничениях

$$\sum_i \sum_{t \text{ баз}} (A_i x_i^t) \lambda_i^t + \sum_i (A_i \hat{x}_i) \hat{\lambda}_i = b,$$

$$\sum_{t \text{ баз}} \lambda_i^t + \sum_i \hat{\lambda}_i = 1, [i = 1, \dots, p, \quad \lambda_i^t \geq 0, \quad \hat{\lambda}_i \geq 0.$$

Эта линейная задача имеет в качестве начального допустимого базис, ассоциированный с переменными λ_i^t , и p столбцов $\hat{p}_i$ вида (17) в качестве небазисных столбцов. Решение ее приводит к новому базису и новым двойственным переменным, с которыми начинается новый цикл. Все результаты, относящиеся к линейаризации на сетке и указанные ранее в этом §, приложимы и здесь. Сходимость также имеет место, что доказано Секайном [17] с помощью рассуждений, близких Данцигу [16].

§ 3.5. РАСЧЕТ ПОТОКОВ В СЕТЯХ

Направленная сеть (N, A) есть множество N элементов $\{i, j, \dots\}$, именуемых узлами, вместе с подмножеством A множества упорядоченных пар (i, j) элементов из N , называемых дугами. С каждой дугой (i, j) связана пропускная способность v_{ij} , т. е. максимальный объем некоторого однородного продукта, который может протечь из i в j за единицу времени. Форд и Фалкерсон [20] определяют поток в сети следующим образом: пусть $A(i)$ («после i ») означает множество всех узлов $j \in N$ таких, что $(i, j) \in A$,

$$A(i) = \{j \in N / (i, j) \in A\}. \quad (1)$$

Аналогично, пусть $B(i)$ («раньше i ») означает множество всех $j \in N$, таких что $(j, i) \in A$;

$$B(i) = \{j \in N / (j, i) \in A\}. \quad (2)$$

Пусть выделены два узла, s и t , именуемые узлом-источником и узлом-стоком соответственно. Тогда значение потока f_{st} из s в t есть функция $\bar{f}_{ij}$, отображающая A на множество неотрицательных вещественных чисел и удовлетворяющая условиям

$$\sum_{j \in A(i)} \bar{f}_{ij} - \sum_{j \in B(i)} \bar{f}_{ji} = \begin{cases} 0, & i \neq s, t, \\ f_{st}, & i = s, \\ -f_{st}, & i = t, \end{cases} \quad (3)$$

$$\bar{f}_{ij} \leq v_{ij} \text{ для всех } (i, j) \in A. \quad (4)$$

Уравнения (3) выражают закон сохранения, т. е. отражают тот факт, что для узлов, которые не являются ни источниками, ни стоками, входящий поток равен выходящему, в то время как поток из источника равен f_{st} , а поток из стока равен $-f_{st}$.

Соотношения (4) являются ограничениями на пропускную способность. Задача о максимальном потоке состоит в нахождении потока из s в t , величина которого является наибольшей среди всех, удовлетворяющих ограничениям.

Эта проблема может быть рассмотрена как проблема анализа: заданы пропускные способности, найти максимальный поток. Может быть рассмотрена и проблема синтеза, введенная Гомори и Ху ([18], [19]): задано множество потребностей на потоки r_{ij} и стоимость за единицу пропускной способности c_{ij} ; определить пропускные способности дуг v_{ij} , такие, что значения максимальных потоков удовлетворяют условиям

$$f_{ij} \geq r_{ij}, \quad i \neq j, \quad (5)$$

и при этом суммарные затраты

$$z = \sum_{(i,j) \in A} c_{ij} v_{ij} \quad (6)$$

минимальны. Такая проблема может возникать, например, при проектировании сетей связи. В дальнейшем предполагается, что если $(i, j) \in A$, то $(j, i) \in A$, и что $v_{ij} = v_{ji}$, $r_{ij} = r_{ji}$, $c_{ij} = c_{ji}$. Из этого следует, что $f_{ij} = f_{ji}$.

Зададим два узла i и j . Пусть X — подмножество из N , которое содержит i , но не содержит j , и $\bar{X}$ является его дополнением относительно N , так что $j \in \bar{X}$. Разрезом, отделяющим i и j , называется совокупность дуг (x, y) , направленных от некоторого узла в X к некоторому узлу в $\bar{X}$. Пропускная способность разреза равна сумме пропускных способностей его дуг. Форд и Фалкерсон доказали в [20], что значение максимального потока f_{ij} от i к j равно минимальной пропускной способности разрезом, отделяющих i от j . Пусть K_{ij}^α есть пропускная способность разреза α , отделяющего i и j . Тогда K_{ij}^α есть сумма пропускных способностей дуг v_{ij} , и неравенство (5) может быть записано в форме

$$K_{ij}^\alpha \geq r_{ij}, \quad i \neq j, \quad \text{для всех } \alpha. \quad (7)$$

Возникает большое число таких ограничений, по одному на каждый разрез, отделяющий каждую пару узлов. В качестве первого шага для преодоления этой трудности Гомори и Ху указывают, что если (7) удовлетворяется для некоторого подмножества дуг в сети, то оно удовлетворяется для всех дуг. Развитие состоит в следующем. Во-первых, даются необходимые и достаточные условия того, что заданная симметричная матрица (f_{ij}) является реализуемой, т. е. является матрицей значений максимальных потоков для некоторой сети. Результат состоит в следующем.

Теорема 1. *Для того, чтобы симметричная неотрицательная матрица (f_{ij}) была реализуемой, необходимо и достаточно, чтобы*

$$f_{ik} \geq \min(f_{ij}, f_{jk}), \quad \text{для всех } i, j, k. \quad (8)$$

Доказательство. Необходимость. Пусть (f_{ij}) реализуема. Предположим, что

$$f_{ik} < \min(f_{ij}, f_{jk}). \quad (9)$$

Значение потока f_{ik} равно минимальной из пропускных способностей всех разрезом, отделяющих i от k . Пусть X — множество узлов, содержащее i , но не содержащее k , и такое, что минимальный разрез связывает X и $\bar{X}$. Если узел j находится в X , то этот разрез разделяет j и k , и имеет пропускную способность

меньше, чем f_{jk} , что противоречит (9). Если j в $\bar{X}$, то разрез отделяет i и j , и обнаруживается аналогичное противоречие. Таким образом, (8) должно выполняться.

Заметим, что (8) может быть по индукции обобщено и записано в виде

$$f_{ip} \geq \min(f_{ij}, f_{jk}, \dots, f_{or}), \quad (10)$$

где $i, j, k, \dots, o, p$ — любой путь в сети между i и p . В дальнейшем используется именно (10).

Достаточность. Предположим, что (10) справедливо, и укажем реализацию для (f_{ij}) . Нам необходимо понятие максимального связывающего дерева. Связывающим деревом в графе называется дерево (подмножество дуг, не содержащее петель), которое связывает все узлы. Если каждая дуга имеет «вес», т. е. число, ассоциированное с ней, тогда связывающее дерево для которого сумма весов всех звеньев дерева является наибольшей, называется максимальным связывающим деревом.

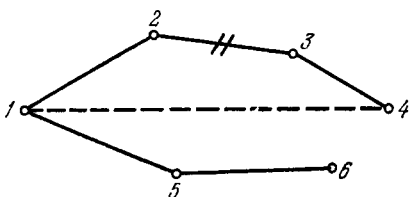


Рис. 3.13. Максимальное связывающее дерево.

Пусть веса заданы числами f_{ij} и пусть $i, j, k, \dots, o, p$ есть путь между i и p в максимальном связывающем дереве. Тогда

$$f_{ip} \leq \min(f_{ij}, f_{jk}, \dots, f_{or}). \quad (11)$$

Поскольку, если это не так, и, например, $f_{jk} < f_{ip}$, то дуга (j, k) может быть изъята из дерева и замещена дугой (i, p) , что приводит к другому связывающему дереву с увеличенной суммой весов, а это противоречит предположению о максимальности дерева (см., например, рис. 3.13). Из соотношений (10) и (11) следует, что для любого пути в максимальном связывающем дереве справедливо

$$f_{ip} = \min(f_{ij}, f_{jk}, \dots, f_{or}). \quad (12)$$

Рассмотрим теперь сеть с пропускной способностью f_{ij} для дуг в дереве и нулевой пропускной способностью для всех других дуг. Значение максимального потока между любыми узлами i и p в этой сети дается (12), так как путь в дереве является единственным путем, по которому поток может двигаться от i к p , и, таким образом, он ограничен дугой с минимальной пропускной способностью. Следовательно, (f_{ij}) является реализуемой.

Используя понятие о максимальном связывающем дереве, введенное в доказательстве предшествующей теоремы, определим доминантное дерево требований T как максимальное

связывающее дерево, где веса дуг равны требуемым потокам r_{ij} . Справедлива следующая теорема.

Теорема 2. Пусть рассматриваемая сеть имеет n узлов. Если ограничения (5) на уровни потоков удовлетворены для всех $n-1$ дуг в доминантном дереве требований T , то они выполняются и для всех пар узлов сети.

Доказательство. Рассмотрим произвольную дугу (i, p) , не принадлежащую T , и путь $(i, j, k, \dots, 0, p)$ в T от i к p . Так как f_{ij} — значение максимального потока, то из (10) следует

$$f_{ip} \geq \min(f_{ij}, f_{jk}, \dots, f_{op}) \quad (13)$$

и так как путь лежит в T , то сохраняет силу (11), где f_{ij} заменены на r_{ij} :

$$r_{ip} \leq \min(r_{ij}, r_{jk}, \dots, r_{op}). \quad (14)$$

Если все ограничения в T выполнены, то

$$f_{ij} \geq r_{ij}, f_{jk} \geq r_{jk}, \dots, f_{op} \geq r_{op}. \quad (15)$$

Тогда из (13) и (14) следует

$$f_{ip} \geq r_{ip}, \quad (16)$$

так что ограничения по потокам для дуг, не принадлежащих T , также удовлетворены.

Теперь показано, что все пары узлов, не принадлежащих доминантному дереву требований, могут не рассматриваться, а следовательно, ограничения (7) могут быть записаны в виде

$$K_{ij}^a \geq r_{ij}, (i, j) \in T, \forall \alpha \text{ для всех } \alpha. \quad (17)$$

Поскольку в T имеется только $n-1$ пара узлов, это может привести к существенному сокращению числа ограничений.

Однако (17) включает пропускные способности по всем разрезам между этими парами узлов, так что неравенств все же остается много. Конечно, для данного $(i, j) \in T$, если минимальный разрез удовлетворяет (17), то все сделано, так как для любого множества дуг должны быть проверены только $n-1$ ограничений. Однако то, какую $n-1$ пару надо рассматривать, изменяется при изменении пропускных способностей. Для того чтобы увидеть, какие из различных подмножеств ограничений могут быть выбраны эффективно, напомним двойственную задачу, в которой выбор ограничений или строк, вводимых в исходной задаче, становится выбором столбцов, вводимых в двойственный базис.

Пусть $r'_{ij\alpha}$ является вектором-строкой коэффициентов в неравенстве (17), задаваемом индексами (i, j, α) . Напомним, что K_{ij}^a есть сумма пропускных способностей дуг в разрезе α , разделяющем узлы

i от j . Следовательно, p_{ija} имеет единицы в положениях, соответствующих номерам дуг, присутствующих в этом разрезе, а остальные компоненты имеет равными нулю. Задача, двойственная к (6), (17), такова:

$$\max \sum_a \sum_{(i,j) \in T} r_{ij} \pi_{ija}, \quad (18)$$

при ограничениях

$$\sum_a \sum_{(i,j) \in T} p_{ija} \pi_{ija} \leq c, \quad (19)$$

$$\pi_{ija} \geq 0, \quad (20)$$

где c — вектор с компонентами c_{ij} . Матрица этой задачи имеет столько строк, сколько имеется дуг в сети, и столько столбцов, сколько имеется разрезов, отделяющих пары узлов $(i, j) \in T$. Исходное допустимое решение есть $\pi_{ija} = 0$, а дополнительные переменные равны c_{ij} . Двойственные переменные к задаче (18) — (20) являются, конечно, исходными переменными, пропускными способностями дуг v_{ij} . Пусть v — вектор с компонентами v_{ij} . Использование оценок для выбора столбца в (17) приводит к требованию

$$\max_{(i,j) \in T} \min_a \bar{c}_{ija} = r_{ij} - v p_{ija}. \quad (21)$$

Эта максимизация при фиксированной (i, j) эквивалентна минимизации $v p_{ija}$, которая по определению p_{ija} является пропускной способностью разреза α , отделяющего i от j . Но минимизация этой величины по α приводит к максимальному потоку между i и j , если в качестве пропускных способностей используются величины v_{ij} . Таким образом, для каждой пары $(i, j) \in T$ мы должны решить задачу о максимальном потоке, т. е. всего $n-1$ задач. Критерий оптимальности для двойственной задачи таков:

$$\bar{c}_{ija} \leq 0 \Rightarrow f_{ij} \geq r_{ij}, \quad \forall (i, j) \in T. \quad (22)$$

Если этот критерий не удовлетворяется, формируются новые столбцы, соответствующие каждому нарушению, и эти столбцы используются вместе с текущими базисными, образуя сокращенную координирующую задачу, в которой (18) максимизируется с использованием симплекс-метода. Это приводит к новым двойственным переменным, которые опять-таки используются в качестве пропускных способностей при формировании новых задач максимизации потока. При достижении оптимальности находят-ся и желаемые оптимальные пропускные способности.

Заметим, что пока оптимум не достигнут, никакая из генерированных последовательностей пропускных способностей не может быть ассоциирована с потоками, удовлетворяющими всем требованиям, поскольку используется двойственный подход. Прямой подход, в котором требования к потокам удовлетворяются на всех промежуточных итерациях, обсуждается в [18].

Пример. Рассмотрим сеть, представленную на рис. 3.14. Пусть матрицы требований к потокам и коэффициентов затрат таковы:

$$(r_{ij}) = \begin{bmatrix} - & 3 & 1 & 2 \\ & - & 4 & 1 \\ & & - & 2 \\ & & & - \end{bmatrix}, \quad (c_{ij}) = \begin{bmatrix} - & 3 & 1 & - \\ & - & 4 & 1 \\ & & - & 2 \\ & & & - \end{bmatrix}.$$

Прежде всего должно быть найдено дерево доминантных требований. Как указано в [18, стр. 261], это может быть сделано следующим образом. Выберем звено с наибольшим весом, r_{ij} . Зафиксировав его, выберем затем из остающихся звено со следующим наибольшим весом, если оно не образует петли при добавлении к уже выбранному. После $n-1$ повторений этого процесса в n -узловой сети дерево доминантных требований оказывается построенным. Для данного примера исходная сеть с указанием

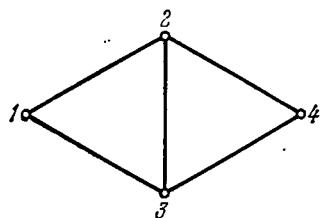


Рис. 3.14

весов дуг и соответствующее дерево показаны на рис. 3.15. Таким образом, только ограничения

$$f_{12} \geq 3, \quad f_{23} \geq 4, \quad f_{34} \geq 2$$

и необходимо учитывать.

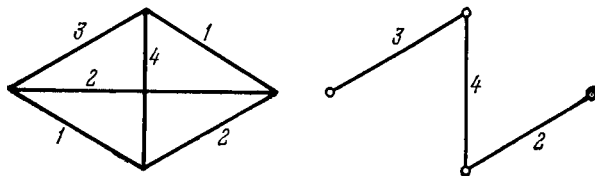


Рис. 3.15. Сеть с указанием весов и доминантное дерево требований.

Двойственная задача (18) — (20) приобретает такой вид:

$$\max 3 \sum_{\alpha} \pi_{12\alpha} + 4 \sum_{\alpha} \pi_{23\alpha} + 2 \sum_{\alpha} \pi_{34\alpha}$$

при ограничениях

$$\sum_{\alpha} p_{12\alpha} \pi_{12\alpha} + \sum_{\alpha} p_{23\alpha} \pi_{23\alpha} + \sum_{\alpha} p_{34\alpha} \pi_{34\alpha} + s = \begin{pmatrix} 3 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}.$$

Соответствие между строками матрицы ограничений и дугами сети было указано выше; например, $p_{12\alpha}$ имеет единицу в положении 3, если дуга (2, 3) включена в разрез α , разделяющий узлы 1 и 2. Переменные v_{ij} — оценки соответствующих ограничений — при достижении оптимальности окажутся искомыми пропускными способностями дуг.

Итерация 1. Начальными базисными переменными являются дополнительные переменные s , так что начальный вектор двойственных переменных $v=0$. Мы генерируем исходное множество столбцов для двойственной задачи, используя v в качестве пропускных способностей сети и решая три вспомогательные задачи максимизации потоков, для того, чтобы определить $\bar{f}_{12}$, $\bar{f}_{23}$ и $\bar{f}_{34}$. Так как начальные пропускные способности равны нулю, то и потоки равны нулю, так что любой разрез является минимальным. Произвольно отбираем следующие разрезы: разрез, разделяющий 1 и 2 $\{(1,2), (1,3)\}$, разрез, разделяющий 2 и 3 $\{(2,1) (2,3) (2,4)\}$, разрез, разделяющий 3 и 4 $\{(3,4) (2,4)\}$. Связанные с ними столбцы *) таковы:

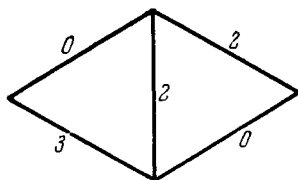


Рис. 3.16.

$$p'_{121} = (1 \ 1 \ 0 \ 0 \ 3), \quad p'_{231} = (1 \ 0 \ 1 \ 1 \ 0 \ 4), \quad p'_{341} = (0 \ 0 \ 0 \ 1 \ 1 \ 2).$$

Далее симплекс-методом решается сокращенная координирующая задача 1. Процесс решения представлен в таблице 3.5.1. В результате находится вектор двойственных переменных $v = (0 \ 3 \ 2 \ 2 \ 0)$.

Итерация 2. Используя переменные v в качестве пропускных способностей, получаем сеть, представленную на рис. 3.16. Поскольку эта сеть оказывается деревом, если опустить дуги с нулевыми пропускными способностями, решение задач, требуемых потоков, является элементарным. Эти решения таковы:

$$\begin{aligned} f_{12} &= 2, \quad \bar{c}_{122} = 3 - 2 = 1, \text{ минимальный разрез} = \{(1,2) (2,3) (3,4)\}, \\ f_{23} &= 2, \quad \bar{c}_{232} = 4 - 2 = 2, \text{ минимальный разрез} = \{(1,2) (2,3) (3,4)\}, \\ f_{34} &= 2, \quad \bar{c}_{342} = 2 - 2 = 0, \text{ минимальный разрез} = \{(2,4) (3,4)\}. \end{aligned}$$

*) Последними элементами столбцов являются коэффициенты целевой функции. (Прим. ред.)

Таблица 3.5.1

s_1	s_2	s_3	s_4	s_5	$-z$	π_{121}	π_{221}	π_{341}	$\bar{c}$
Цикл 0									
1						1	1		3
	1					1			1
		1					1		1
			1				1	1	2
				1				1	2
					1	3	4	2	0
Цикл 1									
							*		
1		-1				1			2
	1					1			1
		1					1		1
		-1	1					1	1
				1				1	2
		-4			1	3	0	2	-4
Цикл 2									
							*		
1	-1	-1							1
	1					1			1
		1					1		1
		-1	1					1	1
				1				1	2
	-3	-4			1			2	-7
Цикл 3: оптимальный									
							*		
1	-1	-1							1
	1					1			1
		1					1		1
		-1	1					1	1
		1	-1	1					1
	-3	-2	-2		1				-9

Соответствующие столбцы:

$$p'_{122} = (101013),$$

$$p'_{232} = (101014),$$

$$p'_{312} = (000112).$$

Они должны быть умножены на обратную базисную матрицу (располагаемую в столбцах от 1 до 6 конечной симплексной таблицы итерации 1) и после этого введены в ограниченную координирующую задачу. Ее структура и решение представлены в таблице. В результате находятся двойственные переменные $v = (0 \ 3 \ 3 \ 1 \ 1)$.

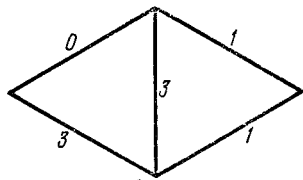


Рис. 3.17

Итерация 3. Используя v в качестве пропусковых способностей, строим новую сеть (рис. 3.17). Решение новых задач о потоках в этой сети дает

$$f_{12} = 3, \quad \min \bar{c}_{12\alpha} = 3 - 3 = 0,$$

$$f_{23} = 4, \quad \min \bar{c}_{23\alpha} = 4 - 4 = 0,$$

$$f_{34} = 2, \quad \min \bar{c}_{34\alpha} = 2 - 2 = 0.$$

Следовательно, полученный вектор v является оптимальным.

Задачи

1. Записать рекуррентные соотношения динамического программирования для задачи о ранце и наметить вычислительную процедуру. (см. [2, стр. 866] по поводу оценки требуемого для вычислений времени и более эффективного алгоритма).

2. Распространить формулировки задачи о раскрое на следующие случаи: а) имеются рулоны различной длины $L_1, L_2, \dots, L_k$, подлежащие раскрою. Рулон длины L_r стоит c_r долларов. б) Вновь полагаем, что имеются рулоны длиной $L_1, \dots, L_k$, но учтем, что раскрой производится k машинами, r -я из которых может разрезать только u_r рулонов длиной L_r в заданный период времени. Задача теперь состоит в том, чтобы определить раскрой, имеющий минимальную стоимость, и такой, что не перегружается ни одна машина, а предъявленные требования удовлетворяются. Заметим, что в б) сокращенная координирующая задача может быть сформирована с одним или более вариантами раскроя для каждой длины L_r в нем. Почему это может быть сделано в б) и не может быть сделано в а)? Рассмотреть также обобщенную схему симплекс-метода с двухсторонними ограничениями на переменные из § 5.4 в приложении к б) и выяснить ее преимущества.

3. Исследовать возможность распространения этой формулировки на двумерную ситуацию, когда раскраиваемый материал имеет как длину, так и ширину (см. [5]).

4. Построить начальное базисное допустимое решение для этапа 1, предложенного в той части § 3.2, где рассматриваются вычислительные результаты. Записать явно матрицу, обратную базисной, связанную с этим решением.

5. Записать рекуррентные соотношения динамического программирования для подзадач, определенных уравнениями (3.2-38) — (3.2-41).

6. Для тех, кто интересуется вычислениями, полные данные для ряда рассматриваемых задач приведены в п. 7.8.3. Процедуры § 3.2 и § 2.10 могут быть запрограммированы и применены к этим задачам. Можно произвести их сравнение на этих задачах или других примерах, которые нетрудно построить.

7. Показать, что оптимальное решающее правило $m(i)$ в задаче управления запасами может быть выявлено на основе оптимального решения (3.3-54) — (3.3-57), когда это решение удовлетворяет теореме 1 § 3.3.

8. Для процесса запасаания, заданного соотношениями (3.3-23) — (3.3-25), положим, что коэффициенты затрат равны

$$c_h(i) = i, \quad c_o(m(i)) = 2m(i), \\ c_p(r - i - m(i)) = \max[0, 4(r - i - m(i))].$$

Записать задачу линейного программирования (3.3-54) — (3.3-57) с указанными коэффициентами, решить ее и показать, что оптимальное решение обладает свойствами, установленными теоремой 1 § 3.3. Выявить оптимальное решающее правило и стационарные вероятности x_i , соответствующие этому правилу, сравнить оптимальные затраты с затратами, получающимися при стратегии $m(i) = 0$, для всех i . По поводу сходной задачи см. [11, стр. 263].

9. ([11]). Рассмотрим задачу управления запасами, в которой издержки включают фиксированную часть, возникающую, когда уровень производства изменяется при переходе от данного периода к следующему. Здесь состояние является двухкомпонентным вектором и определяется начальным запасом и уровнем производства предшествующего периода. Распространить формулировку § 3.3 на такие задачи. Доказать, что если имеется r возможных уровней производства, то результирующая задача линейного программирования имеет $r(n+1)$ строк.

10. Дать стоимостную интерпретацию оптимальных значений двойственных переменных в задаче (3.3-54) — (3.3-57).

11. Распространить формулировку обобщенного линейного программирования на случай, когда множества S_i многогранны, но могут быть неограниченны.

12. Показать, что если f и q_i являются выпуклыми функциями, множество R , описываемое (3.3-15), выпукло.

13. Пусть x^k — решение подзадачи

$$\min \{L(x, u^k) = f(x) + \sum_{i=1}^m u_i^k g_i(x)\},$$

где $u_i^k = -\pi_i^k$, и π_i^k — двойственные переменные, соответствующие оптимальному решению сокращенной координирующей задачи на итерации k . Показать, что если x^k удовлетворяет условиям

$$g_i(x^k) \leq 0, \quad i = 1, \dots, m, \\ u_i^k g_i(x^k) = 0, \quad i = 1, \dots, m,$$

то x^k является решением исходной задачи (3.4-1) — (3.4-2).

14. Дана задача

$$\min \{f = (x - 2)^2 + 2y\}$$

при ограничениях

$$g_1 = x^2 - y \leq 0, \quad g_2 = x + y \leq 2, \quad y \geq 0.$$

а) Решить задачу.

б) Выбрать начальные узлы $x_1=0$, $x_2=1$. Используя начальный базис из двух дополнительных переменных и λ_1 (вес, ассоциированный с x_1), решить начальную сокращенную координирующую задачу. Учтите, что y входит линейно и по y нет необходимости в линеаризации.

с) Сформулировать и решить первую подзадачу. Заметьте, что условия задачи (3.4-14) удовлетворяются, а следовательно, новая узловая точка является оптимальным решением.

д) Ввести новый столбец и решить новую координирующую задачу, используя оптимальный базис из б) в качестве начального. Доказать, что конечное решение является оптимальным.

15. Показать, что если существует x такой, что $g_i(x) < 0$, $i=1, \dots, m$, то возможен выбор узловых точек x_i , для которых ограниченная координирующая задача (3.4-6) — (3.4-9) имеет невырожденное базисное решение. Построить такое решение.

16. В следующем случае процедура линеаризации на сетке является конечной. Рассмотрим задачу

$$\min \left\{ \sum_{j=1}^n f_j(x_j) / Ax = b, 0 \leq x \leq u \right\},$$

где u — положительный вектор, и каждая функция f_j кусочно-линейна на отрезке $[0, u_j]$ с точками излома $\{x_{jk}\}$. Линеаризуем каждую f_j на сетке $\{x_{jk}\}$:

$$f_j(x_j) = \sum_k \lambda_{jk} f_j(x_{jk}), \quad \sum_k \lambda_{jk} = 1, \quad \lambda_{jk} \geq 0.$$

Пусть $\{\lambda_{jk}^0\}$ — решение соответствующей задачи ЛП. Показатель, что

$$x_j^0 = \sum_k \lambda_{jk}^0 x_{jk}, \quad j = 1, \dots, n,$$

является решением исходной задачи.

17. Предполагая, что соотношение (3.5-8) справедливо, установить (3.5-10) по индукции.

18. Для того чтобы задача максимизации (3.5-21) была разрешима, переменные u должны быть неотрицательны. Как это гарантируется с самого начала итераций 2, 3...? Как это гарантируется в начале итерации 1?

ЛИТЕРАТУРА

1. P. C. Gilmore and R. E. Gomory, A Linear Programming Approach to the Cutting Stock Problem, Operations Res., 9, 1961, pp. 849—859.
2. P. C. Gilmore and R. E. Gomory, A Linear Programming Approach to the Cutting Stock Problem, Part II. Operations Res., Nov.-Dec. 1963, pp. 863—887.
3. M. L. Balinski, Integer Programming: Methods, Uses, Computation, Management Sci., 12, No. 3, 1965, pp. 253—313.
4. E. M. I. Beale, A. B. Hughes and R. E. Small, Experiences in Using a Decomposition Program, Computer J., 8, No. 1, 1965, pp. 13—18.
5. P. C. Gilmore and R. E. Gomory, Multistage Cutting Stock Problems of Two or More Dimensions, Operations Res., 13, 1965, pp. 94—120.

6. A. S. M a n n e, Programming of Economic Lot Sizes, Management Sci., 4, 1958, pp. 115—135.
7. B. P. D z i e l i n s k i and R. E. G o m o r y, Optimal Programming of Lot Sizes, Inventory and Labor Allocations, Management Sci., 11, No. 9, 1965, pp. 874—890.
8. L. S. L a s d o n and J. E. M a c k e y, An Efficient Algorithm for Multi-item Scheduling, Systems Research Center Report SRC 68-9, Case Western Reserve University, 1968.
9. G. B. D a n t z i g and R. M. V a n S l y k e, Generalized Upper Bounding Techniques, J. Computer Systems Sci., 1, 1967, pp. 213—226.
10. G. B. D a n t z i g, Linear Programming and Extensions, Princeton University Press, Princeton, N. J., 1963, chap. 22. [Русский перевод: Дж. Данциг, Линейное программирование, его обобщения и применения, «Прогресс», М., 1966.]
11. A. S. M a n n e, Linear Programming and Sequential Decisions, Management Sci., 6, No. 3, 1960, pp. 259—269.
12. P. W o l f e and G. B. D a n t z i g, Linear Programming in a Markov Chain, Operations Res., 10, Sept-Oct. 1962, pp. 702—710.
13. P. W o l f e, Methods of Nonlinear Programming, in «Nonlinear Programming», J. Abadie, ed., John Wiley & Sons, Inc., New York, 1967, pp. 100—142.
14. P. W o l f e, Foundations of Nonlinear Programming: Notes on Linear Programming and Extensions, Part 65. DDC Rept. AD-619-968, also RAND Memorandum RM-4669-PR, 1965.
15. C. E. M i l l e r, The Simplex Method for Local Separable Programming, in Recent Advances in Mathematical Programming, R. L. Graves and P. Wolfe, eds., McGraw-Hill, New York, 1963, pp. 89—100.
16. G. B. D a n t z i g, Linear Programming and Extensions, Princeton University Press, Princeton, N. J., 1963, chaps 22 and 24. [Русский перев. см. [10].]
17. Y. S e k i n e, Decentralized Optimization of an Interconnected System. IEEE Trans. Circuit Theory, June 1963, pp. 161—168.
18. R. E. G o m o r y and T. C. H u, An Application of Generalized Linear Programming to Network Flows, J. Soc. Ind. Appl. Math., 10, No. 2, 1962, pp. 260—283.
19. R. E. G o m o r y and T. C. H u, Multi-terminal Network Flows, J. Soc. Ind. Appl. Math., 9, No. 4, 1961, pp. 551—570.
20. L. R. F o r d and D. R. F u l k e r s o n, Flows in Networks, Princeton University Press, Princeton, N. J., 1962. [Русский перевод: Л. Форд, Д. Фалкерсон, Потоки в сетях, «Мир», М., 1966.]

ГЛАВА IV

РАСЧЛЕНЕНИЕ И РЕЛАКСАЦИОННЫЕ ПРОЦЕДУРЫ В ЛИНЕЙНОМ ПРОГРАММИРОВАНИИ

§ 4.1. ВВЕДЕНИЕ

В этой главе изучаются задачи ЛП, в которых матрицы ограничений имеют блочно-диагональную структуру, искаженную наличием связывающих переменных или ограничений. Задачи такой структуры имеют следующий вид:

$$\max \{z = c_1x_1 + \dots + c_px_p + c_0y\} \quad (1)$$

при ограничениях

$$A_1x_1 + \dots + A_px_p + D_0y = b_0, \quad (2)$$

$$B_1x_1 \quad \quad \quad + D_1y = b_1,$$

$$B_2x_2 \quad \quad \quad + D_2y = b_2, \quad (3)$$

$$\dots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$B_px_p + D_py = b_p,$$

$$x_i \geq 0, \quad i = 1, \dots, p, \quad y \geq 0. \quad (4)$$

Для решения может быть использован принцип декомпозиции Данцига — Вулфа. Если в задаче имеются только связывающие переменные, процедура Данцига — Вулфа применима к двойственной (которая имеет только связывающие ограничения). Если присутствуют как связывающие переменные, так и связывающие ограничения, матрица ограничений может быть расчленена на две части (2) и (3), как показано выше. Соотношения (3) становятся ограничениями подзадачи, а (2) определяют координирующую задачу. Так как подзадача имеет специальную структуру, то она может быть разрешена либо методами этой главы, либо путем перехода к двойственной с вторичным приложением процедуры Данцига — Вулфа. В любом случае полная схема представляет собой трехуровневый алгоритм, включающий итерацию (решение подзадач) внутри внешнего итеративного цикла (решение координирующей задачи). Так как

внутренняя итерация является достаточно сложной, то эта процедура мало приемлема в вычислительном отношении, в особенности, если учесть, что метод Данцига — Вулфа зачастую медленно сходится (см. §§ 3.2, 2.9 и 2.10).

Алгоритмы, описываемые в этой главе, дают полезную альтернативу для решения задач типа (1) — (4). Они называются *методами расчленения*, поскольку переменные сначала разделяются на два подмножества: связывающие переменные y и блочные переменные x . Переменные x_i далее разделяются на множества зависимых и независимых путем анализа базисных матриц. Это расчленение позволяет исключить зависимые переменные и блочные ограничения из задачи, что приводит к проблемам меньшей размерности. При исключении зависимых переменных теряется контроль за выполнением условий их неотрицательности. Таким образом, можно сказать, что эти ограничения ослабляются, релаксируются, и требуются дополнительные итерации, определяющие такое новое расчленение, чтобы в конечном счете условия неотрицательности оказались выполненными. Поскольку, как было указано Джофффрионом [1], концепция релаксации ограничений является объединяющей идеей, в рамках которой может быть понята работа всех этих методов, мы начнем с обсуждения этой концепции.

§ 4.2. РЕЛАКСАЦИЯ

Рассмотрим общую задачу выпуклого программирования

$$\left. \begin{array}{l} \max f(x), \\ \text{при ограничениях} \\ g_i(x) \geq 0, \quad i = 1, \dots, m, \\ x \in S, \end{array} \right\} \text{Задача } P \quad \begin{array}{l} (1) \\ (2) \\ (3) \end{array}$$

где f и g_i — вогнутые функции вектора x и S — выпуклое подмножество E^n . Рассмотрим здесь случай, когда число ограничений $g_i \geq 0$ достаточно велико. Тогда разумная стратегия решения состоит в релаксации, т. е. временном отбрасывании некоторых из этих ограничений и решении оставшейся задачи. Если эта задача является неразрешимой, то неразрешима и исходная. Если она разрешима, и если полученное решение удовлетворяет релаксированным ограничениям, то оно оптимально и для исходной задачи P . В противном случае мы вынуждены ввести одно или более из опущенных ограничений и повторить процедуру. Представляется, что релаксация должна быть эффективной стратегией, если известно, что относительно малое число ограничений являются существенными в оптимальной точке. Это справедливо для невырожденных задач ЛП (точно n существенных) и, вооб-

ще говоря, проявляется даже более сильно в нелинейных задачах [1]. Джоффрион [1] формализовал эту идею следующим образом. Пусть M — множество индексов $\{1, \dots, m\}$, пусть R означает подмножество M . Обозначим через P^R задачу

$$\max f(x), \quad (4)$$

при ограничениях

$$g_i(x) \geq 0, \quad i \in R, \quad (5)$$

$$x \in S'. \quad (6)$$

Предположим, что может быть найдено такое начальное множество R , что P^R имеет конечный супремум, и что все конечные супремумы достижимы. Тогда релаксационная стратегия приобретает следующий вид:

0. Положим $\bar{f} = +\infty$ и выберем начальное множество R такое, что f ограничено на ограничениях P^R .

1. Решим P^R . Если P^R неразрешима, то неразрешима P . В противном случае получаем оптимальное решение x^R .

2. Если $g_i(x^R) \geq 0$, $i \in M - R$, x^R оптимально для исходной задачи P .

3. В противном случае, пусть V — подмножество $M - R$, содержащее индексы по крайней мере одного из невыполненных ограничений. Положим $D = \{i/g_i(x^R) > 0, i \in R\}$.

4. Если $f(x^R) = \bar{f}$, то заменим R на $R' = R \cup V$ и возвращаемся к шагу 1.

5. Если $f(x^R) < \bar{f}$, то заменим R на $R' = R \cup V - D$, заменим $\bar{f}$ на $f(x^R)$ и перейдем к шагу 1.

Эта процедура добавляет одно или более нарушенных ограничений и, если $f(x^R)$ уменьшается, изымает те ограничения в R , которые несущественны в точке x^R . То обстоятельство, что ни одно из ограничений не изымается, если $f(x^R) = \bar{f}$, гарантирует, что приходится решать только конечное число задач типа P^R , что и будет показано ниже. Доказательство требует использования двух предварительных результатов.

Теорема 1. Если x^R — решение P^R , то x^R является также решением P^{R-D} . Далее, если x^R является единственным решением P^R , то оно также является единственным решением P^{R-D} .

Доказательство. Предположим, что имеется вектор $\hat{x}$, удовлетворяющий ограничениям $R - D$, и такой, что

$$f(\hat{x}) > f(x^R). \quad (7)$$

Рассмотрим точки на отрезке, соединяющем x^R и $\hat{x}$:

$$x_\lambda = \lambda \hat{x} + (1 - \lambda) x^R, \quad 0 < \lambda < 1. \quad (8)$$

Так как S выпукло и все g_i вогнуты, $x_\lambda \in S$ и x_λ удовлетворяет ограничениям $R - D$ для всех $0 < \lambda < 1$. Рассмотрим ограничения

в D . Так как g_i вогнуты,

$$g_i(x_\lambda) \geq \lambda g_i(\hat{x}) + (1 - \lambda) g_i(x^R);$$

Поскольку $g_i(x^R) > 0$, $i \in D$, то для $\lambda > 0$ и достаточно малых $g_i(x_\lambda) > 0$, $i \in D$. Точка x_λ является, следовательно, для таких λ допустимой в P^R . Однако в силу вогнутости f

$$f(x_\lambda) \geq \lambda f(\hat{x}) + (1 - \lambda) f(x^R) > f(x^R), \quad (9)$$

что противоречит оптимальности x^R .

Докажем единственность. Если $f(\hat{x}) = f(x^R)$, то (9) дает $f(x_\lambda) = f(x^R)$ для всех $\lambda > 0$ и достаточно малых, что противоречит единственности x^R .

Следующий результат показывает, что последовательность максимумов в подзадачах, строящихся в ходе процедуры релаксации, является монотонно невозрастающей.

Теорема 2.

$$f(x^{R'}) \leq f(x^R). \quad (10)$$

Доказательство. Если $R' = R \cup V$, то теорема, очевидно, справедлива. Далее, в силу теоремы 1

$$f(x^{R-D}) = f(x^R), \quad (11)$$

так как

$$f(x^{(R-D) \cup V}) \leq f(x^{R-D}). \quad (12)$$

Теорема доказана.

Заметим, что вогнутость f и g_i и выпуклость S позволяет нам опускать несущественные ограничения. Теорема 2, конечно, справедлива и при отсутствии выпуклости, если ограничения только добавляются, но не опускаются. Это имеет место, например, в алгоритме Бендерса (§ 6.3). Однако опускание ограничений существенно для того, чтобы, например, метод Риттера и Розена, описываемый ниже, был бы эффективным.

Теорема 3. Процедура релаксации заканчивается после решения конечного числа задач P^R и приводит к оптимальному решению P либо к построению такого множества ограничений, что текущая задача P^R окажется неразрешимой.

Доказательство. Так как M конечно, то оно имеет только конечное число различных подмножеств. При убывании $f(x^R)$ от итерации к итерации ни одно подмножество не может повториться. Так как ни одно ограничение не изымается, если $f(x^{R'}) = f(x^R)$, и по крайней мере одно ограничение добавляется, то f может оставаться постоянным только в течение конечного числа итераций. Таким образом, за конечное число шагов процедура должна закончиться на этапе 1 либо на этапе 2 (возможно, при $R = M$).

Заметим, что релаксация уже использовалась в § 3.5, где неудовлетворявшиеся первичные ограничения генерировались и добавлялись (в качестве столбцов в двойственной), приводя к новой P^R .

Связь с двойственным симплекс-методом. Нетрудно показать, что если P линейна, то при соответствующем выборе V релаксация эквивалентна двойственному симплекс-методу Лемке. Для того чтобы показать это, удобно записать P в виде

$$\max cx \quad (13)$$

при ограничениях

$$Ax = b, \quad (14)$$

$$x_i \geq 0, \quad i = 1, \dots, n, \quad (15)$$

где A является $m \times n$ матрицей, $n > m$, и $\text{rang}(A) = m$. Релаксируемыми ограничениями являются требования неотрицательности (15). Будем обозначать их индексами соответствующих переменных и положим

$$N = \{1, 2, \dots, n\}. \quad (16)$$

Уравнения (14) всегда будут удовлетворяться, т. е. в данном случае множество S имеет вид

$$S = \{x/Ax = b\}. \quad (17)$$

В дальнейшем предположим следующее.

Предположение 1. Все допустимые базисы задачи, двойственной к (13) — (15), являются невырожденными, т. е. точно m двойственных ограничений являются существенными.

Теорема 4. Пусть индексы небазисных переменных в начальном цикле двойственного симплекс-метода содержатся в начальном множестве R процедуры релаксации. Далее, пусть множество V в шаге 3 релаксации представляет собой индекс ограничения неотрицательности, нарушенного в наибольшей степени. Тогда для цикла k двойственного метода и метода релаксации индексное множество небазисных переменных в двойственном методе совпадает с множеством R в релаксации, причем $x^R = x^k$, где x^k — вектор значений x в двойственном симплекс-методе.

Доказательство требует использования трех лемм.

Лемма 1. Если R — индексное множество небазисных переменных на цикле k двойственного метода, то x^k — единственное оптимальное решение P^R .

Доказательство. Пусть B — базис на цикле k двойственного метода. Достаточно показать, что $p_B = c_B B^{-1}$ — допустимое решение для двойственной к P^R со значением целевой функции, равным $c_B x_B$. Ниже описана задача P^R и двойственная

к ней:

$$\begin{array}{ll} P^R & \text{Двойственная к } P^R \\ \max cx & \min v = \pi b \\ Ax = b & \pi B = c_B \\ x_i \geq 0, i \in R & \pi A^i \geq c_i, i \in R \end{array}$$

где A^i — i -й столбец A . В двойственном симплекс-методе базис B является двойственно-допустимым, так что π_B удовлетворяет всем двойственным ограничениям и

$$v(\pi_B) = c_B B^{-1} b = c_B x_B. \quad (18)$$

Таким образом, x^* является решением P^R . В силу предположения 1 все двойственные ограничения в R удовлетворяются строго, так что оптимальное решение единственно.

Л е м м а 2. Пусть R — индексное множество небазисных переменных на цикле k двойственного симплекс-метода. Предположим, что алгоритм заканчивается на этом цикле с получением информации, что прямая задача неразрешима, т. е. $\min \bar{b}_i = \bar{b}_r < 0$ и все $\bar{a}_{rj} \geq 0$. Тогда подзадача в релаксационной процедуре, образуемая путем добавления индекса j_r (индекса r -й базисной переменной) к R , также является неразрешимой.

Доказательство. Достаточно построить неограниченное решение задачи, двойственной к P^{R+j_r} . Рассматриваемая пара двойственных задач такова:

$$\begin{array}{ll} P^{R+j_r} & \text{Двойственная к } P^{R+j_r} \\ \max z = cx & \min v = \pi b \\ Ax = b & \pi A^i \geq c_i, i \in N - R - j_r \\ x_i \geq 0, i \in R + j_r & \pi A^i \geq c_i, i \in R + j_r \end{array}$$

Рассмотрим решение двойственной задачи

$$\pi = c_B B^{-1} + \theta B_r^{-1}, \quad \theta \geq 0, \quad (19)$$

где B_r^{-1} — r -я строка B^{-1} . Так как

$$\begin{aligned} \bar{a}_{rj} &= (B^{-1} A^j)_r = B_r^{-1} A^j \geq 0, j \in R, \\ &= 0, j \in N - R, j \neq j_r, \\ &= 1, j = j_r, \end{aligned}$$

то B_r^{-1} удовлетворяет однородным двойственным ограничениям, т. е. таким, что в них все c_i заменены нулями. Таким образом, π в (19) является двойственно допустимым для всех $\theta \geq 0$, причем

целевая функция равна

$$v(\pi) = c_B x_B + \bar{\theta} \bar{b}_r. \quad (20)$$

Так как $\bar{b}_r < 0$, то $v(\pi) \rightarrow -\infty$ при $\theta \rightarrow \infty$.

Лемма 3. Если x_s входит в базис на цикле k двойственного метода, то $x_s > 0$ в окончательном базисном решении.

Доказательство. В силу ведущей операции $x_s = \frac{\bar{b}_r}{\bar{a}_{rs}}$, а по схеме решения $\bar{b}_r < 0$ и $\bar{a}_{rs} < 0$.

Доказательство теоремы 4. В силу выбора начального множества R , из леммы 1 следует, что теорема верна при $k=1$. Предположим, что результат справедлив для k и докажем его для $k+1$. Цикл k двойственного метода либо 1) заканчивается на оптимальном решении, либо 2) приводит к информации, что задача неразрешима, либо 3) задает переход к новому циклу. В случае 1) согласно лемме 1 релаксация также заканчивается. По лемме 2 в случае 2) релаксация также заканчивается на цикле $k+1$ с выяснением неразрешимости подзадачи. В случае 3) ситуации на цикле k и $k+1$ двойственного метода представлены на рис. 4.1. По индуктивному предположению R таково, как показано, и $x^R = x^*$. Так как все переменные в R равны нулю, при релаксации не опускается ни одно ограничение. Выбирается x_{j_r} как наиболее отрицательная переменная в $N - R$ и добавляется индекс j_r к R . Затем решается P^{R+j_r} . По лемме 1 решение x^{k+1} является единственным решением P^{R+j_r-s} . По лемме 3 в этом решении x_s положительна, так что x^{k+1} является единственным решением P^{R+j_r} . Так как двойственная невырождена, $cx^{R+j_r} < cx^R$, и при релаксации изымаются все положительные переменные из $R + j_r$. Единственной такой переменной является x_s , так что $R + j_r$ переходит в $R + j_r - s$, и тем самым теорема доказана.

Вычислительные аспекты. Для того чтобы релаксация была эффективной, необходимо иметь эффективные процедуры идентификации нарушенных ограничений (т. е. выбора множества V) и решения последовательности задач P^R . Широкие возможности заключаются в выборе V . Обычным является выбор индексов ограничений с наибольшей невязкой, как в двойственном симплекс-методе. Этот выбор зачастую используется, когда множество ограничений действительно велико, и когда функции, определяющие ограничения, имеют специальную структуру. Тогда индекс наиболее нарушаемого ограничения часто может быть определен без оценки всех ограничений, а путем решения специальной подзадачи (см. § 3.5 и 6.3). Эта процедура генерации строк двойственной процедурам генерации столбцов,

обсуждавшимся выше. Таким образом, вычислительный опыт использования различных тактик для выделения столбцов в линейных задачах здесь также может быть использован. Как иллюстрировалось при решении задачи о раскрое в § 3.1, случайный выбор единственного нарушенного ограничения, вероятно, является много менее эффективным, чем выбор наиболее нарушенного. Далее, если в V выбирать так, чтобы оно содержало

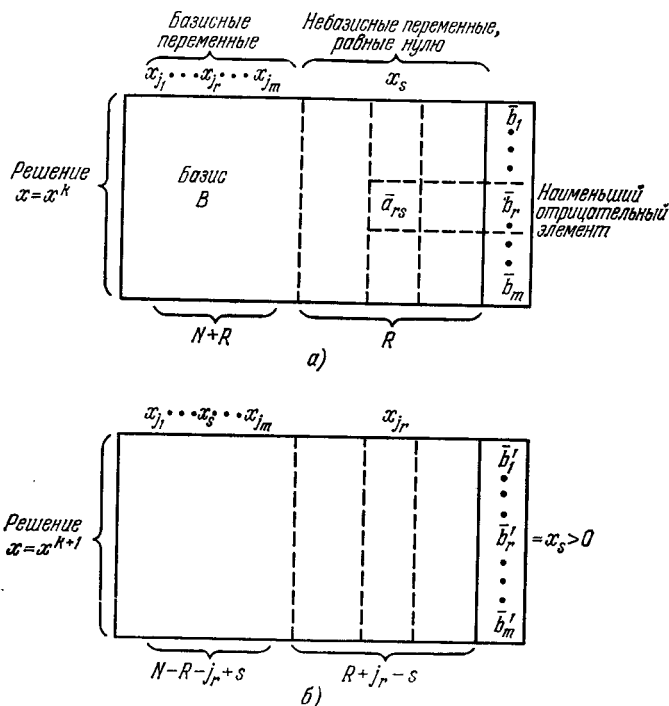


Рис. 4.1. Таблицы двойственного метода: а) на итерации k ; б) на итерации $k+1$.

больше индексов, например, вводить в V индексы k наиболее нарушенных ограничений, то изменения в $f(x^k)$ при переходе от итерации к итерации возрастут. Однако затраты труда, требуемого на «перерешивание» последовательных задач P^k , также должны возрасти. Вариант, который полезен, когда имеется много ограничений, состоит в том, чтобы выделить k наиболее нарушенных ограничений и применить релаксацию к ограниченной задаче, учитывающей эти f вместе с текущими из $R-D$. Внутри этой ограниченной задачи V может быть выбрано, например, как индекс наиболее нарушенного ограничения. В своей простей-

шей форме такая тактика состоит в нахождении столбцов с k наиболее отрицательными характеристическими разностями и проведении стандартных симплексных итераций в задаче, где выделенные столбцы фигурируют в качестве начальных небазисных. Эта процедура именуется множественным оцениванием, или субоптимизацией (см. § 5.2), и приводит к заметному сокращению времени вычислений в тех случаях, когда используется мультипликативное представление обратной матрицы. Введение более, чем одного нарушенного ограничения легко выполняется, когда релаксация применяется к задачам ЛП со связывающими ограничениями и переменными, как показано в следующих параграфах.

Генерация двойственных допустимых точек. С каждой задачей P^R можно связать вектор оптимальных множителей Лагранжа, который существует при некоторых дополнительных предположениях. Легко показать, что эти векторы являются допустимыми для минимаксной двойственной к P и образуют монотонно невозрастающую последовательность значений ее целевой функции (см. § 7.4 по поводу дальнейших деталей):

$$\min_{u \geq 0} \sup_{x \in S} (f(x) + \sum_{i=1}^m u_i g_i(x)).$$

По теореме о седловой точке любой вектор множителей Лагранжа $u^R \geq 0$, связанный с P^R , удовлетворяет соотношению

$$f(x^R) = \max_{x \in S} (f(x) + \sum_{i \in R} u_i^R g_i(x)).$$

Выбирая $u_i^R = 0$, $i \notin R$, мы получаем допустимую точку для задачи, двойственной к P . Так как значение целевой функции в двойственной задаче равно $f(x^R)$, то последовательность этих значений является невозрастающей. Для описываемых далее релаксационных алгоритмов будет существенной двойственная допустимость. На каждом цикле будет генерироваться не только точка, допустимая для двойственной к P , но окажется, что оптимальное решение задачи, двойственной к P^R , будет допустимым и для двойственной к последующей P^R . Это позволит эффективно решать последовательные задачи P^R с помощью двойственного симплекс-метода.

Закрепление. Интересно отметить, что тактика релаксирования ограничений в исходной задаче соответствует тому, что в двойственной условия $u_i \geq 0$ заменяются на $u_i = 0$. Как указано в [1], такое закрепление является полезным приемом в тех случаях, когда оно используется при решении прямой задачи, и может быть формализовано почти по такой же схеме, как и релаксация. Для того чтобы показать это, определим

подзадачу P^R следующим образом:

$$\begin{aligned} \max f(x), \\ g_i(x) = 0, \quad i \in R, \\ g_i(x) \geq 0, \quad i \in M - R, \\ x \in S, \end{aligned}$$

где R вновь обозначает подмножество M . Для того чтобы обеспечить выпуклость любой задачи P^R , все g_i должны предполагаться линейными. Любые нелинейные ограничения следует отнести к множеству S . Предположим, что может быть найдено такое начальное непустое множество R , что P^R имеет допустимое решение, что все конечные максимумы достижимы и что, если P^R имеет оптимальное решение, то оно имеет и оптимальный вектор множителей Лагранжа. Тогда процедура закрепления может быть описана следующим образом:

0. Положим $\bar{f} = -\infty$ и выберем начальное множество R так, что P^R имеет допустимое решение.

1. Решим P^R . Если целевая функция неограничена, то и исходная задача P не имеет ограниченного решения. В противном случае получаем оптимальное решение x^R .

2. Пусть u_i^0 — оптимальный множитель Лагранжа, связанный с i -м ограничением. Если $u_i^0 > 0$, $i \in R$, то x^R есть решение P .

3. В противном случае пусть V — подмножество R , содержащее индексы по крайней мере одного отрицательного u_i^0 , и пусть

$$D = \{i/g_i(x^R) = 0, i \in M - R\}.$$

4. Если $f(x^R) = \bar{f}$, заменим R на $R' = R - V$ и перейдем к шагу 1.

5. Если $f(x^R) > \bar{f}$, заменим R на $R' = R \cup D - V$, заменим $\bar{f}$ на $f(x^R)$ и перейдем к шагу 1.

Нетрудно показать, что последовательность значений целевых функций $f(x^R)$, возникающая в результате этой процедуры, является неубывающей, и что процедура заканчивается за конечное число циклов. Процедура закрепления, очевидно, двойственна процедуре релаксации. Поэтому неудивительно, что она оказывается полезной в задачах, где число переменных много больше числа ограничений. Чтобы убедиться в этом, заметим, что ограничения вида $g_i \geq 0$ часто имеют вид условий $x_i \geq 0$, или могут быть приведены к этому виду соответствующими преобразованиями. Но тогда закрепление $x_i = 0$ просто исключает переменную из задачи. Таким образом, если P задача ЛП, $g_i(x) = x_i$ и R выбирается, как множество индексов текущих небазисных переменных, то процедура закрепления становится идентичной

обычному симплекс-методу (по крайней мере, если все базисы невырождены). В линейных задачах с большим числом столбцов закрепление приводит к схеме генерирования столбцов (см. главу III). Множество V , возникающее на шаге 3 процедуры закрепления, является тогда подмножеством небазисных переменных, имеющих неотрицательные характеристические разности, и эффективность алгоритма зависит от того, насколько легко может быть найдено это подмножество. Конечно, необязательно для сокращения числа переменных, чтобы $g_i(x)$ совпадали с x_i . Закрепление любого подмножества p линейно независимых ограничений позволяет исключить p переменных.

Хотя закрепление подобно релаксации находит широкое приложение в задачах ЛП, его полезность не ограничивается линейными проблемами. Блестящим примером приложения закрепления к нелинейным задачам является алгоритм расчленения Розена (см. § 6.2). В нем ограничения $g_i(x)$ являются нелинейными по всем переменным, что приводит к невыпуклым задачам P^R . Поэтому приходится предпринимать дополнительные операции, не указанные здесь, и включающие, в частности, линеаризацию P^R .

Комбинированное использование закрепления и релаксации обсуждается в [11], а изложение некоторого вычислительного опыта можно найти в [10].

§ 4.3. ЗАДАЧИ СО СВЯЗЫВАЮЩИМИ ОГРАНИЧЕНИЯМИ И СВЯЗЫВАЮЩИМИ ПЕРЕМЕННЫМИ

Риттером [3] была развита процедура расчленения для задач с блочно-диагональной матрицей, дополненной связывающими ограничениями и связывающими переменными. На каждой итерации переменные задачи разделяются на два подмножества — подмножество S_1 , в котором релаксируются ограничения неотрицательности, и подмножество S_2 , в котором они сохраняются. Если переменные в S_1 окажутся неотрицательными, то текущее решение будет оптимальным для исходной задачи. Итеративная процедура исходит из начального расчленения и уточняет его вплоть до нахождения оптимального. Процедура является двойственным методом, в котором генерируется последовательность двойственно-допустимых крайних точек и невозрастающих значений целевой функции. Запишем далее рассматриваемую задачу и ей двойственную.

Прямая:

$$\max \{z = \sum_{i=1}^p c_i' x_i + c_0' y\} \quad (1)$$

при ограничениях:

$$\sum_{i=1}^p A_p x_p + D_0 y = b_0, \quad (2)$$

$$B_i x_i + D_i y = b_i, \quad i = 1, \dots, p, \quad (3)$$

$$x_i \geq 0, \quad i = 1, \dots, p, \quad y \geq 0. \quad (4)$$

Двойственная:

$$\min \{v = \sum_{i=1}^p b'_i u_i + b'_0 u_0\} \quad (5)$$

при ограничениях:

$$\left. \begin{aligned} \sum_{i=1}^p D'_i u_i + D'_0 u_0 &\geq c_0, \\ B'_i u_i + A'_i u_0 &\geq c_i, \quad i = 1, \dots, p. \end{aligned} \right\} \quad (6)$$

Как и ранее, A_i есть $m_i \times n_i$ матрица, B_i — $m_i \times n_i$ матрица, D_i — $m_i \times n_0$ матрица.

Предположим, что каждая матрица B_i имеет ранг m_i и, следовательно, содержит неособую подматрицу B_{ii} . Заметим, что это не обязательно имеет место, даже если система (2) — (3) имеет полный ранг. Если $\text{ранг}(B_i) < m_i$, искусственные переменные (со столбцами в виде единичных векторов) могут быть добавлены в матричные строки, включающие B_i , таким образом, что увеличенная матрица будет иметь ранг m_i . Если этим искусственным переменным приписать коэффициенты в целевой функции, произвольно большие по величине и отрицательные, то все они в оптимальном решении обратятся в нуль. Если ранг B_i равен m_i для каждого i , и если известен начальный вектор $y_0 \geq 0$, такой, что существует допустимое решение системы ограничений

$$B_i x_i = b_i - D_i y_0, \quad x_i \geq 0, \quad i = 1, \dots, p, \quad (7)$$

то может быть найдено «хорошее» множество начальных базисов путем решения подзадач

$$\left. \begin{aligned} \max c'_i x_i \\ B_i x_i = b_i - D_i y_0, \quad x_i \geq 0 \end{aligned} \right\} \quad (8)$$

и использования оптимальных базисных матриц в качестве B_{ii} . После выделения B_{ii} ограничения (3) могут быть переписаны в виде

$$B_{i1} x_{i1} + B_{i2} x_{i2} + D_i y = b_i. \quad (9)$$

Так как каждая из B_{ii} — неособая, то уравнение (9) разрешимо

относительно x_{i1} :

$$x_{i1} = B_{i1}^{-1}b_i - B_{i1}^{-1}B_{i2}x_{i2} - B_{i1}^{-1}D_i y. \quad (10)$$

Расчленим также переменные в целевой функции (1) и в связывающем ограничении (2):

$$z = \sum_{i=1}^p (c'_{i1}x_{i1} + c'_{i2}x_{i2}) + c'_0 y, \quad (11)$$

$$\sum_{i=1}^p (A_{i1}x_{i1} + A_{i2}x_{i2}) + D_0 y = b_0. \quad (12)$$

Подставляя (10) в (11) и (12), исключаем ограничение (3) и переменные x_{i1} . Положим:

$$d'_i = c'_{i2} - c'_{i1}B_{i1}^{-1}B_{i2}, \quad (13)$$

$$d'_0 = c'_0 - \sum_{i=1}^p c'_{i1}B_{i1}^{-1}D_i, \quad (14)$$

$$\alpha = \sum_{i=1}^p c'_{i1}B_{i1}^{-1}b_i, \quad (15)$$

$$M_i = A_{i2} - A_{i1}B_{i1}^{-1}B_{i2}, \quad (16)$$

$$M_0 = D_0 - \sum_{i=1}^p A_{i1}B_{i1}^{-1}D_i, \quad (17)$$

$$b = b_0 - \sum_{i=1}^p A_{i1}B_{i1}^{-1}b_i. \quad (18)$$

Теперь можно записать сокращенную задачу в виде

$$\left. \begin{aligned} \max \{z - \alpha = \sum_{i=1}^p d'_i x_{i2} + d'_0 y\}, \\ \sum_{i=1}^p M_i x_{i2} + M_0 y = b, \\ x_{i2} \geq 0, \quad y \geq 0. \end{aligned} \right\} \quad (19)$$

Эта задача имеет лишь столько ограничений, сколько в исходной было связывающих. Ограничения в (19), совместно с уравнениями (10), задают то же самое множество допустимых решений, что и исходные ограничения (2) — (3). Однако поскольку сокращенная задача не содержит явно переменных x_{i1} , то в ней нет возможности и зафиксировать ограничения $x_{i1} \geq 0$.

Таким образом, эти ограничения релаксируются. Допустимое множество, задаваемое (19) и (10), очевидно, содержит допустимое множество исходной задачи, так что если сокращенная задача неразрешима, то неразрешима и исходная. Однако сокращенная задача может иметь неограниченное решение в случаях, когда исходная имеет конечное оптимальное решение. Для того чтобы исключить эту возможность, можно добавить в сокращенную задачу ограничение вида

$$\sum_{j=1}^p e_j x_{j2} + e_0 y \leq t, \quad (20)$$

где e_j — векторы со всеми компонентами, равными единице, а t достаточно большое положительное число. Если это ограничение является существенным при окончании работы алгоритма, то оптимум является неограниченным.

Пусть решение сокращенной задачи равно $(\{x_{i2}^0\}, Y^0)$. Подставляя это решение в (10), получим новые значения для x_{i1} :

$$x_{i1}^0 = B_{i1}^{-1} b_i - B_{i1}^{-1} B_{i2} x_{i2}^0 - B_{i1}^{-1} D_i Y^0. \quad (21)$$

Вектор $(\{x_{i1}^0, x_{i2}^0\}, Y^0)$ является решением модифицированной исходной задачи, в которой опущены условия $x_{i1} \geq 0$. Если окажется, что эти ограничения выполнены, то полученное решение является решением исходной задачи.

Теорема 1 (критерий оптимальности). *Решение $(\{x_{i1}^0, x_{i2}^0\}, y_0)$ является решением исходной задачи если и только если*

$$x_{i1}^0 \geq 0, \quad i = 1, \dots, p. \quad (22)$$

Преобразование сокращенной задачи. Если условия оптимальности не выполнены, то, следуя общей процедуре релаксации (§ 4.2), необходимо построить новую сокращенную задачу, в которой некоторые из нарушенных условий неотрицательности вводятся как обязательные, тогда как требования неотрицательности тех переменных x_{i2} , для которых $x_{i2}^0 > 0$, опускаются.

Рассмотрим произвольный вектор x_{j1} , обладающий только тем свойством, что по крайней мере одна из компонент x_{j1}^0 в ранее найденном решении отрицательна. Положим для определенности, что отрицательными являются первые l компонент x_{j1}^0 ,

$$(x_{j1}^0)_1 < 0, (x_{j1}^0)_2 < 0, \dots, (x_{j1}^0)_l < 0, l \geq 1. \quad (23)$$

Пусть x_{j2}^0 имеет положительными первые q компонент.

$$(x_{j2}^0)_1 > 0, (x_{j2}^0)_2 > 0, \dots, (x_{j2}^0)_q > 0, q \geq 0. \quad (24)$$

Для удобства перепишем вновь соотношения, связывающие x_{j_1} и x_{j_2} :

$$x_{j_1} + B_{j_1}^{-1} B_{j_2} x_{j_2} = B_{j_1}^{-1} b_i - B_{j_1}^{-1} D_{j_1} y. \quad (25)$$

Возможны два случая.

Случай 1. По крайней мере один элемент подматрицы, образованной первыми l строками и первыми q столбцами в $B_{j_1}^{-1} B_{j_2}$, отличен от нуля. В этом случае может быть проведена ведущая операция, использующая в качестве ведущего элемента произвольный ненулевой элемент указанной подматрицы. Такая операция преобразует (25), вводя в базис компоненту x_{j_2} , положительную в $x_{j_2}^0$, и выводя компоненту x_{j_1} , отрицательную в $x_{j_1}^0$. Эта замена приводит к новой базисной матрице $\hat{B}_{j_1}$, которая и используется для построения новой сокращенной задачи.

Случай 2. Подматрица, образуемая первыми l строками и первыми q столбцами $B_{j_1}^{-1} B_{j_2}$, содержит только нулевые элементы. В этом случае введение компонент из x_{j_1} вместо компонент x_{j_2} с помощью ведущей операции невозможно. Выберем некоторую отрицательную компоненту $x_{j_1}^0$, скажем, $(x_{j_1}^0)_k$, $1 \leq k \leq l$ (хорошим выбором может быть выбор наиболее отрицательной компоненты), и используем (25) для того, чтобы явно выразить условие неотрицательности этой переменной через x_{j_2} и y :

$$(x_{j_1})_k = \alpha_{jk} - \bar{b}_{jk} x_{j_2} - \bar{d}_{jk} y \geq 0, \quad (26)$$

где α_{jk} , $\bar{b}_{jk}$ и $\bar{d}_{jk}$ — k -е строки матриц $B_{j_1}^{-1} b_i$, $B_{j_1}^{-1} B_{j_2}$, $B_{j_1}^{-1} D_{j_1}$ соответственно. Это ограничение добавляется к сокращенной задаче (19).

Повторяя процедуру, указанную в случаях 1 или 2 для каждого вектора $x_{j_1}^0$, имеющего отрицательные компоненты, приходим к новой сокращенной задаче, в которой могут быть изменены матрицы B_{j_1} и могут быть добавлены ограничения вида (26). Если рассматриваемая итерация является не первой, то сокращенная задача может содержать ограничения типа (26), добавленные на предшествующих итерациях. Перед применением процедур, указанных в случаях 1 и 2, эти ограничения необходимо исследовать в соответствии с правилами, излагаемыми ниже.

Случай 3. Если ограничения (26) не являются существенными на оптимальном решении, т. е., если

$$\alpha_{jk} - \bar{b}_{jk} x_{j_2}^0 - \bar{d}_{jk} y^0 > 0, \quad (27)$$

то эти ограничения изымаются из сокращенной задачи. В силу теоремы 4.2.1 эта процедура не меняет оптимального решения задачи в целом.

Случай 4. Если ограничения (26) являются существенными, т. е. если

$$\alpha_{jk} - \bar{b}_{jk}x_{j2}^0 - \bar{d}_{jk}y^0 \neq 0, \quad (28)$$

то ограничение опускается в случае, когда возможно изменение в базисе, связанное с заменой $(x_{j1})_k$ на некоторую компоненту x_{j2} , которая положительна в x_{j2}^0 . При этом просматриваются элементы $\bar{b}_{jk}$, связанные с положительными компонентами x_{j2}^0 ; если имеется некоторый ненулевой элемент, то он используется в качестве ведущего, и после проведения ведущей операции соответствующее ограничение (26) опускается. Так как это ограничение было только записано в более удобной форме, оптимальное решение вновь остается неизменным.

Случай 5. Если выполняется (28), и все компоненты $\bar{b}_{jk}$, соответствующие положительным компонентам в x_{j2}^0 , равны нулю, то ограничения (26) сохраняются и, очевидно, условия (28) приобретают вид

$$\bar{d}_{jk}y^0 = \alpha_{jk}. \quad (29)$$

Требуется показать, что если решение $\{x_{i2}^0\}$, y^0 сокращенной задачи невырождено, то возможно добавление не более n_0 (числа связывающих переменных) таких уравнений. Для того чтобы

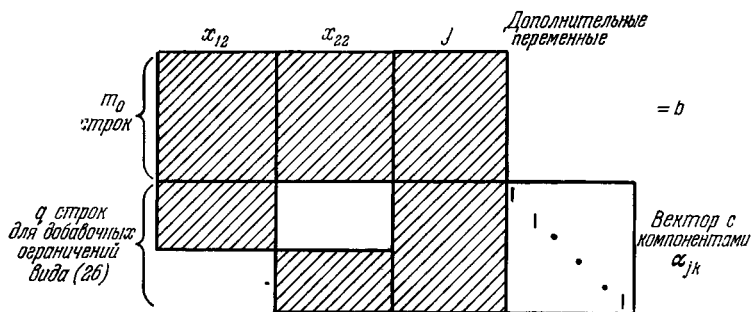


Рис. 4.2.

убедиться в этом, заметим, что в случаях 3 и 4 возникает преобразованная сокращенная задача, где опущены некоторые строки, не имеется дополнительных переменных в оптимальном базисе, но оптимальное решение является тем же самым, что и в исходной задаче. Структура этой преобразованной задачи для исходной, имеющей только два блока, показана на рис. 4.2. В ней последние q строк — это строки, остающиеся после применения процедур, указанных в случаях 3 и 4. Если исходная система (2) — (3) имеет полный ранг, то верхние m_0 строк независимы. Вместе с добавленными строками матрица имеет ранг $m_0 + q$.

Предположим теперь, что решение $\{x_{i2}^0\}$, y^0 имеет m_0+q положительных компонент, и рассмотрим базисную матрицу B^0 , ассоциированную с этим решением. Она должна иметь форму, показанную на рис. 4.3. Так как все базисные переменные положительны и $\bar{b}_{jh}x_{j2}^0=0$, то коэффициенты при x_{12} и x_{22} в последних q строках равны нулю. Если $q > n_0$, то, так как число столбцов, соответствующих переменным y , не превышает n_0 , последние q строк в B^0 зависимы, что невозможно. Следовательно, $q \leq n_0$.

Заметим, что если некоторая базисная переменная $x_{j2}^0=0$, то подматрица B^0 , соответствующая этим компонентам, может содержать некоторые ненулевые элементы. В этом случае, если имеются d столбцов, соответствующих нулевым компонентам x_{12}^0 и x_{22}^0 , мы сгруппируем их со столбцами, соответствующими y , и получим, что $q \leq n_0 + d$. Таким образом, в общем случае никогда не возникает более, чем $p+n_0+d$ (где p — число блоков в (2), n_0 — число связывающих переменных) добавочных ограничений типа (26) в сокращенной задаче. Это имеет место вследствие того, что может сохраниться не более n_0+d ограничений и добавиться не более, чем p .

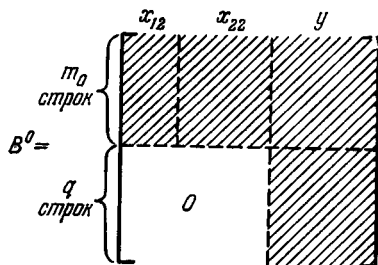


Рис. 4.3.

Формулировка алгоритма.

1. Выберем начальные базисные матрицы B_{ii} , $i=1, \dots, p$. Это может быть сделано произвольно или путем решения подзадач (8) и использования их оптимальных базисов в качестве B_{ii} . С помощью этих матриц строим сокращенную задачу (19).

2. Решим сокращенную задачу, что дает $\{x_{i2}^0\}$, y^0 . Вычислим x_{i1}^0 с помощью (21). Если $x_{i1}^0 > 0$, $i=1, \dots, p$, то решение $(\{x_{i1}^0, x_{i2}^0\}, y^0)$ оптимально.

3. Если $x_{i1}^0 \geq 0$, $i \in I$ и если имеются добавочные ограничения вида (26) в сокращенной задаче, то рассматриваем их с применением процедур, указанных в случаях 3—5.

4. Для каждого $i \in I$ применяем ведущую операцию, как в случае 1, либо добавляем ограничения, как в случае 2.

5. Используя новые базисные матрицы и ограничения, найденные на этапах 3—4, строим новую сокращенную задачу и переходим к шагу 2.

Сходимость. Так как описанный алгоритм является релаксационной процедурой, то приложимы теоремы § 4.2. В силу

теоремы 4.2—2 последовательность целевых функций, получаемая при оптимизации сокращенных задач, является монотонно невозрастающей. В то же время теорема 4.2—3 дает следующие условия для сходимости за конечное число итераций:

Теорема 2. Если требования неотрицательности положительных x_{i2} опускаются только на тех итерациях, когда целевая функция сокращенной задачи убывает, то алгоритм сходится за конечное число итераций либо к оптимальному решению задачи (1)—(4), либо к построению сокращенной задачи, не имеющей допустимого решения.

Так как ограничения, добавляемые к сокращенной задаче, не удовлетворяются на некотором оптимальном решении этой задачи, то целевая функция убывает на каждой итерации за исключением тех случаев, когда имеется какое-либо другое оптимальное решение, которое удовлетворяет всем новым ограничениям. Если в этом случае опускать ограничения, то возникает возможность заикливания, если не использовать схему возмущения для преодоления вырожденности.

Двойственная допустимость. Следующая теорема в сочетании с тем фактом, что алгоритм Риссера генерирует последовательность недопустимых решений исходной задачи, сходящуюся по целевой функции к максимуму сверху, показывает, что алгоритм Риттера является двойственным.

Теорема 3. Каждому решению $(\{x_{i1}^0, x_{i2}^0\}, y^0)$, полученному на итерации k , соответствует крайняя точка допустимого множества двойственной задачи (5)—(6), $\{u_i^0\}$ с равными целевыми функциями:

$$\sum_{i=1}^p b_i u_i^0 = c_0 y^0 + \sum_{i=1}^p (c_{i1} x_{i1}^0 + c_{i2} x_{i2}^0). \quad (30)$$

Доказательство. Решение $(\{x_{i1}^0, x_{i2}^0\}, y^0)$ является решением модифицированной исходной проблемы, задаваемой условиями (1)—(3) и требованиями неотрицательности:

$$x_{i2} \geq 0, \quad i = 1, \dots, p, \quad y \geq 0, \quad (31)$$

$$(x_{ik1})_{jk} \geq 0, \quad (32)$$

где (32) соответствует k -му ограничению вида (26). Двойственная к этой модифицированной исходной содержит равенство вместо каждого из неравенств (6), соответствующих компонентам $\{x_{i1}\}$, не ограниченным по знаку. По теореме двойственности, если модифицированная исходная имеет конечное оптимальное решение, то двойственная к ней также имеет его. Таким

образом, модифицированная двойственная достигает оптимума в крайней точке $\{u_i^0\}$, удовлетворяющей (30). Поскольку любая крайняя точка модифицированной двойственной является одновременно крайней точкой исходной двойственной (5) — (6), теорема доказана.

Вычислительные вопросы. Будучи релаксационной процедурой, этот алгоритм допускает много вариантов в выборе тех нарушенных ограничений, которые должны быть учтены в каждой итерации. В силу теоремы 4.2-4, если учитывается только одно наиболее сильно нарушенное условие неотрицательности (среди всех блоков), процедура идентична двойственному симплекс-методу. Однако блочная структура задачи позволяет легко учесть большее число ограничений, по крайней мере, если требуемые ведущие элементы отличны от нуля. В противном случае приходится добавлять в сокращенную задачу дополнительные ограничения, что делает ее решение более трудным. Таким образом, имеется возможность выбора между учетом на каждой итерации большего числа условий неотрицательности и увеличением размера сокращенной задачи.

Обобщение этого алгоритма на случай нелинейной вогнутой целевой функции и описание некоторого опыта вычислений в линейном случае можно найти в [9].

§ 4.4. ПРОЦЕДУРА РАСЧЛЕНЕНИЯ РОЗЕНА

4.4.1. Описание алгоритма

Розеном [4] был предложен метод расчленения для решения пары двойственных задач, имеющих следующий вид: двойственная задача:

$$\max \{v = \sum_{i=1}^p b'_i u_i + b'_0 y\}, \quad (1)$$

$$B'_i u_i + A'_i y \leq c_i, \quad i = 1, \dots, p, \quad (2)$$

прямая задача:

$$\min \{z = \sum_{i=1}^p c'_i x_i\}, \quad (3)$$

$$\sum_{i=1}^p A_i x_i = b_0, \quad (4)$$

$$B_i x_i = b_i, \quad i = 1, \dots, p, \quad (5)$$

$$x_i \geq 0, \quad i = 1, \dots, p. \quad (6)$$

Размерность матриц A_i — $m_i \times n_i$, размерность матриц B_i — $m_i \times n_i$. Описываемая далее процедура может быть рассматриваема как специализация метода Риттера, излагавшегося в предшествующем параграфе для задач, где имеются связывающие ограничения либо связывающие переменные. Следовательно, она также является релаксационной процедурой. Она генерирует последовательность недопустимых базисных решений прямой задачи и соответствующую последовательность допустимых базисных решений двойственной. Мы покажем далее, что в этом случае возможен ряд упрощений по сравнению с процедурой Риттера. Во-первых, не требуется вводить дополнительные ограничения в сокращенную задачу, так что эта задача будет иметь не более m_0 строк. Во-вторых, последовательные сокращенные задачи могут быть эффективно решены двойственным симплекс-методом, поскольку оптимальный базис на итерации i является двойственно-допустимым для задачи, возникающей на итерации $i+1$.

Во всем дальнейшем изложении в этом параграфе мы будем использовать следующее предположение.

Предположение 1. Матрица ограничений прямой задачи имеет полный ранг. Отсюда вытекает следующий результат.

Теорема 1. Если выполнено предположение 1, то каждая матрица B_i содержит неособую подматрицу B_{ii} размерности m_i .

Доказательство. Если B_i имеет ранг меньше, чем m_i , то соответствующие строки (5) оказываются зависимыми, что противоречит предположению 1. Следовательно, $n_i \geq m_i$ и B_i содержит желаемую подматрицу.

Мы можем также предположить, что $n_i > m_i$, поскольку, если $n_i = m_i$, то переменные x_i однозначно определены и могут быть исключены из рассмотрения.

Начальное множество базисных матриц B_{ii} всегда может быть найдено с помощью фазы 1 процедуры для каждого блока. Лучшее начальное приближение обычно может быть найдено путем решения одной из следующей пары подзадач:

двойственная подзадача:

$$\left. \begin{aligned} \max b'_{ii} u_i, \\ B'_{ii} u_i \leq c_i - A'_i y^*, \end{aligned} \right\} \quad (7)$$

прямая подзадача:

$$\left. \begin{aligned} \min c'_i x_i, \\ B_i x_i = b_i, \quad x_i \geq 0. \end{aligned} \right\} \quad (8)$$

Здесь y^* — вектор, при котором существует допустимая точка в (7). При любом способе выбора необходимо, чтобы начальные базисные матрицы были такими, что соответствующие им

базисные решения неотрицательны, и, следовательно, допустимы для i -го блока ограничений. Это условие выполняется на всех итерациях алгоритма. Как будет показано ниже в теореме 2, такой выбор гарантирует существование последовательности ведущих операций в каждом неоптимальном блоке, приводящих к желаемому выводу — вводу базисных переменных.

Предположение 2. Начальные базисные матрицы B_{i1} удовлетворяют условиям

$$B_{i1}^{-1}b_i \geq 0, \quad i = 1, \dots, p.$$

Базисные матрицы B_{i1} используются далее для преобразования прямой и двойственной задач следующим образом:

двойственная задача:

$$\max \left\{ \sum_i b'_i u_i + b'_0 y \right\},$$

$$B'_{i1} u_i + A'_{i1} y \leq c_{i1}, \quad (9)$$

$$B'_{i2} u_i + A'_{i2} y \leq c_{i2}, \quad (10)$$

прямая задача:

$$\min \sum (c'_{i1} x_{i1} + c'_{i2} x_{i2}),$$

$$\sum_i (A_{i1} x_{i1} + A_{i2} x_{i2}) = b_0, \quad (11)$$

$$B_{i1} x_{i1} + B_{i2} x_{i2} = b_i, \quad (12)$$

$$x_{i1} \geq 0, \quad x_{i2} \geq 0, \quad i = 1, \dots, p.$$

Как и в § 4.3, релаксируя ограничения $x_{i1} \geq 0$ и используя (12) для исключения x_{i1} из связывающих ограничений (11), приходим к сокращенной задаче. Имеем

$$x_{i1} = B_{i1}^{-1}b_i - B_{i1}^{-1}B_{i2}x_{i2}. \quad (13)$$

Следовательно, сокращенная задача имеет такой вид:

$$\left. \begin{aligned} \min \sum_i d'_i x_{i2}, \\ \sum_i M_i x_{i2} = \hat{b}_0, \quad x_{i2} \geq 0, \end{aligned} \right\} \quad (14)$$

где

$$d'_i = c'_{i2} - c'_{i1} B_{i1}^{-1} B_{i2}, \quad (15)$$

$$M_i = A_{i2} - A_{i1} B_{i1}^{-1} B_{i2}, \quad (16)$$

$$\hat{b}_0 = b_0 - \sum_{i=1}^p A_{i1} B_{i1}^{-1} b_i. \quad (17)$$

Матрица ограничений имеет только m_0 строк. Как и в § 4.3, допустимое множество, определяемое (14), (13), содержит в себе допустимое множество исходной задачи, так что если сокращенная задача не имеет допустимых решений, то неразрешима и исходная. Сокращенная задача может иметь неограниченное решение в отличие от исходной, поэтому можно вновь использовать ограничение вида

$$\sum_i e_{ij} x_{j2} \leq t,$$

где t — достаточно большое положительное число.

Пусть $\{x_{i2}^0, i = 1, \dots, p\}$ решение сокращенной задачи. Подставляя эти векторы в (13), находим соответствующие значения x_{i1} :

$$x_{i1}^0 = B_{i1}^{-1} b_i - B_{i1}^{-1} B_{i2} x_{i2}^0, \quad i = 1, \dots, p. \quad (18)$$

Согласно предшествующим результатам, касающимся релаксации, это решение оптимально, если и только если оно допустимо.

Теорема 2 (критерий оптимальности). *Векторы $\{x_{i1}^0, x_{i2}^0\}$ являются решением исходной задачи, если и только если*

$$x_{i1}^0 \geq 0, \quad i = 1, \dots, p. \quad (19)$$

Если критерий оптимальности не выполняется, то, вновь следуя общей релаксационной стратегии (§ 4.2), необходимо учесть по крайней мере одно из нарушенных ограничений неотрицательности и при этом релаксировать ограничения неотрицательности на положительные переменные. В задаче, где присутствовали как связывающие переменные, так и связывающие ограничения, не удавалось всегда выполнить это преобразование с помощью ведущей операции и приходилось вводить добавочные ограничения. Здесь же всегда существует последовательность ведущих операций, приводящая к желаемому решению.

Теорема 3 (Розен [4]). *Если для некоторого i x_{i1}^0 имеет отрицательные компоненты, то может быть построена последовательность ведущих операций в i -м блоке, осуществляющих ввод в базис компоненты x_{i2} , бывшей положительной в x_{i2}^0 , и вывод компоненты x_{i1} , которая была отрицательна в x_{i1}^0 .*

Доказательство этой теоремы требует использования следующей леммы.

Лемма 1 [5]. *Рассмотрим линейную систему $Ax = b$, где A — матрица размера $m \times n$, $n \geq m$ и $\text{rank}(A) = m$. Если эта система имеет неотрицательное решение, то она имеет и базисное неотрицательное решение.*

Доказательство теоремы 3. Для простоты обозначений опустим индекс блока i и сконцентрируем внимание на

некотором одном неоптимальном блоке, используя далее дополнительные индексы для обозначения компонент x_1 , x_2 и т. д.

Мы располагаем двумя решениями системы

$$B_1 x_1 + B_2 x_2 = b, \quad x_1 \geq 0, \quad x_2 \geq 0. \quad (20)$$

Они таковы:

допустимое

$$\begin{bmatrix} x_{B_1} \\ \dots \\ 0 \end{bmatrix} \quad (21)$$

и недопустимое

$$\begin{bmatrix} x_1^0 \\ \dots \\ x_2^0 \end{bmatrix},$$

где

$$x_{B_1} = B_1^{-1} b. \quad (22)$$

Первое решение неотрицательно, второе имеет отрицательные компоненты в x_1^0 . Все векторы вида

$$(1 - \theta) \begin{bmatrix} x_{B_1} \\ \dots \\ 0 \end{bmatrix} + \theta \begin{bmatrix} x_1^0 \\ \dots \\ x_2^0 \end{bmatrix} \quad (23)$$

также удовлетворяют условиям $B_1 x_1 + B_2 x_2 = b$, $x_2 \geq 0$. При $\theta = 0$ это решение удовлетворяет (20), в то время как при $\theta = 1$ оно недопустимо. Пусть

$$I = \{j / x_{1j}^0 < 0\}. \quad (24)$$

Наибольшее θ , при котором (23) остается неотрицательным, ограничено компонентами, имеющими индексы $j \in I$. Полагая эти компоненты в (23) равными нулю, получаем

$$x_{B_{1j}} + \theta (x_{1j}^0 - x_{B_{1j}}) = 0, \quad j \in I \quad (25)$$

или

$$\theta_j = \frac{x_{B_{1j}}}{x_{B_{1j}} - x_{1j}^0}. \quad (26)$$

Заметим, что $0 \leq \theta_j \leq 1$. Наибольшее θ , при котором (23) остается неотрицательным, равно

$$\theta_s = \min_{j \in I} \theta_j. \quad (27)$$

Пусть $K = \{k / x_{2k}^0 > 0\}$. Заметим, что K должно быть непусто, так как если все $x_{2k}^0 = 0$, то $x_1^0 = B_1^{-1} b \geq 0$, что противоречит нашей

гипотезе, что x_1^0 имеет отрицательные компоненты. Когда $\theta = \theta_s$, вектор (23) удовлетворяет уравнению

$$B_1(x_{B_1} + \theta_s(x_1^0 - x_{B_1})) + \sum_{k \in K} B_2^k(\theta_s x_{2k}^0) = b, \quad (28)$$

где B_2^k есть k -й столбец B_2 . По определению θ_s , коэффициент при столбце B_1^s (s -й столбец B_1) равен нулю. Поскольку матрица B_1 не особая и имеет размер $m \times m$, матрица $[B_1 : B_2]$ имеет ранг m . Опустим столбец B_1^s в (28) и проверим ранг получившейся системы. Умножая (28) на B_1^{-1} , получаем

$$\sum_{\substack{i=1 \\ i \neq s}}^m v_i^0 e_i + \sum_{k \in K} \bar{B}_2^k(\theta_s x_{2k}^0) = B_1^{-1}b, \quad (29)$$

где v_i^0 есть i -я компонента $x_{B_1} + \theta_s(x_1^0 - x_{B_1})$, e_i — i -й единичный вектор и $\bar{B}_2^k$ есть k -й столбец $B_1^{-1}B_2$. Система имеет тот же самый ранг, что и система (28), без B_1^s . Более того, по крайней мере один из векторов $\bar{B}_2^k$, $k \in K$, скажем, $\bar{B}_2^{k*}$, должен иметь ненулевую s -ю компоненту, поскольку для $j \in I$ вектор

$$x_1^0 = x_{B_1} - B_1^{-1}B_2 x_2^0 \quad (30)$$

имеет отрицательные компоненты. Так как $s \in I$, если все компоненты $B_1^{-1}B_2$ в строке s и столбцах $k \in K$ равны нулю, то

$$x_{1s}^0 = x_{B_1s} \geq 0, \quad (31)$$

что и приводит к противоречию. Таким образом, векторы e_i , $i = 1, \dots, m$, $i \neq s$ и $\bar{B}_2^{k*}$ являются независимыми, и система (28) с опущенным B_1^s имеет ранг m . Поскольку эта система имеет неотрицательное решение, то в силу леммы 1 она имеет и базисное неотрицательное решение. Любое такое решение не включает B_1^s и может включать только столбцы B_2^k , где $k \in K$, так что теорема доказана.

Заметим, что предшествующая теорема не утверждает, что существует единственная ведущая операция, приводящая к новому базису $\hat{B}_{i1}$. Однако эти базисы могут быть найдены решением задачи

$$\left. \begin{aligned} \min x_{1s}, \\ \sum_{k=1}^m B_1^k x_{1k} + \sum_{k \in K} B_2^k x_{2k} = b, \\ x_{1k} \geq 0, \quad x_{2k} \geq 0, \end{aligned} \right\} \quad (32)$$

где s — индекс, выбираемый в силу (27). Начальный базис в этой задаче есть B_1 , и имеется оптимальное решение, в котором x_i не базисная. Для того чтобы решить эту задачу, должно потребоваться небольшое число ведущих операций. Конечно, такая задача должна решаться для каждого неоптимального блока, в котором желательно учесть нарушенные ограничения.

Решение задач (32) дает новые базисные матрицы $\hat{B}_i$ для каждого неоптимального блока. Далее с их помощью строится новая сокращенная задача, и начинается новый цикл. Поскольку алгоритм есть частная реализация релаксационной процедуры, сходимость за конечное число итераций следует из теоремы 4.3—2 при исходных условиях. Здесь, однако, нет необходимости учета нарушений неотрицательности путем включения дополнительных ограничений типа (4.3—26) в сокращенную задачу. Если никаких дополнительных ограничений не вводится, то предположение, что каждая сокращенная задача не имеет других оптимальных решений, удовлетворяющих вновь учтенным условиям неотрицательности, гарантирует сходимость за конечное число шагов.

Формулировка алгоритма.

1. Выберем начальные допустимые базисы для каждого блока.

2. Используя текущее множество базисов, построим и решим сокращенную задачу (14), получая оптимальные решения $\{x_{i2}^0\}$.

3. Вычислим x_{i1}^0 по формуле (18).

4. Если $x_{i1}^0 \geq 0$, $i = 1, \dots, p$, то $\{x_{i1}^0, x_{i2}^0\}$ является решением исходной задачи (3) — (6).

5. В противном случае для каждого i такого, что $x_{i1}^0 \not\geq 0$, вычислим индекс s согласно (27) и решим линейную задачу (32).

6. Используя новые базисные матрицы $\hat{B}_i$, полученные при решении (32), перейдем к шагу 2.

Генерация двойственно-допустимых точек. Сокращенная задача, дополненная уравнениями (13), эквивалентна исходной задаче (3) — (6), в которой релаксированы (опущены) ограничения $x_{i1} \geq 0$. Двойственная этой задаче имеет ограничения вида (9), но выполняющиеся как равенства, т. е.

$$B'_{i1}u_i + A'_{i2}y = c_{i1}, \quad i = 1, \dots, p. \quad (33)$$

Они могут быть использованы для нахождения u_i через y :

$$u_i = (B'_{i1})^{-1} c_{i1} - (B'_{i1})^{-1} A'_{i2} y. \quad (34)$$

Легко доказать, что подстановка (34) в (10) приводит к задаче, двойственной к сокращенной задаче:

$$\max b'_0 y, \quad M'_i y \leq d_i, \quad i = 1, \dots, p. \quad (35)$$

Пусть y^0 — вектор двойственных переменных, соответствующих оптимальному решению $\{x_{i2}^0\}$ сокращенной задачи. Подставляя y^0 в (34), найдем значения остальных двойственных переменных u_i :

$$u_i^0 = (B'_{i1})^{-1}(c_{i1} - A'_{i1}y^0), \quad i = 1, \dots, p. \quad (36)$$

Теорема 4. Векторы $\{u_i^0\}$ соответствуют крайней точке допустимого множества двойственной задачи (1) — (2). Последовательность значений целевых функций (1), задаваемая этими решениями в последовательных итерациях, является неубывающей.

Доказательство. Решение $\{u_i^0\}$, y^0 является допустимым по построению. Для того чтобы показать, что оно задает крайнюю точку, заметим, что, будучи вектором двойственных переменных, y^0 удовлетворяет m_0 ограничениям (35) как равенствам, и $m_0 \times m_0$ -подматрица, ассоциированная с этими равенствами, является неособой. Обозначая M эту подматрицу, и d — соответствующую правую часть, получим

$$My^0 = d. \quad (37)$$

Таким образом, $\{u_i^0\}$, y^0 удовлетворяет как равенствам подмножеству ограничений двойственной задачи, имеющих вид

$$\left. \begin{array}{l} u_1 \quad \quad \quad + (B'_{11})^{-1} A'_{11} y = (B'_{11})^{-1} c_{11}, \\ u_2 \quad \quad \quad \vdots \quad \quad \quad \vdots \\ \quad \quad \quad \vdots \quad \quad \quad \vdots \\ u_p + (B'_{p1})^{-1} A'_{p1} y = (B'_{p1})^{-1} c_{p1}, \\ \quad \quad \quad My = d. \end{array} \right\} \quad (38)$$

Эта система имеет столько же уравнений, сколько и неизвестных, и ее матрица является верхней блочно-треугольной с неособыми блоками. Следовательно, эта матрица неособая. Согласно [6] это доказывает, что $\{u_i^0\}$, y^0 является крайней точкой.

Докажем далее монотонность. Вектор $\{u_i^0\}$, y^0 является решением двойственной к модифицированной исходной задаче, отличающейся релаксацией ограничений $x_{i1} \geq 0$. Целевая функция $v^0 = \sum_i b_i u_i^0 + b_0 y^0$ равна, таким образом, оптимальному значению целевой функции z^0 этой модифицированной задачи. По теореме 4.2 — 2 последовательность z^0 является неубывающей, что и завершает доказательство.

Заметим, что доказанная теорема является частным случаем общего результата, полученного в § 4.2. Аналогичный комментарий приложим и к теореме 4.3—3.

4.4.2. Соображения по поводу вычислений

Выбор ограничений, которые должны быть учтены. Как и в методе Риттера, существует много вариантов выбора того, какие из нарушенных ограничений стоит учесть на текущей итерации. Комментарии, которые были сделаны в § 4.3, приложимы и здесь, за исключением того факта, что размер сокращенной задачи остается постоянным и не зависящим от числа учитываемых ограничений неотрицательности. Это делает более привлекательным учет на каждой итерации возможно большего числа ограничений. Если принять эту точку зрения, то в задаче (32) целесообразно заменить целевую функцию на

$$z = \sum_{j \in I} x_{1j}, \quad (39)$$

где I определено в силу (24). Вновь каждая из таких задач должна решаться для каждого неоптимального блока.

Решение последовательных сокращенных задач. При использовании алгоритма расчленения большая часть вычислительного времени, вероятно, затрачивается на решение сокращенной задачи. Поэтому важно, чтобы информация о решении на итерации k использовалась эффективно при решении задачи на итерации $k+1$. Следующие результаты показывают, как это может быть сделано.

Для простоты мы сначала рассмотрим задачу с одним блоком:

$$\left. \begin{aligned} \min \{z = c_1 x_1 + \hat{c}_2 x_2\}, \\ m_0 \text{ строк } \{A_1 x_1 + \hat{A}_2 x_2 = \hat{b}_0\}, \\ m_1 \text{ строк } \{B_1 x_1 + \hat{B}_2 x_2 = \hat{b}_1, x_1 \geq 0, x_2 \geq 0\}, \end{aligned} \right\} \quad (40)$$

где B_1 — базисная матрица размерности m_1 . Заметим, что в таком виде может быть записана любая задача ЛП, так что алгоритм расчленения Розена является вполне общим. Используя матрицу B_1 для исключения x_1 из первых m_0 равенств, и из целевой функции, приходим к следующей задаче:

$$\min c_2 x_2, \quad A_2 x_2 = b_0, \quad x_1 + B_2 x_2 = b_1, \quad x_1 \geq 0, \quad x_2 \geq 0. \quad (41)$$

Сокращенная задача получается из нее, если опустить ограничения $x_1 \geq 0$:

$$\min c_2 x_2, \quad A_2 x_2 = b_0, \quad x_2 \geq 0. \quad (42)$$

Пусть x_2^0 — решение этой задачи и пусть B_0 — базисная матрица, соответствующая x_2^0 . Для простоты предположим, что B_0 состоит из первых m_0 столбцов A_2 :

$$B_0 = [A_2^1, \dots, A_2^{m_0}]. \quad (43)$$

Пусть

$$x_2 = (y; w), \quad (44)$$

где y содержит базисные компоненты x_2^0 , а w — небазисные. Запишем сокращенную задачу в канонической форме, разрешив ограничения относительно базисных переменных:

$$\left. \begin{aligned} y + Ew &= y^0, \\ -z + \bar{C}w &= -z^0, \\ x + Cy + Dw &= b, \end{aligned} \right\} \quad (45)$$

где $x \equiv x_1$, $b \equiv b_1$. Предположим теперь, что компонента r вектора

$$x^0 = b - Cy^0 \quad (46)$$

отрицательна, и что имеется единственная ведущая операция, приводящая к вводу в базис x_r вместо некоторой переменной y_s , имевшей положительное значение в y^0 . Если провести эту операцию над первыми m_0 строками (45), то y_s исключится из сокращенной задачи, x_r введется, и матрица сокращенной задачи изменится соответствующим образом. Пусть новая базисная матрица есть $\bar{B}$. Ниже доказано, что $\bar{B}$ определяет те же самые двойственные переменные, что и B_0 , и что она является оптимальным, но недопустимым базисом для новой сокращенной задачи. Этот результат сохраняет силу, когда произвольное число компонент y заменяется в базисе компонентами x . Отсюда следует, что новая сокращенная задача может быть эффективно решена при использовании $\bar{B}$ в качестве начального базиса в двойственном симплекс-методе.

Важные соотношения между последовательными сокращенными задачами дает теорема, являющаяся обобщением ранее известного результата Гасса [7]:

Теорема 5. Пусть проведено некоторое число ведущих операций над матрицей задачи (45), при которых некоторое число компонент y , положительных в y^0 , заменяется одной или более компонентами x , по крайней мере одна из которых отрицательна в x^0 . Если $\bar{B}$ означает результат преобразования оптимальной базисной матрицы B_0 , предшествующей сокращенной задаче, то (а) $\bar{B}$ определяет те же самые двойственные переменные, что и B_0 , откуда следует, что всем небазисным столбцам в новой сокращенной задаче соответствуют неотрицательные характеристические разности. Таким образом, $\bar{B}$ является оптимальным базисом. (б). Базисное решение, ассоциированное с $\bar{B}$, состоит из компонент $y_i = y_i^0$, если y_i входит в базис, и $x_j = x_j^0$, если x_j также является базисной. Следовательно, это решение недопустимо только по компонентам $x_j^0 < 0$.

Так как эта операция оставляет z строку неизменной, новый базис также удовлетворяет условиям оптимальности $\bar{c}_i \geq 0$. Матрица, обратная к новой базисной, такова:

$$\bar{B}^{-1} = E_{rs} B_0^{-1}. \quad (48)$$

Последняя строка в $\bar{B}^{-1}$ и B_0^{-1} , очевидно, одна и та же. Именно эта строка и содержит двойственные переменные, так что пункт (а) теоремы доказан для одной ведущей операции. Та же схема непосредственно распространяется и на случай нескольких ведущих операций. Новая каноническая форма получается умножением исходной на произведение элементарных матриц, каждая из которых отличается от единичной только одной строкой. Обратная матрица, соответствующая новому базису, такова:

$$\bar{B}^{-1} = E_{r_k s_k} E_{r_{k-1} s_{k-1}} \dots E_{r_1 s_1} B_0^{-1}. \quad (49)$$

Вновь замечаем, что все столбцы, соответствующие переменным y_i , имеют нулевые компоненты в z -й строке, так что эта строка не изменяется при ведущих операциях, и каждая элементарная матрица в последней строке содержит единичный вектор $u_{m_0} + 1$. Таким образом, $\bar{B}^{-1}$ определяет те же самые двойственные переменные, что и B_0^{-1} , и является оптимальным базисом.

Для того, чтобы доказать пункт (b), мы будем рассуждать следующим образом. При $w=0$ уравнение (45) имеет единственное решение $y=y^0$, $z=z^0$, $x=x^0=b-Cy^0$. После проведения любого числа ведущих операций это решение остается единственным (при $w=0$), какова бы ни была форма преобразуемой системы. Следовательно, любая переменная x_i , введенная в первые m_0 строк (45), сохраняет свое значение x_i^0 , а переменная y_i — значение y_i^0 .

Обобщение на случай, когда число блоков больше единицы. Для рассмотрения многоблочной задачи достаточно ограничиться случаем двух блоков:

минимизировать z при ограничениях

$$\left. \begin{aligned} \left[\begin{array}{ccc} y_1 & & \\ \cdot & \cdot & \cdot \\ 0 & & \end{array} \right] + \left[\begin{array}{ccc} E_{11} & & \\ \cdot & \cdot & \cdot \\ E_{21} & & \end{array} \right] w_1 + \left[\begin{array}{ccc} 0 & & \\ \cdot & \cdot & \cdot \\ y_2 & & \end{array} \right] + \left[\begin{array}{ccc} E_{12} & & \\ \cdot & \cdot & \cdot \\ E_{22} & & \end{array} \right] w_2 = \left[\begin{array}{ccc} y_1^0 & & \\ \cdot & \cdot & \cdot \\ y_2^0 & & \end{array} \right], \\ \left. \begin{aligned} \bar{c}_1 w_1 & & + \bar{c}_2 w_2 - z & = -z^0, \\ x_1 + C_1 y_1 + D_1 w_1 & & & = b_1, \\ & x_2 + C_2 y_2 + D_2 w_2 & = b_2. \end{aligned} \right\} \quad (50)$$

Первая группа ограничений представляет собой сокращенную задачу в канонической форме, где переменные $y = (y_1, y_2)$ вы-

ражены через небазисные. Как и ранее, очевидно, что ведущие операции, приводящие к замене переменных x_i на y_i , не влияют на строку, содержащую характеристические разности, поскольку y_1 и y_2 имеют в ней нулевые коэффициенты. Таким образом, новая каноническая форма остается двойственно-допустимой, т. е. оптимальной. Благодаря блочной структуре, ведущие операции в одном блоке не затрагивают подматриц и векторов, ассоциированных с другим. Ввиду этого, матрица, обратная новой базисной для сокращенной задачи, $\bar{B}^{-1}$, представима через старую следующим образом:

$$\bar{B}^{-1} = \left[\begin{array}{cc|c} E_1 & & 0 \\ & E_2 & \\ \hline & & B_0^{-1} \\ 0 & & 1 \end{array} \right] \quad (51)$$

где E_1 и E_2 представляют собой произведение элементарных матриц типа (47), отличающихся от единичной только одной строкой. Поскольку (m_0+1) -е строки матриц $\bar{B}^{-1}$ и B_0^{-1} идентичны, эти матрицы определяют те же самые двойственные переменные, что и доказывает пункт а) теоремы 5 в многоблочном случае. Обоснование справедливости пункта б) остается тем же самым, что и для задач с одним блоком. Таким образом, теорема 5 доказана.

Нетрудно показать, что теорема 5 без существенных изменений приложима к алгоритму Риттера (§ 4.3). Небольшие модификации должны быть сделаны для учета того факта, что в этом алгоритме изменяется число ограничений в сокращенной задаче.

В силу полученных результатов двойственный симплекс-метод является естественным кандидатом на решение сокращенных задач. Если осуществляется только одна ведущая операция (по всем блокам), то начальный базис окажется двойственно-допустимым, а ограничения исходной задачи нарушаются только по одной компоненте. Следовательно, для решения сокращенной задачи будет требоваться малое число итераций двойственного симплекс-метода. Чем больше осуществлено ведущих операций, тем больше возникает нарушений неотрицательности, больше требуется итераций, но зато и достигаются большие изменения в целевой функции новой сокращенной задачи. При выборе между двумя указанными альтернативами стоит учесть, что для больших задач данные, над которыми не производится текущих преобразований, вероятно, будут сохраняться вне

оперативной памяти, во внешних накопителях. Переходы между решением сокращенной задачи и операциями в каждом блоке, таким образом, будут, вероятно, относительно медленными. Ввод данных для решения также может быть связан с большими затратами времени. Поэтому для того, чтобы уменьшить общие затраты времени на вычисление, представляется целесообразным производить за один цикл возможно больше ведущих операций. Это должно уменьшать число циклов и, следовательно, число обращений к внешней памяти.

Интерпретация с точки зрения процедур закрепления. Хотя алгоритмы этого и предшествующего параграфов описывались как релаксационные процедуры, им можно дать интерпретацию и в терминах процедуры закрепления. Как указывалось ранее, релаксация условий неотрицательности $x_{ij} \geq 0$ приводит к тому, что соответствующие двойственные ограничения должны выполняться как равенство, т. е. имеет место закрепление. Мы уже показывали, что последовательность допустимых двойственных решений, получаемая при решении этих равенств, является «улучшающейся». Действительно, нетрудно показать, что если релаксация применена к прямой задаче (3) — (6), то получаемые прямые и двойственные решения эквивалентны тем, которые были бы получены, если бы было проведено закрепление в двойственной задаче (1) — (2). Мы должны только обеспечить, чтобы правило выбора множества V в обоих процедурах было одним и тем же: если $x_{ij}^0 < 0$ и условие $x_{ij} \geq 0$ учитывается, то соответствующее двойственное ограничение, которое было закреплено, учитывается как неравенство, и наоборот.

Отметим лишь, что если в линейном случае применение закрепления приводит к тем же самым алгоритмам, что и релаксация, то в нелинейном случае (см. § 6.2) закрепление может оказаться более полезным.

4.4.3. Вычислительный опыт

Имеется некоторый вычислительный опыт применения алгоритма, описанный Розеном [4]. Были решены шесть проблем на ЦВМ IBM 7090/94. Размерность задач, плотность заполнения матриц и данные о числе итераций и времени решения приведены в таблице 4.4.1. Задача с 15 блоками имела подматрицы B_i , содержащие более 50% ненулевых элементов. Связывающие матрицы A_i были заполнены на 100%. Элементы матриц и векторы c_i генерировались датчиком случайных чисел, равномерно распределенных на отрезке (0; 100). Векторы b_i в правых частях выбирались так, чтобы обеспечить существование решений для блоков. Дополнительные данные о решении этой задачи приведены в таблице 4.4.2.

Таблица 4.4.1

Сводка результатов решения тестовых задач

Название	Блоки	Размер ограничения	Переменные	Число нул. элементов матриц	Плотность матриц %	Число циклов для решения	Время (мин.) решения
РР, задача 1	1	4	5	19	95	2	1,3
Бил, 2 блока	2	9	15	39	29	4	2,0
Розен, 2 блока	2	60	100	1660	27,6	7	3,6
Бендерс, 6 блоков	6	71	164	2490	19,3	14	6,0
Розен, 10 блоков	10	220	420	8220	8,9	39	75
Розен, 15 блоков со случайными элементами	15	930	1,200	72000	6,4	16	179

Таблица 4.4.2

Процесс решения для 15-блочной задачи со случайными элементами

Цикл	Число оптимальных блоков	Число $x_{i1} < 0$	Значение функции	Цикл	Число оптимальных блоков	Число $x_{i1} < 0$	Значение функции
1			39 365,24	9	3	56	40 395,18
2	2	121	39 968,80	10	7	46	40 440,69
3	*)	*)	40 109,08	11	8	17	40 470,01
4	4	70	40 132,92	12	7	18	40 478,85
5	3	72	40 167,84	13	10	7	40 486,65
6	1	151	40 258,11	14	11	6	40 488,97
7	2	57	40 335,48	15	12	3	40 489,39
8	4	92	40 367,09	16	15	0	40 489,55

*) Повторный старт.

Применение этого алгоритма к транспортным задачам описано в [8], где приведены и результаты вычислений.

4.4.4. Пример применения метода расчленения

Рассмотрим задачу $\min(-x_1 - x_2 - 2x_5 - x_6)$ при ограничениях

$$\begin{array}{rcll}
 x_1 + 2x_2 & + 2x_5 + x_6 & + x_{10} & = 40, \\
 x_1 + x_2 & + 4x_5 + 2x_6 & + x_{11} & = 50, \\
 x_1 + 3x_2 + x_3 & & & = 30, \\
 2x_1 + x_2 & + x_4 & & = 20, \\
 & x_5 & + x_7 & = 10, \\
 & & x_6 & + x_8 & = 10, \\
 & x_5 + x_6 & & + x_9 & = 15. \\
 & x_i \geq 0, & \forall i.
 \end{array}$$

Задача получена из примера, приведенного в § 2.5, путем добавления второго ограничения.

Цикл 1. Для того чтобы получить начальные базисные матрицы, решим следующие подзадачи:

$$\begin{array}{cc} \text{Блок 1} & \text{Блок 2} \\ \min(-x_1 - x_2) & \min(-2x_5 - x_6) \end{array}$$

при ограничениях

$$\begin{array}{rcl} x_1 + 3x_2 + x_3 & = & 30, \\ 2x_1 + x_2 + x_4 & = & 40, \\ x_i & \geq & 0, \end{array} \quad \begin{array}{rcl} x_5 + x_7 & = & 10, \\ x_6 + x_8 & = & 10, \\ x_5 + x_6 + x_9 & = & 15, \\ x_i & \geq & 0. \end{array}$$

Решение:

$$\begin{array}{l} x_1 = 6, \quad x_2 = 8, \quad x_5 = 10, \quad x_6 = 5, \quad x_8 = 5, \\ x_3 = x_4 = 0, \quad x_7 = x_9 = 0. \end{array}$$

Начальные базисные матрицы и им обратные таковы:

$$\begin{array}{l} B_{11} = \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix}, \quad B_{21} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \\ B_{11}^{-1} = \frac{1}{5} \begin{bmatrix} -1 & 3 \\ 2 & -1 \end{bmatrix}, \quad B_{21}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix}. \end{array}$$

Таким образом, задача может быть приведена к канонической форме с выделением базисных переменных в каждом блоке путем применения ведущих операций в этих базисах. В результате получается следующая система:

$$\begin{array}{cccccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & \\ & & 1/5 & 2/5 & & & 1 & & 1 & & & = z + 39, \\ & & -3/5 & -1/5 & & & -1 & & -1 & 1 & & = -7, \\ & & -1/5 & -2/5 & & & -2 & & -2 & & 1 & = -14, \\ 1 & & -1/5 & 3/5 & & & & & & & & = 6, \\ & 1 & 2/5 & -1/5 & & & & & & & & = 8, \\ & & & & 1 & & 1 & & & & & = 10, \\ & & & & & & 1 & 1 & -1 & & & = 5, \\ & & & & & 1 & -1 & & 1 & & & = 5. \end{array}$$

Начальная сокращенная задача получается из верхних трех строк.

Сокращенная задача 1:

$$\min \{ \hat{z} = \frac{1}{5} x_3 + \frac{2}{5} x_4 + x_7 + x_9 \}$$

при ограничениях

$$\begin{aligned} -\frac{3}{5}x_3 - \frac{1}{5}x_4 - x_7 - x_9 + x_{10} &= -7, \\ -\frac{1}{5}x_3 - \frac{2}{5}x_4 - 2x_7 - 2x_9 + x_{11} &= -14, \\ x_i &\geq 0. \end{aligned}$$

Оптимальная каноническая форма имеет следующий вид:

x_3	x_4	x_7	x_9	x_{10}	x_{11}	$-\hat{z}$	Постоянные
1	0	0	0	-2	0	0	0
0	$1/5$	1	1	$1/5$	$-3/5$	0	7
0	$1/5$	0	0	$1/5$	$2/5$	1	-7

Таким образом,

$$x_3^0 = 0, x_7^0 = 7, \text{ остальные } x_i^0 = 0.$$

Проверка на оптимальность.

Подставляя x_i^0 в каждый из преобразованных блоков, получаем $x_1^0 = 6$, $x_2^0 = 8$, $x_5^0 = 3$, $x_6^0 = 12$, $x_8^0 = -2$ со значением целевой функции $z^0 = -32$. Поскольку $x_8^0 < 0$ — решение не оптимально.

Ведущее преобразование в блоке 2.

Поскольку x_7 единственная положительная небазисная переменная в блоке 2, то x_8 необходимо вывести из базиса, а x_7 ввести. Преобразование приводит к следующей форме сокращенной задачи:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	
		$1/5$	$2/5$				-1	2			$= z + 34,$
		$-3/5$	$-1/5$				1	-2	1		$= -2,$
		$-1/5$	$-2/5$				2	-4		1	$= -4,$
1		$-1/5$	$3/5$								$= 6,$
	1	$2/5$	$-1/5$								$= 8,$
				1			-1	1			$= 5,$
					1		1	-1			$= 5,$
					1		1				$= 10.$

Сокращенная задача 2:

$$\min \{ \frac{1}{5}x_3 + \frac{2}{5}x_4 - x_8 + 2x_9 \}$$

при ограничениях

$$\begin{aligned} -\frac{3}{5}x_3 - \frac{1}{5}x_4 + x_8 - 2x_9 + x_{10} &= -2, \\ -\frac{1}{5}x_3 - \frac{2}{5}x_4 + 2x_8 - 4x_9 + x_{11} &= -4, \\ x_i &\geq 0. \end{aligned}$$

Преобразование предшествующего оптимального базиса дает

$$\bar{B} = \begin{bmatrix} x_3 & x_4 & -\hat{z} \\ -\frac{3}{5} & 1 & 0 \\ -\frac{1}{5} & 2 & 0 \\ \frac{1}{5} & -1 & 1 \end{bmatrix}.$$

Записываем далее сокращенную задачу в канонической форме с выделением переменных:

x_1	x_4	x_5	x_6	x_{10}	x_{11}	$-\hat{z}$	Постоянные
1				-2	1		0
	$-\frac{1}{5}$	1	-2	$-\frac{1}{5}$	$\frac{3}{5}$		-2
	$\frac{1}{5}$			$\frac{1}{5}$	$\frac{3}{5}$	1	-2

Так как все $\bar{c}_j \geq 0$, $\bar{B}$ является двойственно-допустимым, как было показано ранее, с единственным нарушением неотрицательности. Производя единственную операцию двойственного симплекс-метода над приведенной выше таблицей, приходим к следующей оптимальной таблице:

x_1	x_4	x_5	x_6	x_{10}	x_{11}	$-\hat{z}$	Постоянные
1				-2	1		0
	$\frac{1}{10}$	$-\frac{1}{2}$	1	$\frac{1}{10}$	$\frac{3}{10}$		1
	$\frac{1}{5}$			$\frac{1}{5}$	$\frac{1}{5}$	1	-2

Оптимальное решение таково: $x_5^0 = 1$, все другие $x_i \geq 0$. Подставляя эти величины в блоки второй системы, получаем $x_1^0 = 6$, $x_3^0 = 8$, $x_8^0 = 4$, $x_6^0 = 10$, $x_7^0 = 6$, $z^0 = -32$. Поскольку все $x_i^0 \geq 0$, это решение оптимально.

Задачи

1. Применить релаксацию к задаче

$$\min z$$

при ограничениях

$$\begin{aligned} x_1 + 3x_3 + x_4 &= 1, \\ x_2 - x_3 + 5x_4 &= 2, \\ -z + 2x_3 + x_4 &= -3, \\ x_i &\geq 0, \quad \forall i, \\ x_1 + x_2 &\leq 2, \end{aligned}$$

и обосновать соответствие между проводимыми операциями и операциями двойственного симплекс-метода.

2. Для процедуры закрепления (§ 4.2) доказать, что (а) выполняется критерий оптимальности, (б) последовательность целевых функций является не-

убывающей, (в) процедура требует решения только конечного числа задач P^k .

3. Показать, что при выборе g_i и R , указанном в конце § 4.2, процедура закрепления может быть сделана совпадающей с симплекс-методом.

4. Показать, как может быть построено двойственное решение в теореме 4.3—3 с помощью оптимального двойственного решения сокращенной задачи и матриц B_{ii}^{-1} .

5. Показать, что если решение сокращенной задачи в каждом цикле единственно, то алгоритм Риттера сходится к оптимальному решению за конечное число циклов.

6. Обсудить возможность преобразования алгоритма § 4.3 в алгоритм, генерирующий последовательность допустимых решений, путем применения его к задаче, двойственной к решаемой. Показать, как могут быть немедленно получены базисные матрицы в двойственных блоках, если такие матрицы известны для исходных блоков. Каковы размерности этих базисов и каков размер сокращенной задачи? Выявить относительные преимущества и недостатки применения алгоритма к прямой и двойственной задачам.

7. Рассмотреть обобщения алгоритма Риттера на задачи с ограничениями вида (4.3-2)—(4.3-4), но с целевой функцией, являющейся нелинейной дифференцируемой и вогнутой.

8. Решить задачу, двойственную к примеру из § 2.5, используя метод Розена.

9. Распространить результаты теоремы 4.4-5 на алгоритм Риттера.

10. Рассмотреть линейную задачу с двухсторонними ограничениями на переменные. Преобразовать ее в одноблочную добавлением вспомогательных переменных в неравенства, задающие границы. Приложить алгоритм Розена. Выяснить этапы процедуры решения, указывая любые упрощения, которые можно внести. Каков простейший выбор начального множества переменных, для которых релаксируются ограничения неотрицательности?

11. В качестве обобщения задач с двухсторонними ограничениями рассмотреть задачу

$$\min c_1x_1 + c_2x_2 + dy$$

при ограничениях

$$A_1x_1 + A_2x_2 + ay = b,$$

$$x_2 \leq ey,$$

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

где e — вектор, все компоненты равны единице, а y — скалярная переменная. Повторить задание задачи 10 для этого случая.

12. Показать, что если правила выбора множеств V соотносятся так, как указано в конце § 4.4, то применение релаксации к прямой задаче (4.4-3) — (4.4-6) приводит к тем же самым результатам, что и применение закрепления к двойственной.

ЛИТЕРАТУРА

1. A. M. Geoffrion, Relaxation and the Dual Method in Mathematical Programming. Working Paper 135, Western Management Science Institute, University of California at Los Angeles, 1968.
2. R. Graves and P. Wolfe, eds., Recent Advances in Mathematical Programming, McGraw-Hill, Inc., New York, 1963, pp. 177—200.
3. K. Ritter, A Decomposition Method for Linear Programming Problems with Coupling Constraints and Variables. Mathematics Research Center, University of Wisconsin, Rept. 739, 1967.

4. J. B. Rosen, Primal Partition Programming for Block Diagonal Matrices. *Numerische Mathematik*, **6**, 1964, pp. 250—260.
5. G. Hadley, *Linear Programming*, Addison-Wesley Publishing Company, Inc., Reading, Mass., 1962, pp. 80—83.
6. M. Simonnard, *Linear Programming*, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1966, p. 397.
7. S. I. Gass, The Dualplex Method for Large—Scale Linear Programs. Ph. D. Thesis, University of California at Berkeley, 1966, ORC Rept. 66—15.
8. M. D. Grigoriadis and W. F. Walker, A Treatment of Transportation Problems by Primal Partition Programming. *Management Sci.*, **14**, No. 9, 1968, pp. 565—599.
9. M. D. Grigoriadis and K. Ritter, A Decomposition Method for Structured Linear and Nonlinear Programs. *Computer Sciences Tech.*, Rept. 10, Computer Sciences Department, University of Wisconsin, 1968.
10. A. M. Geoffrion, Constrained Maximum Likelihood Estimation of Several Stochastically Ordered Distributions, The RAND Corporation, forthcoming.
11. A. M. Geoffrion, Reducing Concave Programs with Some Linear Constraints, *SIAM J. Appl. Math.*, **15**, 1967, pp. 653—664.

ГЛАВА V

МЕТОДЫ КОМПАКТНОГО ОБРАЩЕНИЯ МАТРИЦ

§ 5.1. ВВЕДЕНИЕ

Большинство существующих методов решения больших задач математического программирования включает некоторые приемы децентрализации.

Это справедливо для всех методов, излагаемых в других главах этой книги, где изучаются методы типа декомпозиции или типа расчленения. Все они требуют итеративного решения большого числа подзадач меньшего размера, которые могут быть сами по себе задачами оптимизации или могут просто требовать осуществления, например, ведущей операции. Это предполагает обмен информацией между подзадачами и некоторой координирующей задачей.

Эффективные процедуры решения больших задач могут быть и другого типа. Существует ряд эффективных «централизованных» методов решения больших задач специальной структуры. Все они являются специализированными вариантами модифицированного симплекс-метода, позволяющими сократить время вычислений и ослабить требования к объему памяти. Это удастся достичь путем использования специальных свойств обратных матриц для данного класса решаемых задач, изменяя способы хранения обратной матрицы или оперируя с компактной записью связанных с ней уравнений. Сам по себе модифицированный симплекс-метод с мультипликативным представлением обратной матрицы является компактным методом, который имеет особые преимущества при малой плотности ненулевых элементов в матрицах.

Эта глава начинается с обсуждения мультипликативной формы записи. Далее следуют методы решения задач с двусторонними ограничениями [1], а также их обобщения, в том числе процедура Данцига и Ван Слайка [2] и ее обобщения на блочные задачи [3].

§ 5.2. МОДИФИЦИРОВАННЫЙ СИМПЛЕКС-МЕТОД С МУЛЬТИПЛИКАТИВНЫМ ПРЕДСТАВЛЕНИЕМ ОБРАТНОЙ МАТРИЦЫ

До рассмотрения мультипликативной формы кратко опишем этапы модифицированного симплекс-метода с хранением обратной матрицы в явной форме. Решаемая задача такова:

$$\min z = cx \text{ при } Ax = b, \quad x \geq 0, \quad (1)$$

где

$$A = [P_1, P_2, \dots, P_n] \quad (2)$$

является $m \times n$ -матрицей ранга m , b — m -вектор постоянных и c — n -вектор коэффициентов целевой функции. Уравнение

$$cx - z = 0 \quad (3)$$

добавляется к ограничениям, при этом — z рассматривается как дополнительная $(m+1)$ -я базисная переменная. Пусть B — допустимый базис расширенной системы. В начале цикла k предположим, что известны B^{-1} , базисное решение $x_B = B^{-1}b$ и исходные данные (A, b, c) . Цикл k состоит в следующем:

1. Строка $m+1$ обратной матрицы B^{-1} имеет вид

$$B_{m+1}^{-1} = (-\pi_1, -\pi_2, \dots, -\pi_m, 1), \quad (4)$$

где π_i — двойственные переменные. Образует для каждой небазисной переменной характеристическую разность $\bar{c}_j$, используя уравнение

$$\bar{c}_j = c_j - \sum_i \pi_i a_{ij} = c_j - \pi P_j. \quad (5)$$

2. Предполагая, что используется стандартное правило выбора столбца, находим

$$\min_j \bar{c}_j = \bar{c}_s. \quad (6)$$

3. Если $\bar{c}_s \geq 0$ — процедура останавливается. Текущее базисное решение является оптимальным.

4. Если $\bar{c}_s < 0$, вычисляем преобразованный столбец

$$\begin{pmatrix} \bar{P}_s \\ \bar{c}_s \end{pmatrix} = B^{-1} \begin{pmatrix} P_s \\ c_s \end{pmatrix}. \quad (7)$$

5. Пусть

$$\bar{P}_s = (\bar{a}_{1s}, \bar{a}_{2s}, \dots, \bar{a}_{ms})'. \quad (8)$$

Если все $\bar{a}_{is} \leq 0$ — процедура останавливается: оптимум неограничен. Пусть

$$\hat{x} = (x_B : 0) + x_s (-\bar{P}_s : u_s), \quad (9)$$

где u_s есть s -й единичный вектор. Тогда $\hat{x}$ — решение, допустимое при всех $x_s \geq 0$, причем целевая функция

$$\hat{z} = c_B x_B + \bar{c}_s x_s \quad (10)$$

стремится к $-\infty$ при $x_s \rightarrow +\infty$.

6. В противном случае вычисляем

$$\frac{\bar{b}_r}{\bar{a}_{rs}} = \min_{\bar{a}_{rs} > 0} \frac{\bar{b}_i}{\bar{a}_{is}} = \theta. \quad (11)$$

7. Строим увеличенную матрицу

$$\begin{bmatrix} & : & \bar{P}_s \\ B^{-1} & : & \\ & : & \bar{c}_s \end{bmatrix}$$

и трансформируем ее с ведущим элементом $\bar{a}_{rs}$. Первые $m+1$ столбцов результата дают матрицу, обратную новому базису. Преобразуем базисное решение

$$\begin{aligned} (x_B)_i &\leftarrow (x_B)_i - \theta \bar{a}_{is}, \quad i \neq r, \\ (x_B)_r &\leftarrow \theta \end{aligned} \quad (12)$$

и переходим к этапу 1.

Мультипликативная форма обратной матрицы. В дальнейшем будет удобно считать коэффициенты $c_i (m+1)$ -ми элементами P_i ; $\bar{c}_i$ считать $(m+1)$ -ми элементами $\bar{P}_i$, т. е.

$$a_{m+1,i} = c_i, \quad \bar{a}_{m+1,i} = \bar{c}_i. \quad (13)$$

Назовем элементарной матрицей матрицу, отличающуюся от единичной только одной строкой или столбцом. При мультипликативном представлении матрица B^{-1} не дается явно, а представляется в виде произведения накапливаемых элементарных матриц. Для того чтобы показать, как это делается, примем, что B_c^{-1} — матрица, обратная текущей базисной, и предположим, что новая обратная вычисляется с помощью ведущей операции, опирающейся на ведущий элемент $\bar{a}_{rs}$. Тогда над B_c^{-1} следует произвести следующие операции:

1. Заменить строку r на $1/\bar{a}_{rs}$ · (строка r).
2. Для $i=1, \dots, m+1, i \neq r$ заменить строку i на

$$(\text{строка } i) - \frac{\bar{a}_{is}}{\bar{a}_{rs}} \cdot (\text{строка } r).$$

Легко доказать непосредственным перемножением матриц, что эти

операции соответствуют умножению B_c^{-1} слева на элементарную матрицу

$$E = \begin{bmatrix} 1 & & & \eta_1 \\ & 1 & & \eta_2 \\ & & \ddots & \vdots \\ & & & \eta_r \\ & & & \vdots \\ & & & \eta_{m+1} & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}, \quad (14)$$

где

$$\eta_i = \begin{cases} -\bar{a}_{is}/\bar{a}_{rs}, & i \in 1:m+1, i \neq r, \\ 1/\bar{a}_{rs}, & i = r, \end{cases} \quad (15)$$

т. е.

$$B_n^{-1} = EB_c^{-1}, \quad (16)$$

где B_n^{-1} — новая обратная матрица. Если начальный базис был единичным и если осуществлены k ведущих операций, то на цикле k обратная матрица B_k^{-1} дается формулой

$$B_k^{-1} = E_k E_{k-1} \dots E_1, \quad (17)$$

где каждая из матриц E_i является элементарной матрицей типа (14). Полученное представление B_k^{-1} в виде произведения элементарных матриц называется *мультипликативной формой обратной матрицы*.

Важным свойством элементарных матриц является то, что они могут сохраняться в памяти путем записи только элементов неединичного столбца и его положения в матрице (практически необходимо запоминать только ненулевые элементы и их положение). Поскольку обратная матрица теперь в явной форме недоступна, все вычисления, требующие ее использования, сводятся теперь к последовательности умножений на элементарные матрицы. Рассмотрим эти вычисления в том порядке, в котором они производятся в симплекс-методе.

Оценка двойственных переменных. Двойственные переменные даются формулой

$$\pi = c_B B^{-1}, \quad (18)$$

так что если B^{-1} дана в мультипликативной форме, то приходится оценить следующее матричное произведение:

$$\pi = (\dots ((c_B E_k) E_{k-1}) \dots) E_1. \quad (19)$$

Операции производятся в том порядке, как указано скобками,

т. е. сначала вычисляется строка $c_B E_k$. Затем $(c_B E_k) E_{k-1}$ и т. д. Каждая из этих операций требует умножения элементарной матрицы слева на вектор-строку. Пусть

$$E = [u_1, u_2, \dots, u_{r-1}, \eta, u_{r+1}, \dots, u_{m+1}], \quad (20)$$

где u_i есть i -й единичный вектор, и пусть

$$v = (v_1, \dots, v_{m+1}) \quad (21)$$

— произвольный вектор-строка. Тогда

$$vE = (v_1, \dots, v_{r-1}, \delta, v_{r+1}, \dots, v_{m+1}), \quad (22)$$

где

$$\delta = v\eta = \sum_{i=1}^{m+1} v_i \eta_i, \quad (23)$$

т. е. вектор vE есть строка, отличающаяся от v только одним элементом δ , который равен скалярному произведению v на неединичный столбец η . Таким образом, вычисление π при задании обратной матрицы в виде произведения k элементарных матриц требует вычисления k скалярных произведений.

Вычисление ведущего столбца $\bar{P}_s$. Вектор $\bar{P}_s$ дается формулами

$$\bar{P}_s = B^{-1}P_s, \quad (24)$$

$$\bar{P}_s = E_k(\dots(E_2(E_1 P_s))\dots). \quad (25)$$

Здесь произведения вычисляются в прямой последовательности: сначала $E_1 P_s$, затем $E_2(E_1 P_s)$ и т. д. Каждая операция требует вычисления произведения вида Ev . Если E и v таковы, как в (20), (21), но v теперь вектор-столбец, то

$$Ev = (\alpha_1, \dots, \alpha_{m+1}), \quad (26)$$

где

$$\alpha_i = \begin{cases} v_i + \eta_i v_r, & i \in 1, \dots, m+1, i \neq r, \\ \eta_r v_r, & i = r. \end{cases} \quad (27)$$

Таким образом, оценка Ev требует $(m+1)$ -го умножения и m сложений. Поскольку для ЦВМ затраты времени на сложение значительно меньше, чем на умножение, потребное для оценки vE , время определяется, по существу, $m+1$ -умножением.

Изменение обратной матрицы. Здесь требуется только добавление к списку, сохраняемому в памяти, нового вектора η вида (14), (15), элементы которого вычисляются по текущему $\bar{P}_s$.

Относительные преимущества. Модифицированный симплекс-метод, в особенности в мультипликативной форме, обладает значительными преимуществами по сравнению со стандартной формой. Это относится к точности, скорости и требованиям к па-

матри. Большая часть этих преимуществ определяется тем фактом, что, как правило, матрицы больших линейных задач (т. е. с $n > m > 100$) являются слабозаполненными, содержат малый процент ненулевых элементов. Обычной является плотность в 5% или менее. Модифицированная форма симплекс-метода в большей степени способна использовать преимущества, вытекающие из этого факта. В этой форме характеристические разности и ведущий вектор вычисляются непосредственно по исходным данным. Поскольку исходная матрица слабозаполнена, а перемножение следует производить только тогда, когда оба сомножителя отличны от нуля, то время вычислений значительно сокращается. В дополнение к этому использование только исходных данных приводит к тому, что уменьшается возможность накопления ошибок округления. Наоборот, стандартные симплексные таблицы, даже если они первоначально являются слабозаполненными, в ходе итеративного процесса быстро заполняются ненулевыми элементами. Таким образом, время вычислений увеличивается, и, поскольку каждая таблица вычисляется из предшествующей, накопление ошибок может начать играть более серьезную роль.

Проведем сравнение числа умножений и делений за одну итерацию при трех формах симплекс-метода (операции, общие для всех трех, опускаются). Стандартная форма требует арифметических действий при применении ведущей операции ко всей $(m+1) \times n$ -таблице. Расчет операций в предположении, что преобразуются только небазисные столбцы, дает следующее:

1. Требуется разделить строку r на $\bar{a}_{rs}$: $n - m$ делений.
2. Для $i \in 1 : m+1$, $i \neq r$ требуется умножить строку r на $-\bar{a}_{is}$ и прибавить результат к строке i : $m(n - m)$ умножений.

В целом получаем

$$n_s = (m + 1)(n - m) \text{ операций.} \quad (28)$$

Фактор сокращения времени из-за наличия нулевых элементов не действует, если прошло достаточно итераций для того, чтобы таблицу можно было считать заполненной.

Для модифицированного метода с представлением обратной матрицы в явной форме предположим, что матрица исходной задачи имеет плотность ненулевых элементов d , но обратная матрица B^{-1} заполнена целиком. Каждый из $n - m$ небазисных столбцов требует для расчета характеристической разности вычисления одного скалярного произведения πP_i , что можно осуществить за $dm(n - m)$ операций. Вычисление ведущего столбца $\bar{P}_s = B^{-1}P_s$ требует dm^2 операций. Преобразование обратной матрицы требует $(m+1)^2$ операций, так что общее число операций равно

$$n_{re} = dm(n - m) + dm^2 + (m + 1)^2. \quad (29)$$

Если обратная матрица хранится в мультипликативной форме, объем вычислений зависит от числа векторов η и их заполненности. Как будет показано далее, плотность ненулевых элементов в этих векторах значительно меньше, чем это может быть достигнуто при обычной процедуре обращения. Предположим, что средняя плотность ненулевых элементов для векторов η равна d_1 , и что среднее число этих векторов равно m^*). Тогда при вычислении двойственных переменных должно быть найдено m скалярных произведений, т. е. произведено $d_1 m(m+1)$ операций. Для характеристических разностей требуется $dm(n-m)$ операций. Вычисление $B^{-1}P_s$ требует m скалярных произведений, т. е. $d_1 m(m+1)$ операций, в то время как вычисление нового вектора η включает $m+1$ операцию. Таким образом, полное число операций равно

$$n_{rp} = d_1 m(m+1) + dm(n-m) + d_1 m(m+1) + m + 1. \quad (30)$$

Заметим, что если d_1 и d равны единице, то стандартный метод требует меньше работы, чем два других. Однако для задач с низкой плотностью ненулевых элементов модифицированные формы становятся более эффективными. Условие большой эффективности модифицированного метода с явным заданием обратной матрицы, очевидно, имеет вид $n_{re} < n_s$:

$$dm(n-m) + dm^2 + (m+1)^2 < (m+1)(n-m) \quad (31)$$

или, в пренебрежении величинами порядка, меньшего, чем nm или m^2 ,

$$dmn < nm - 2m^2 \quad (32)$$

или

$$d < 1 - 2\left(\frac{m}{n}\right). \quad (33)$$

Для отношения $n/m > 4$ это дает $d < 50\%$, что обычно выполняется при больших m . Для сравнения явной и мультипликативной форм запишем приближенные оценки числа операций в пренебрежении величинами меньшими m^2 :

$$n_{re} \doteq dm^2 + dm(n-m) + m^2 = m^2(1+d) + dm(n-m), \quad (34)$$

$$n_{rp} \doteq d_1 m^2 + dm(n-m) + d_1 m^2 = m^2(2d_1) + dm(n-m). \quad (35)$$

Мультипликативная форма потребует меньшего числа операций, если

$$1 + d > 2d_1 \quad (36)$$

*) Когда число становится больше m , базис обращается заново, что кратко обсуждается ниже.

или

$$d_1 < \frac{1+d}{2}. \quad (37)$$

Для малых d это равносильно условию $d_1 < 1/2$.

В задачах с неплотными матрицами это условие обычно может выдерживаться с помощью периодического обращения базисной матрицы заново. Преимущество модифицированных форм с точки зрения затрат памяти очевидно. Требуется сохранить только B^{-1} , что заведомо меньше полной преобразуемой таблицы. Далее для больших задач B^{-1} должно сохраняться во внешней памяти (на дисках, лентах и т. д.). Перенос информации из внешней памяти в МОЗУ обычно является узким местом, операцией значительно более медленной, чем арифметические. Таким образом, поскольку B^{-1} меньше, чем полная таблица, то это экономит не только память, но и затраты времени.

При мультипликативном представлении, если векторы η поддерживаются достаточно разреженными, требования к памяти еще более сокращаются. Вулф и Катлер [4] сообщают, что при использовании мультипликативной формы записи задача с матрицей, имеющей 245 строк, могла целиком решаться без обращения к внешней памяти на ЦВМ IBM-7090, имеющей объем МОЗУ в 32 768 слов. Сходная задача с использованием явного представления обратной матрицы требует 75 тыс. слов, а при стандартной форме симплекс-метода 118 тыс. слов.

Одновременное вычисление характеристических разностей. Тот факт, что в модифицированном симплекс-методе характеристические разности должны вычисляться отдельно, приводит к большой гибкости. При мультипликативном представлении время вычислений зависит главным образом от того, сколько раз приходится использовать полную систему данных (A, b, c) . Это очевидно в тех случаях, когда данные хранятся во внешней памяти, поскольку каждое использование их требует затрат времени на перенос. Однако даже в том случае, когда все данные удастся разместить в МОЗУ, основной объем работы состоит в вычислении двойственных переменных, характеристических разностей и нахождении ведущего столбца, причем каждая из этих операций требует обращения к исходным данным.

Назовем каждое из этих обращений термином «ход» [4]. Число ходов может быть сокращено с помощью следующей тактики, называемой одновременным расчетом разностей или субоптимизацией. Вычислим двойственные переменные π и найдем небазисные столбцы с k наиболее отрицательными характеристическими разностями, $k > 1$. Преобразуем эти столбцы и построим сокращенную задачу, включающую только текущие базисные переменные и указанные k небазисных. Применим к этой задаче стандартную симплексную методику (все процедуры

осуществляются в МОЗУ). Когда сокращенная задача оказывается решенной, переходим к следующему ходу. Решение сокращенной задачи не требует обращения к исходным данным и вместе с тем может дать значительное изменение в величине целевой функции. Число ходов, а следовательно, полное время на вычисление может быть значительно сокращено [5]. В стандартной программе решения задач ЛП, входящей в систему математического программирования IBM MPS-360 [6], k принимается целым числом от единицы до десяти. Брать слишком большие величины k не целесообразно, поскольку это может замедлить сходимость из-за затрат времени на столбцах, ввод которых относительно невыгоден. Эта процедура описывалась и ранее в § 3.2, где она привела к блестящим результатам.

Повторное обращение. При использовании мультипликативной формы на каждом шаге добавляется новый вектор η . Чем больше таких векторов имеется, тем больше требования к памяти и тем больше затраты времени на вычисление в каждом цикле. Эти проблемы усложняются и тем фактом, что векторы η , генерируемые в последовательных итерациях, как правило, содержат тем больше ненулевых элементов, чем больше итераций. Таким образом, после некоторого числа циклов желательно сделать представление текущей обратной матрицы более компактным с помощью повторного обращения, использующего исходные данные. Это имеет и дополнительные преимущества с точки зрения исключения накопления ошибок. Существуют эффективные процедуры, которые позволяют значительно сократить как число, так и плотность заполнения векторов η . Такие процедуры являются ключевым элементом для обеспечения эффективности мультипликативного представления.

Эффективные алгоритмы повторного обращения основываются на том факте, что хотя матрица, обратная к базисной, является единственной, ее представление в виде произведения элементарных матриц неединственно. Следовательно, может быть найдено представление, которое отличается определенными желательными свойствами, например, малой плотностью. Простая, но эффективная процедура была предложена Ларсеном [7]. Ее модификации содержатся в большинстве существующих стандартных программ, например LP/90 для ЦВМ IBM-7090 [5].

Пусть обращаемая матрица такова:

$$B = [P_1, P_2, \dots, P_{m+1}]. \quad (38)$$

Процедура Ларсена для вычисления B^{-1} состоит в следующем:

1. Начать с $i=1$.
2. Выбрать P_i как вектор, входящий в базис. Умножить P_i на текущее произведение элементарных матриц, генерированных до сих пор, что позволяет получить $\bar{P}_i$.

3. Выбрать ведущую строку в $\bar{P}_i$. Прежде всего, определить допустимые ведущие строки из тех, которые не были раньше выбраны, и в которых элементы $\bar{P}_i$ имеют абсолютную величину, большую, чем некоторое допустимое T (малое положительное число; Ларсен выбирал $T=10^{-3}$). Тогда, исходя из этого допустимого множества, выбрать ведущую строку как ту, в которой матрица $[P_i, \dots, P_{m+1}]$ имеет минимальное число ненулевых элементов.

4. Зная ведущую строку и элементы $\bar{P}_i$, образуем новый вектор η , используя (15), и добавим его к накапливаемому множеству.

5. Заменяем i на $i+1$ и переходим к шагу 2.

В этой процедуре векторы, выбираемые для ввода в базис, занумеруем в порядке, в котором они появляются в памяти: 1) P_1 , затем 2) P_2 и т. д. В конце цикла i текущие векторы η представляют собой обратную матрицу, содержащую $P_1, \dots, P_i$ и единичные векторы. Вектор P_{i+1} вносится в эту матрицу путем умножения ее на текущую обратную (в мультипликативной форме) и преобразовании с помощью допустимого ведущего элемента. P_{i+1} должен заменить некоторый единичный вектор, и ведущий элемент должен быть не слишком близок к нулю, поскольку это может серьезно повлиять на накопление ошибок, отсюда и возникает число T .

С учетом этих ограничений правило выбора ведущего элемента может быть рационализировано следующим образом. Вектор η , вычисляемый из $\bar{P}_i$ с ведущим элементом $\bar{a}_{rs}$, имеет компоненты

$$\eta_i = \frac{-\bar{a}_{is}}{\bar{a}_{rs}}, \quad i = 1, \dots, m+1, \quad i \neq r, \quad (39)$$

$$\eta_r = \frac{1}{\bar{a}_{rs}}. \quad (40)$$

Единственный путь для того, чтобы создать нулевые компоненты η_i , состоит в том, чтобы создать нулевые элементы $\bar{a}_{is}$.

Элементы $\bar{a}_{is}$ в цикле i образуются следующим образом. Пусть B_c^{-1} есть произведение текущих элементарных матриц. Рассмотрим следующую матрицу:

$$B_c^{-1}[P_i, P_{i+1}, \dots, P_{m+1}]. \quad (41)$$

Элемент $\bar{a}_{is}$ образуется путем умножения некоторой строки этой матрицы, ведущей строки, на отличную от нуля константу и добавления результата умножения ее к другим строкам. Если ведущая строка такова, что она имеет минимальное число нулевых элементов, то вероятно, что будет создано мало новых

ненулевых элементов. К сожалению, столбцы $B_c^{-1}P_j$, $j=i, \dots, m+1$ не все доступны (мы располагаем только $B_c^{-1}P_i$). Вместо этого на шаге 3 выбирается строка $[P_i, P_{i+1}, \dots, P_{m+1}]$ с минимальным числом ненулевых элементов, с учетом предположения, что эта строка в матрице (41) должна также иметь мало ненулевых элементов.

Т а б л и ц а 5.2.1

Задача 1

Строк 354, векторов 529, элементов 6170, плотность 3,29%

	Стандартный метод	Модифицированный метод	% уменьшения
Обращение 1 (2333 не нулевых элемента в базисе), мин.	3,79	2,05	46
Число ненулевых элементов в преобразующих векторах	17 950	7496	53
60 симплекс-итераций, мин.	14,80	9,58	35
Обращение 2 (2084 ненулевых элемента в базисе), мин.	3,64	2,06	43
Число ненулевых элементов в преобразующих векторах	17 382	7626	56
Всего, мин.	22,23	11,69	48

Задача 2

Строк 360, векторов 1239, элементов 7168, плотность 1,61%

	Стандартный метод	Модифицированный метод	% уменьшения
Обращение 1 (1060 ненулевых элементов в базисе), мин.	1,71	1,43	16
Число ненулевых элементов в преобразующих векторах	4368	1694	61
50 симплекс-итераций, мин.	4,41	3,50	21
Обращение 2 (1117 ненулевых элементов в базисе), мин.	1,70	1,53	10
Число ненулевых элементов в преобразующих векторах	4868	1958	60
Всего, мин.	7,82	6,46	17

Ларсен сообщает о блестящих результатах этой процедуры в двух задачах, описанных в таблице 5.2.1; метод инвертирования, описанный выше, сравнивается со стандартным методом,

в котором ведущие операции производятся с элементами, имеющими наибольшую абсолютную величину. Сокращение плотности и времени вычисления достигает 40%. Ларсен устанавливает, что эти данные являются достаточно представительными для широкого круга задач, которые рассчитывались в течение нескольких месяцев действия программы.

Новые процедуры повторного обращения *). Улучшенные процедуры обращения были развиты после опубликования процедуры Ларсена в 1962 г. Они используют более сложный критерий для выбора ведущего элемента. Во всех этих процедурах используется упорядочение базисной матрицы, которая должна быть обращена таким образом, чтобы она имела форму, показанную на рис. 5.1, где только заштрихованная часть содержит ненулевые элементы. Подматрицы $UT \times UT$ и $LT \times LT$ являются неособенными нижними треугольными матрицами, в то время как $K \times K$ -подматрица называется *ядром*. Так как верхняя правая подматрица

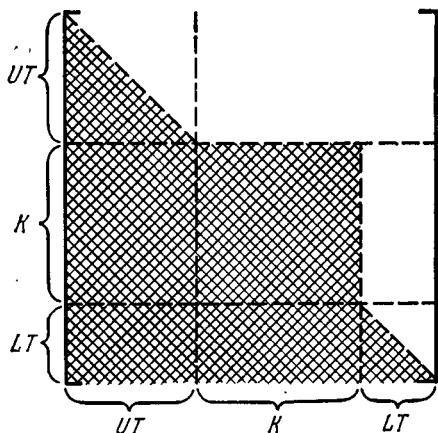


Рис. 5.1. Треугольные матрицы и ядро.

рица равна нулю, осуществление ведущей операции в верхней треугольной не затрагивает никаких других столбцов и строк. Таким образом, векторы η , представляющие эти ведущие операции, образуются и передаются во вспомогательную память, например, на диски, ленты и т. д. Сходным образом осуществление ведущей операции в строках и столбцах подматрицы $LT \times LT$ не затрагивает никаких строк, лежащих выше их. Эти ведущие операции оставляются на последнюю очередь по двум причинам. Они приводят к увеличению плотности в подматрице $LT \times K$, вызывая появление η векторов, образуемых из столбцов, пересекающих их и имеющих больше ненулевых элементов. Ядро также будет приведено к треугольной форме так, что нижняя треугольная матрица будет в конечном счете расширена, чтобы включать ядро.

Описанная процедура осуществления ведущих операций в ядре использована в процедуре линейного программирования

*) Материал этого параграфа был сообщен автору Дэвидом Sommerom.

«Allegro» для машин типа IBM-3600 и 3800. В отличие от метода Ларсена, после того как образована каждая элементарная матрица, преобразуются, т. е. умножаются на текущую элементарную матрицу, все столбцы ядра, которые еще не были преобразованы. Выбор ведущего элемента основывается на результате, принадлежащем Марковицу. Пусть r_i число ненулевых элементов в строке i , k_j — число ненулевых элементов в столбце j . Если ведущая операция использует элемент a_{ij} в качестве ведущего, то результат умножения строки i добавляется к $k_j - 1$ другим строкам и создает не более $r_i - 1$ новых ненулевых элементов в каждом *).

Таким образом, верхняя грань числа новых ненулевых элементов в текущей матрице равна

$$u_{ij} = (r_i - 1)(k_j - 1). \quad (42)$$

Естественно, что можно выбрать в качестве ведущего ненулевой элемент, с которым связано наименьшее u_{ij} .

На практике поиск минимального u_{ij} может быть сделан только приближенно.

В дополнение развития этих идей укажем, что нет необходимости в действительности переупорядочивать обрабатываемую матрицу к форме, показанной на рис. 5.1. Определим все строки и столбцы первоначально как допустимые. Пусть строка оценивается числом ненулевых элементов в строке, входящей в допустимые столбцы. Исходным образом производится оценка столбцов. Для того чтобы опознать, выделить строку, принадлежащую к передней треугольной матрице, найдем ту строку, которая имеет оценку, равную единице, и будем считать в дальнейшем эту строку недопустимой. Пусть ненулевой элемент расположен в столбце r_i ; тогда сделаем этот столбец недопустимым. Изменим оценки строк и столбцов для того, чтобы отразить удаление этой строки и столбца из допустимого множества. Тогда любая другая строка с оценкой, равной единице, также попадет в переднюю треугольную. Повторяя эту процедуру вплоть до того, что все допустимые строки будут иметь оценку 2 или более, мы найдем все строки, принадлежащие передней треугольной подматрице.

Та же самая процедура заменой роли столбцов и строк позволяет выделить столбцы, принадлежащие задней треугольной подматрице.

*) Может быть создано и меньшее число ненулевых элементов, в силу того, что ненулевые элементы уже могут существовать в этих других $k_j - 1$ строках.

Для примера в матрице

$$\begin{array}{c}
 \begin{array}{cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \left[\begin{array}{cccccc}
 & & x & x & & \\
 x & & x & & x & \\
 & x & & & & \\
 x & & x & & & \\
 x & x & & x & & x \\
 x & x & x & & x &
 \end{array} \right] & \begin{array}{c} i \\ i \\ i \\ i \\ i \\ i \end{array}
 \end{array}
 \end{array}$$

где x представляет собой ненулевой элемент, строка 3 имеет один ненулевой элемент. Осуществление ведущей операции с элементом a_{32} не задевает других столбцов, за исключением столбца 2, так что вектор η образуется из столбца 2 и затем переносится во внешний накопитель. Строка 3 и столбец 2 помечены индексом i как недопустимые, и столбец 2 становится единичным вектором. Все другие строки имеют оценки равные 2 или более, так что завершается операция выделения передней треугольной подматрицы. Обращаясь к формированию задней треугольной подматрицы, мы видим, что столбец 6 имеет оценку 1 с ненулевым элементом, появляющимся в строке 5. Помечая строку 5 и столбец 6, как недопустимые, индексом i , переходим к столбцу 4, который теперь имеет оценку 1 и имеет ненулевой элемент в строке 1. Делая столбец 4 и строку 1 недопустимыми, видим, что все другие столбцы имеют оценку 2 или более, так что мы осуществили все упорядочение и можем перейти к триангуляризации ядра.

Процедура выбора ведущего элемента в ядре такова:

1. Найти допустимый столбец с наименьшей оценкой строки. Из остающихся строк вновь найти имеющую наименьшую оценку и повторить процедуру k раз (типичное значение $k=3$).

2. Для каждой из k строк, найденных в шаге 1, найдем допустимый столбец с минимальной оценкой столбца и пересекающийся с этой строкой, т. е. имеющий в ней ненулевой элемент.

3. Повторим шаги 1 и 2 с заменой роли строк и столбцов.

4. Из найденных $2k$ пар индексов столбцов и строк выберем пару с наименьшей величиной u_{ij} . Это определит ведущий элемент.

Выбирая $k=2$ для нашего маленького примера, приходим к некоторому числу элементов, имеющих наименьшее u_{ij} . Выбирая по произволу элемент a_{25} , помечаем строку 2 и столбец 5 как недопустимые. Осуществление ведущей операции на элементе 2—5 требует только нормирования a_{25} до величины 1 и преобразования элементов столбца 5, в которых имеются допустимые строки (здесь элемент 6—5), к нулю. Мы образуем век-

тор η , представляющий это преобразование, используя столбцы 1 и 3, и запоминаем их.

После проведения ведущих операций на элементах a_{32} и a_{25} , получаем следующий базис:

$$\begin{array}{c}
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \left[\begin{array}{cccccc}
 1 & 2 & 3 & 4 & 5 & 6 \\
 & & x & x & & \\
 x & & x & & 1 & \\
 & 1 & & & & \\
 x & & x & & & \\
 x & & & x & & x \\
 x & & x & & &
 \end{array} \right] \begin{array}{c} i \\ i \\ f \\ f \\ i \\ i \end{array}
 \end{array}$$

Теперь все четыре допустимых ведущих элемента, оставшиеся в таблице, имеют $u_{ij}=1$, так что, выбирая по произволу элемент a_{41} , пометим строку 4 и столбец 1 как недопустимые. Мы образуем вектор η для того, чтобы ведущий элемент преобразовался в единицу, а элемент (6,1) преобразовался в нуль; используем его для преобразования столбца 3 и запоминаем его. Элемент 6—3 является единственным остающимся возможным ведущим элементом, так как единственный ненулевой элемент в столбце 3, который находится в ведущей строке, является ведущим элементом, процедура сводится только к делению строки 6 на этот элемент. Образуя и запоминая вектор η , приходим к завершению операции с ядром. Окончательная базисная матрица имеет такой вид:

$$\begin{array}{c}
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \left[\begin{array}{cccccc}
 1 & 2 & 3 & 4 & 5 & 6 \\
 & & x & x & & \\
 x & & x & & 1 & \\
 & 1 & & & & \\
 1 & & x & & & \\
 x & & & x & & x \\
 & & 1 & & &
 \end{array} \right] \begin{array}{c} i \\ i \\ f \\ i \\ i \\ i \end{array}
 \end{array}$$

Подматрица, состоящая из столбцов и строк, помеченных индексом i , является нижней треугольной. Конечная стадия процедуры обращения состоит в применении передних треугольных операций, описанных выше, к этой подматрице. Генерация и запоминание соответствующих векторов η завершает процедуру.

В существующей программе «Allegro» элементы базисной матрицы накапливаются в виде двойного списка с указателем, ведущим от каждого ненулевого элемента a_{ij} к адресу в памяти следующего ненулевого элемента в строке i и столбце j . Элемент определяется как ненулевой, если его абсолютная величина больше, чем некоторая положительная величина ε с типичным значением 10^{-11} .

Более строгий критерий отличия от нуля должен быть принят при выборе ведущего элемента. Предположим, что ведущий элемент выбирается из столбца s с компонентами $\bar{a}_{is}$. Вычислим

$$\rho = \max_i |\bar{a}_{is}|. \quad (43)$$

Здесь индекс i берется из множества всех текущих допустимых строк. Тогда элемент $\bar{a}_{is}$ считается достаточно большим для того, чтобы быть ведущим, только если выполняется условие

$$|\bar{a}_{is}| \geq \delta \quad (44)$$

и

$$|\bar{a}_{is}| \geq \gamma \rho, \quad (45)$$

где типичные величины для δ и γ суть 10^{-6} и 10^{-2} соответственно. Соотношение (45) предохраняет против возникновения очень больших элементов в векторах η . Если ведущий элемент, выбираемый в передней треугольной подматрице, не проходит по этому критерию, обрабатываемая матрица рассматривается как особая. Столбец замещается искусственным единичным вектором и вновь начинается фаза 1. Если критерий не проходит всюду, то выбирается следующий наилучший ведущий элемент в столбце в смысле малости оценки строки. Если все потенциальные ведущие элементы окажутся меньше чем δ , вновь начинается фаза 1.

При преобразовании ядра, как только столбец преобразован, его ненулевые элементы, принадлежащие к недопустимым строкам, могут быть удалены во внешнюю память вплоть до осуществления нижних треугольных операций. Несмотря на это, список матричных элементов в допустимых строках и столбцах может оказаться первоначально слишком большим для того, чтобы быть хранимым в МОЗУ или может стать позднее слишком большим в силу создания новых ненулевых элементов. На любой стадии операции на ядре, если список матричных элементов превышает возможности оперативной памяти, столбцы, выбираемые для того, чтобы храниться во вне, это столбцы, которые имеют наибольшие оценки. Эти столбцы не вызываются назад вплоть до того момента, когда появляется пространство для них в магнитной памяти. Таким образом, по крайней мере на нескольких операциях они не играют никакой роли в выборе ведущих элементов. Это может привести к плохому выбору ведущего столбца. Таким образом, процедура часто приводит к плохим результатам, когда большая часть столбцов должна остаться во внешней памяти.

Следует отметить, что число новых ненулевых элементов, образуемых на данной ведущей операции, может быть вычислено

без особых трудностей, если доступна текущая матрица. Процедура, которая позволяет выбрать ведущий элемент, для которого это число является наименьшим, и включается в программу «Allegro». Однако результаты использования ее, вообще говоря, незначительно лучше, чем те, при которых используются верхние оценки u_{ij} .

§ 5.3 МЕТОДЫ, УЧИТЫВАЮЩИЕ ДВУСТОРОННИЕ ОГРАНИЧЕНИЯ

Задачи ЛП, в которых на некоторые переменные наложены двусторонние ограничения или, что то же самое, указаны верхние границы их допустимого изменения, часто встречаются на практике. Мы предположим здесь для простоты, что все переменные имеют такие верхние ограничения, так что задача приобретает вид такой:

$$\min z = cx, \quad (1)$$

$$Ax = b, \quad (2)$$

$$x \geq 0, \quad (3)$$

$$x \leq u, \quad (4)$$

где A есть $m \times n$ -матрица ранга m , а u — положительный n -вектор. Последующие результаты легко модифицируются на случай, когда только некоторые из переменных действительно ограничены.

Наиболее прямолинейный способ работы с такими задачами состоит в том, чтобы преобразовать ограничение (4) в равенство с помощью добавочных переменных s :

$$x + s = u, \quad s \geq 0. \quad (5)$$

Эти равенства могут рассматриваться совместно с равенствами (2), и задача в целом решается обычным симплекс-методом. Однако этот прием увеличивает как размер базисной матрицы, так и число переменных на n . Так как $n > m$, и часто $n \gg m$, то это представляет собой серьезную неприятность с вычислительной точки зрения.

Наиболее привлекательной альтернативой является использование специальной методики, развитой для таких задач. Эти специальные приемы позволяют решать задачи с учетом ограничений (4), сохраняя размер базисной матрицы равным $m \times m$. Вычисления производятся так же, как и в симплекс-методе, применяемом для задач без ограничений (4), но с изменением критерия оптимальности и критерия ввода в базис и вывода из базиса векторов.

Дополнение. Значения независимых или небазисных переменных при этом не всегда являются нулями; некоторые из них

равны нулю, а некоторые равны верхним допустимым значениям u_j . Как мы увидим далее, операции симплексной процедуры не слишком серьезно изменяются, если допустить положительные значения небазисных переменных.

Теоретические основы метода. Алгоритм основывается на двух главных теоремах. Для удобства определим множество ограничений (1)–(4) как множество

$$S = \{x/Ax = b, 0 \leq x \leq u\}. \quad (6)$$

Теорема 1. Вектор $x \in S$ является крайней точкой S , если и только если $n - m$ компонент x либо равны нулю, либо равны u_j , в то время как остающиеся m лежат между нулем и u_j и ассоциированы с базисной матрицей ограничения (2).

Доказательство. Переписывая (2), (3) и (5) в компонентной форме, получаем

$$\left. \begin{array}{rcl} P_1 x_1 + P_2 x_2 + \dots + P_n x_n & = & b, \\ x_1 & + s_1 & = u_1, \\ & + s_2 & = u_2, \\ & & \vdots \\ & & + s_n = u_n, \end{array} \right\} \quad (7)$$

$$x_i \geq 0, \quad s_i \geq 0.$$

Если (x, s) является крайней точкой (7), тогда x является крайней точкой S . Вектор (x, s) является крайней точкой (7), если и только если он является базисным допустимым решением, т. е. является неотрицательным, имеет $n - m$ переменных, равных нулю, и остающиеся переменные ассоциированы с неособенной подматрицей. Так как каждое u_i положительно, то каждое из последних n ограничений должно содержать по крайней мере одну положительную переменную. Таким образом, только m из этих ограничений может содержать две базисные переменные; остающиеся $n - m$ должны содержать только одну. Значение x_j в каждом из них равно нулю или u_j .

Рассмотрим теперь те из последних n ограничений, которые содержат две базисные переменные. Для простоты, пусть это будут ограничения с номерами $m+1, \dots, 2m$. Тогда базис включает векторы

$$\left\{ \begin{bmatrix} P_1 \\ \dots \\ e_1 \end{bmatrix}, \dots, \begin{bmatrix} P_m \\ \dots \\ e_m \end{bmatrix} \right\} \quad \text{и} \quad \left\{ \begin{bmatrix} 0 \\ \dots \\ e_1 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ \dots \\ e_m \end{bmatrix} \right\}, \quad (8)$$

где e_i есть i -й единичный вектор. Для того чтобы эти векторы были независимы, векторы $\{P_1, \dots, P_m\}$ должны быть независимы, что доказывается непосредственно.

Перейдем к доказательству необходимости. Пусть переменные, связанные с базисной матрицей для (2), будут $x_1, \dots, x_m$. Рассмотрим квадратную $(n+m)$ -подматрицу (7):

$$\begin{array}{c|cc} s_1 \dots s_m & x_{m+1} \dots x_n & x_1 \dots x_m \\ \hline 0 \dots 0 & P_{m+1} \dots P_n & P_1 \dots P_m \\ \hline e_1 \dots e_m & e_{m+1} \dots e_n & e_1 \dots e_m \end{array} \quad (9)$$

Вычитая столбцы, соответствующие переменным s_i , из столбцов, соответствующих переменным x_i для $i=1, \dots, m$, преобразуем эту матрицу к виду

$$\begin{bmatrix} 0 \dots 0 & P_{m+1} \dots P_n & P_1 \dots P_m \\ \vdots & \vdots & \vdots \\ e_1 & \dots & e_n & 0 \end{bmatrix}. \quad (10)$$

Матрицы (9) и (10) имеют один и тот же ранг. Так как $P_1, \dots, P_m$ независимы, то матрица (9) неособая. Следовательно, переменные $(x_1, \dots, x_n, s_1, \dots, s_m)$ образуют множество базисных переменных для (7), и $(x_1, \dots, x_n)$ является крайней точкой S .

Предположим теперь, что мы нашли крайнюю точку S . Следующая теорема показывает, как это решение может быть проверено на оптимальность.

Теорема 2. Пусть базисные переменные, определенные в силу теоремы 1, обозначены $x_1, \dots, x_m$, и пусть эти переменные исключены из выражения для целевой функции (z -уравнение) так, что

$$z = z_0 + \sum_{j=m+1}^n \bar{c}_j x_j, \quad (11)$$

где $\bar{c}_j$ — характеристические разности. Тогда это базисное решение оптимально для задачи (1)–(4), если и только если

$$\left. \begin{array}{l} x_j = 0 \Rightarrow \bar{c}_j \geq 0, \\ x_j = u_j \Rightarrow \bar{c}_j \leq 0, \end{array} \right\} \quad j = m+1, \dots, n. \quad (12)$$

Доказательство. Приведенное выражение для z не зависит от $x_1, \dots, x_m$. Для $j=m+1, \dots, n$, если $x_j=0$, увеличение x_j может только увеличить z , в то время как если $x_j=u_j$, уменьшение x_j может только увеличить z . Таким образом, текущая точка является локальным, а следовательно, и глобальным минимумом.

Вычислительная процедура. Предположим, что система (1) — (3) записана в канонической форме, т. е. разрешена относительно базисных переменных $x_1, \dots, x_m$:

$$\left. \begin{aligned} x_1 + \bar{a}_{1m+1} x_{m+1} + \dots + \bar{a}_{1n} x_n &= \bar{b}_1, \\ \dots & \\ x_m + \bar{a}_{mm+1} x_{m+1} + \dots + \bar{a}_{mn} x_n &= \bar{b}_m, \\ -z + \bar{c}_{m+1} x_{m+1} + \dots + \bar{c}_n x_n &= -z_0. \end{aligned} \right\} \quad (13)$$

В приведенном выражении переменные $x_{m+1}, \dots, x_n$ равны либо нулю либо u_i . Величины $x_1, \dots, x_m$ находятся путем подстановки величин $x_{m+1}, \dots, x_n$ в (13). Таким образом, величины $\bar{b}_i$ не равны текущим значениям базисных переменных и могут быть и отрицательными и большими, чем u_i .

Шаг 1. Выбор переменной для ввода в базис. Если критерий оптимальности не удовлетворяется, тогда существует небазисная переменная x_k такая, что либо

$$(a) \quad x_k = 0, \quad \bar{c}_k < 0, \quad (14)$$

либо

$$(b) \quad x_k = u_k, \quad \bar{c}_k > 0. \quad (15)$$

Если $\bar{c}_k$ заменить на $-\bar{c}_k$ в случае (b), то для ввода в базис выбирается переменная x_s , где номер s определяется отношением

$$\min_j \bar{c}_j = \bar{c}_s. \quad (16)$$

Если такая переменная может войти в базис с положительным значением, то z должно уменьшиться.

Шаг 2. Выбор переменной, выводимой из базиса. Когда переменная x_s увеличивается от нулевого значения (уменьшается от значения u_s), то значения базисных переменных также должны измениться. Существуют две возможности.

Случай 1. Переменная x_s достигает своей верхней (нижней) границы, но ни одна из базисных переменных не выходит за пределы своей верхней или нижней границы.

Случай 2. Некоторая базисная переменная достигает одной из своих границ раньше, чем x_s .

В случае 1 полагаем x_s равным его верхней (нижней) границе, записываем новые значения базисных переменных и переходим к шагу 1 без изменения базиса. После возможно нескольких повторений этого случая мы должны либо достичь выполнения условий оптимальности, либо перейти к случаю 2.

В случае 2 пусть x_r есть одна из базисных переменных, которая достигла своей верхней или нижней границы первой. Про-

изведем изменение базиса, при котором x_s введем в базис, а x_r выведем из базиса, и перейдем к шагу 1.

Предположим, что x_s уменьшается от своего верхнего значения u_s . Из (13) следует, что базисные и небазисные переменные связаны условиями

$$x_i = \bar{b}_i - \sum_{j=m+1}^n \bar{a}_{ij} x_j, \quad i = 1, \dots, m. \quad (17)$$

Пусть

$$R = \{j/x_j = u_j, j \in \{m+1, \dots, n\}\}. \quad (18)$$

Тогда текущие значения базисных переменных окажутся равными

$$x_i = \bar{b}_i - \sum_{j \in R} \bar{a}_{ij} u_j = x_B^i. \quad (19)$$

Рассмотрим теперь уменьшение некоторой переменной x_s , $s \in R$ от своей верхней границы. В силу (17) результирующие значения x_i таковы:

$$x_i = \bar{b}_i - \sum_{\substack{j \in R \\ j \neq s}} \bar{a}_{ij} u_j - \bar{a}_{is} x_s \quad (20)$$

или, добавляя и вычитая $\bar{a}_{is} u_s$,

$$x_i = x_B^i + \bar{a}_{is} (u_s - x_s). \quad (21)$$

Если $\bar{a}_{is} > 0$, то с уменьшением x_s переменная x_i возрастает, так что x_i может превзойти свой верхний предел. Поэтому возникают ограничения

$$x_B^i + \bar{a}_{is} (u_s - x_s) \leq u_i \quad (22)$$

или

$$x_s \geq u_s + \frac{x_B^i - u_i}{\bar{a}_{is}}, \quad \bar{a}_{is} > 0, \quad (23)$$

так что наименьшее допустимое значение x_s таково:

$$\theta_1 = \max_{\bar{a}_{is} > 0} \left\{ u_s + \frac{x_B^i - u_i}{\bar{a}_{is}} \right\} \quad (\text{достигается при } i = r_1). \quad (24)$$

Если $\bar{a}_{is} < 0$, то с уменьшением x_s уменьшается x_i , так что возникает опасность того, что x_i перейдет нижнюю границу. Поэтому должны быть выполнены ограничения

$$x_B^i + \bar{a}_{is} (u_s - x_s) \geq 0, \quad \bar{a}_{is} < 0, \quad (25)$$

или

$$x_s \geq u_s + \frac{x_B^i}{\bar{a}_{is}}. \quad (26)$$

Наименьшее допустимое значение x_s таково:

$$\theta_2 = \max_{\bar{a}_{is} < 0} \left\{ u_s + \frac{x_B^i}{\bar{a}_{is}} \right\} \text{ и достигается при } i = \bar{r}_2. \quad (27)$$

Так как x_s должно удовлетворять и условию (24) и условию (27), а кроме того, быть неотрицательным, то его новое значение должно быть равно

$$x_s = \theta = \max \{0, \theta_1, \theta_2\}. \quad (28)$$

а) Если $\theta=0$, то изменение базиса не происходит. Новые значения базисных переменных равны в силу (21)

$$x_i = x_B^i + \bar{a}_{is} u_s, \quad i = 1, \dots, m. \quad (29)$$

б) Если $\theta=\theta_1$, переменная x_{r_1} покидает базис.

с) Если $\theta=\theta_2$, переменная x_{r_2} покидает базис.

В случаях б) и с) новые значения базисных переменных вычисляются в силу уравнения (21). В случае совпадений, т. е. в случаях, когда, например, $\theta_1=0$ или $\theta_2=0$, или $\theta_1=\theta_2$, выбор производится произвольно, но всегда в пользу случая а).

Легко показать, что аналогичные соотношения имеют место и в случае, когда x_s имеет текущее значение 0 и увеличивается от этого значения вплоть до значения

$$x_s = \theta = \min \{u_s, \theta_1, \theta_2\}, \quad (30)$$

где

$$\theta_1 = \min_{\bar{a}_{is} > 0} \frac{x_B^i}{\bar{a}_{is}} \quad (\text{достигается при } i = r_1), \quad (31)$$

$$\theta_2 = \min_{\bar{a}_{is} < 0} \frac{x_B^i - u_i}{\bar{a}_{is}} \quad (\text{достигается при } i = r_2). \quad (32)$$

а) Если $\theta=u_s$, то изменений в базисе не производится.

б) Если $\theta=\theta_1$, то переменная x_{r_1} покидает базис.

с) Если $\theta=\theta_2$, то переменная x_{r_2} покидает базис.

Если требуется произвести изменения базиса, то приходится вычислить вектор $\bar{P}_s = B^{-1}P_s$, и произвести преобразование текущей обратной матрицы с помощью ведущего элемента, находящегося в строках r_1 или r_2 вектора $\bar{P}_s$. Предполагая, что значения базисных переменных также вычислены, переходим к новому циклу.

§ 5.4. ЗАДАЧИ С ОБОБЩЕННЫМИ ДВУСТОРОННИМИ ОГРАНИЧЕНИЯМИ

5.4.1. Изложение алгоритма

В этом параграфе будем рассматривать задачи ЛП вида

$$\begin{aligned} & \min x_0 \\ & \begin{array}{l} \text{матрица} \left\{ \begin{array}{c} S_0 \\ S_1 \\ S_p \end{array} \right. \left\{ \begin{array}{l} A^0 x_0 + \dots + A^{n_0} x_{n_0} + A^{n_0+1} x_{n_0+1} + \dots + A^{n_1} x_{n_1} + \dots + A^{n_{p-1}+1} x_{n_{p-1}+1} + \dots + A^{n_p} x_{n_p} = b \end{array} \right. \\ \text{строки} \left\{ \begin{array}{l} x_{n_0+1} + \dots + x_{n_1} \dots \dots \dots x_{n_{p-1}+1} + \dots + x_{n_p} = 1 \end{array} \right. \end{array} \quad (1) \\ & x_i \geq 0, \quad \forall i. \quad (2) \end{aligned}$$

где столбцы A^i и вектор b имеют m компонент. Такую структуру имеют транспортная и обобщенная транспортная задача, а также координирующие задачи в методе Данцига — Вулфа (см. главу II). Любая задача ЛП, в которой каждая переменная имеет не более одного ненулевого коэффициента в последних p уравнениях, причем этот коэффициент и правая часть уравнения оба являются положительными, могут быть приведены к указанной форме путем деления уравнения на величину, стоящую в правой части, и масштабирования переменных. Модификации для случая отрицательных коэффициентов будут рассмотрены позже.

Излагаемая здесь процедура описана Данцигом и Ван-Слайком в [2]. Она является специальным вариантом симплекс-метода, который позволяет решать эти задачи, сохраняя размерность «рабочего базиса» равной $m \times m$. Это существенно, когда p достаточно велико по сравнению с m . Все характерные величины, необходимые для осуществления одной итерации симплекс-метода, т. е. значения базисных переменных, двойственных переменных и представление вводимого столбца через базисные, находятся через соответствующие величины, ассоциированные с рабочим базисом. Этот обобщенный алгоритм связан с изложенным ранее в § 3.2, где аналогичная идея была использована для уменьшения размера базисной матрицы в задаче составления программы многопродуктового производства.

Пусть множество столбцов, содержащих единицу в качестве своего $(m+i)$ -го коэффициента, обозначено S_i , $i=1, \dots, p$ (см. (1)–(2)). Множество столбцов, в которых элементы от $(m+1)$ -го до $(m+p)$ -го равны нулю, обозначим S_0 . Эти множества будут также использованы для обозначения переменных, связанных с указанными столбцами, и для указания индексов этих переменных. Векторы, содержащие все $m+p$ компонент,

будут в дальнейшем обозначаться жирными буквами, так что j -й столбец матрицы (1) — (2) будет обозначаться A^j , в то время как соответствующий вектор, в котором изъяты последние p компонент, будет обозначаться A^j .

Для обоснования процедуры необходимы две теоремы.

Теорема 1. *Любой допустимый базис задачи (1) — (2) должен включать по крайней мере один столбец из каждого множества S_i , $i=0, \dots, p$.*

Доказательство. Переменная $x_0 \in S_0$ всегда является базисной. Поскольку в каждом из других множеств все переменные в сумме должны быть равны единице, и они являются неотрицательными, то по крайней мере одна из них должна быть положительной и входить в базисное решение.

Теорема 2. *Число множеств, содержащих две или более базисных переменных, равно не более чем $m-1$.*

Доказательство. В любом множестве из $m+p$ базисных переменных должно содержаться по крайней мере по одной переменной из каждого множества S_i , $i=0, \dots, p$, всего $p+1$ переменная. Таким образом, остается еще $m-1$ базисная переменная. Если каждая из них входит в свое множество, то $m-1$ множеств содержат по две базисные переменные. В противном случае меньше, чем $m-1$, множество содержит две переменные или более.

Предположим, что известен начальный допустимый базис **B** для задачи (1) — (2)

$$\mathbf{B} = [A^{i_1}, A^{i_2}, \dots, A^{i_{m+p}}]. \quad (3)$$

Такой базис может быть найден с помощью использования процедуры, описываемой в фазе 1 симплекс-метода. Так как каждое множество должно содержать по меньшей мере один столбец, входящий в **B**, то мы можем выбрать по одному базисному столбцу из каждого множества $S_1, \dots, S_p$ и назвать его *ключевым столбцом*, а связанные с этим столбцом переменные назвать *ключевыми переменными*. Множество S_0 не содержит никаких ключевых столбцов. Если ключевой столбец в S_i обозначить A^{k_i} , то базис в (3) может быть расчленен так, как показано на рис. 5.2, где $I_{p \times p}$ есть единичная матрица размерности p . m векторов, входящих в подматрицу $C_{p \times m}$, есть либо нулевые векторы (для столбцов из S_0) либо единичные векторы. Столбец с номером $p+t$ — это столбец из множества S_{r_t} , а столбец A^0 располагается как $(m+p)$ -й столбец в матрице **B**. Следовательно, если каждый из ключевых столбцов A^{k_i} вычесть из каждого из других базисных столбцов, входящих в его же множество, то $C_{p \times m}$ приводится к матрице, содержащей только нули, а **B** становится верхней блочно-треугольной матрицей. Пусть **T** — матрица

$$\begin{array}{c}
 \text{Ключевые столбцы} \\
 A^{k_1} A^{k_2} \dots A^{k_p}
 \end{array}
 \quad
 \begin{array}{c}
 \text{Неключевые} \\
 \text{столбцы,} \\
 t\text{-й столбец берут} \\
 \text{из множества } S_{r_t}
 \end{array}$$

$$B = \left[\begin{array}{c|c} A_{m \times p} & \hat{B}_{m \times m} \\ \hline I_{p \times p} & C_{p \times m} \end{array} \right] \begin{array}{l} \left. \vphantom{\begin{array}{c} A_{m \times p} \\ \hat{B}_{m \times m} \end{array}} \right\} m \text{ строк} \\ \left. \vphantom{\begin{array}{c} I_{p \times p} \\ C_{p \times m} \end{array}} \right\} p \text{ строк} \end{array} \quad (4)$$

Рис. 5.2.

такая, что умножение B на нее справа эквивалентно указанному вычитанию. Так как

$$Bx_B = b, \quad (5)$$

если

$$x_B = Ty_B, \quad (6)$$

то

$$[BT]y_B = b; \quad (7)$$

если BT должно иметь структуру вида

$$BT = \left[\begin{array}{c|c} A_{m \times p} & B_{m \times m} \\ \hline I_{p \times p} & 0_{p \times m} \end{array} \right], \quad (8)$$

то T должно иметь структуру вида

$$T = \left[\begin{array}{c|c} I_{p \times p} & -C_{p \times m} \\ \hline 0_{m \times p} & I_{m \times m} \end{array} \right]. \quad (9)$$

Так как $\det(T) = 1$, то T является неособой.

Если в преобразованной системе (7) все ключевые столбцы перенести в правую часть, то остающаяся система приобретает вид

$$B_{m \times m} \begin{bmatrix} y_B^{p+1} \\ \vdots \\ y_B^{p+m} \end{bmatrix} = d, \quad (10)$$

где

$$d = b - \sum_i A^{k_i}. \quad (11)$$

Определим

$$D^j = A^j - A^{k_i}, \text{ если } A^j \in S_i. \quad (12)$$

Так как $B_{m \times m} \equiv B$ образована из $\hat{B}_{m \times m}$ путем вычитания ключевых столбцов, то столбцы B задаются условием

$$B = \{D^j/A^j - \text{базисный и не ключевой}\}. \quad (13)$$

Выразим с помощью уравнений (2) ключевые переменные через неключевые:

$$x_{k_i} = 1 - \sum_{j \neq k_i} x_{n_{i-1}+j}. \quad (14)$$

Заменяя x_{k_i} с помощью этих выражений в первых m уравнениях (1), получаем

$$\sum_i D^j y_j = d. \quad (15)$$

Система (15) называется *сокращенной системой*. Заметим, что операции, приводящие к этой системе, были также использованы при получении сокращенных задач § 4.3 и 4.4. Каждое из уравнений типа (2) играет роль углового блока и x_{k_i} является базисной или зависимой переменной в блоке i .

Ясно, что рабочий базис B составляется из столбцов, входящих в сокращенную систему. Мы можем теперь доказать следующую теорему.

Теорема 3. *Рабочий базис B является базисом сокращенной системы.*

Доказательство. Необходимо доказать, что B должна быть неособой матрицей. Если столбцы в B являются зависимыми, то зависимыми являются и последние m столбцов в BT , так как они образуются путем добавления нулей к столбцам B . Таким образом, BT будет особой, а поскольку определитель T равен 1, то $\det(BT) = \det(B) \cdot \det(T) = 0$, откуда следует, что $\det(B) = 0$. Но B — базисная матрица, следовательно, неособая, а это приводит к противоречию.

Мы связали с каждым допустимым базисом B исходной системы множество ключевых столбцов $\{A^{k_i}\}$ и рабочий базис B уменьшенной размерности. Мы покажем далее, что каждая операция, требуемая при осуществлении симплекс-метода, применяемого к исходной задаче (1) — (2), может быть произведена при использовании величин, связанных с B .

Вычисление двойственных переменных. Пусть $(\pi \vdots \mu)$ — двойственные переменные, соответствующие базису B , причем π связано с первыми m уравнениями, а μ — с последними p . Эти переменные должны удовлетворять соотношению

$$(\pi' \vdots \mu)B = c_B = (0, 0, \dots, 0, 1). \quad (16)$$

Умножая справа на T , получаем

$$(\pi \vdots \mu)(BT) = c_B T = (0, 0, \dots, 0, 1). \quad (17)$$

В блочной форме имеем

$$(\pi \vdots \mu) \begin{bmatrix} A_{m \times p} & B \\ \hline I_{p \times p} & 0_{p \times m} \end{bmatrix} = (0, 0, \dots, 0, 1). \quad (18)$$

Поскольку BT блочно-треугольная, то эта система легко разрешима:

$$\pi B = \overbrace{(0, \dots, 0, 1)}^{m \text{ компонент}} \quad (19)$$

и

$$\pi A^{k_i} + \mu_i = 0, \quad i = 1, \dots, p, \quad (20)$$

или

$$\mu_i = -\pi A^{k_i}, \quad i = 1, \dots, p. \quad (21)$$

Так как B — неособая матрица, то из (19) следует, что π совпадает с m -й строкой матрицы B^{-1} , а μ_i выражаются через π по формулам (21). Таким образом, если известна матрица B^{-1} , то все множество переменных двойственной задачи может быть найдено.

Определение столбца, вводимого в базис. Это делается обычным образом путем вычисления характеристических разностей

$$\bar{c}_j = -(\pi \vdots \mu) A^j = -\pi A^j - \mu_i, \quad \text{если } A^j \in S_i \quad (22)$$

для небазисных столбцов A^j . Если

$$\min \bar{c}_j = \bar{c}_s \geq 0, \quad (23)$$

текущее базисное решение является оптимальным. В противном случае столбец A^s должен быть введен в базис B . Пусть $A^s \in S_s$.

Нахождение представления вектора, вводимого в базис, по текущему базису. Для определения номера вектора, выходящего из базиса, необходимо вычислить вектор коэффициентов разложения

$$\bar{A}_s = B^{-1} A^s. \quad (24)$$

Умножая обе части уравнения на B , получаем

$$B \bar{A}_s = A^s. \quad (25)$$

Мы вновь осуществляем преобразование переменных для того, чтобы привести $\mathbf{B}$ к блочно-треугольной форме

$$\bar{\mathbf{A}}^s = \mathbf{T}\mathbf{Z}^s, \quad (26)$$

так что

$$(\mathbf{B}\mathbf{T})\mathbf{Z}^s = \mathbf{A}^s. \quad (27)$$

В блочной форме эта матрица имеет вид

$$\left[\begin{array}{c|c} A_{m \times p} & B \\ \hline I_{p \times p} & 0_{p \times m} \end{array} \right] \cdot \begin{bmatrix} Z_1^s \\ \vdots \\ Z_p^s \\ Z_{p+1}^s \\ \vdots \\ Z_{p+m}^s \end{bmatrix} = \begin{bmatrix} A_s \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{положение } \sigma. \quad (28)$$

Из последних p уравнений вытекает

$$Z_i^s = 0, \quad 1 \leq i \leq p, \quad i \neq \sigma, \quad (29)$$

$$Z_\sigma^s = 1. \quad (30)$$

Используя эти величины в первых m уравнениях, получаем

$$A^{k\sigma} + B \begin{bmatrix} Z_{p+1}^s \\ \vdots \\ Z_{p+m}^s \end{bmatrix} = A^s. \quad (31)$$

Обозначим

$$\bar{D}^s = \begin{bmatrix} Z_{p+1}^s \\ \vdots \\ Z_{p+m}^s \end{bmatrix} = \begin{bmatrix} \bar{D}_1^s \\ \vdots \\ \bar{D}_m^s \end{bmatrix}. \quad (32)$$

Тогда из (31) следует

$$B\bar{D}^s = A^s - A^{k\sigma}. \quad (33)$$

Это уравнение имеет решение

$$\bar{D}^s = B^{-1}(A^s - A^{k\sigma}). \quad (34)$$

Вектор $\bar{D}^s$ является представлением по векторам рабочего базиса вектора $A^s - A^{k\sigma}$, входящего в базис сокращенной системы. Он легко

вычисляется, если B^{-1} известна. Тогда из (26) следует:

$$\bar{A}^s = \begin{bmatrix} \bar{A}_1^s \\ \vdots \\ \bar{A}_p^s \\ \dots \\ \bar{A}_{p+1}^s \\ \vdots \\ \bar{A}_{p+m}^s \end{bmatrix} = \begin{bmatrix} I_{p \times p} & -C_{p \times m} \\ \hline & \\ \hline 0_{m \times p} & I_{m \times m} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \dots \\ \bar{D}^s \end{bmatrix} \leftarrow \begin{matrix} \text{позиция.} \\ \sigma \end{matrix} \quad (35)$$

В силу (35) $\bar{A}_i^s$ есть скалярное произведение строки i матрицы T на вектор Z^s . Напомним, что t -й столбец матрицы $-C_{p \times m}$ имеет -1 в строке r_t (где столбец $p+t$ матрицы B является элементом S_{r_t}) и нули в других местах. t -й столбец состоит только из нулей, если $r_t = 0$. Таким образом, при вычислении скалярного произведения строки i матрицы $-C_{p \times m}$ на $\bar{D}^s$ ненулевое значение получается только, когда $r_t = i$. Если мы введем в рассмотрение множества

$$R(i) = \{t/t \in \{1:m\}, r_t = i\}$$

то $\bar{A}_i^s$ дается соотношением

$$\bar{A}_i^s = \begin{cases} -\sum_{t \in R(i)} \bar{D}_t^s, & 1 \leq i \leq p, i \neq \sigma, \\ 1 - \sum_{t \in R(i)} \bar{D}_t^s, & i = \sigma, \end{cases} \quad (36)$$

$$\bar{A}_{p+t}^s = \bar{D}_t^s, \quad t = 1, \dots, m. \quad (37)$$

Если известен вектор $\bar{D}^s$, то $\bar{A}_i^s$ легко вычисляется с помощью этих формул.

Выбор столбца, выводимого из базиса. Это делается обычным способом, путем вычисления

$$\min_{\bar{A}_i^s > 0} \frac{\bar{b}_i}{\bar{A}_i^s} = \frac{\bar{b}_r}{\bar{A}_r^s} = \theta, \quad (38)$$

где $\bar{b}_i$ — текущее значение i -й базисной переменной. Если все $\bar{A}_i^s \leq 0$, то задача (1) — (2) имеет неограниченное решение. В противном случае столбец r матрицы B , $A^{r'}$ выводится из базиса. Предполагаем $A^{r'} \in S_p$.

Вычисление значений базисных переменных. Эта операция производится согласно стандартным формулам

$$\begin{aligned}(\bar{b}_i)_{\text{нов}} &= (\bar{b}_i)_{\text{ст}} - \theta A_{i,i}^s, \quad i = 1, \dots, m+p, i \neq r, \\ (\bar{b}_r)_{\text{нов}} &= \theta.\end{aligned}\quad (39)$$

Наряду с этим новые базисные переменные могут быть вычислены по значениям базисных переменных в сокращенной системе, $B^{-1}d$, с использованием формул, сходных с (36) — (37).

Вычисление обратной матрицы нового рабочего базиса. Для удобства перепишем формулы (4) и (8) в следующем виде:

$$B = \left[\begin{array}{c|c} A_{m \times p} & \hat{B}_{m \times m} \\ \hline I_{p \times p} & C_{p \times m} \end{array} \right], \quad BT = \left[\begin{array}{c|c} A_{m \times p} & B_{m \times m} \\ \hline I_{p \times p} & 0_{p \times m} \end{array} \right].$$

Напомним, что столбец, вводимый в B , есть $A^s \in S_\sigma$, а столбец, выводимый из базиса, есть $A^{ir} \in S_\rho$. Требуется рассмотреть два случая.

Случай 1. A^{ir} не является ключевым столбцом. В этом случае столбец, выводимый из базиса, есть один из последних m столбцов матрицы B , например, столбец $p+i_2$. Если A^{ir} замещается в B столбцом A^s , а затем B преобразуется к блочно-треугольной форме типа (8), то единственное изменение в рабочем базисе B состоит в том, что столбец B , соответствующий A^{ir} , а именно, $A^{ir} - A^{k\sigma}$, заменяется на $A^s - A^{k\sigma}$. Таким образом, для того, чтобы вычислить новую матрицу B^{-1} , присоединим к ней столбец

$$\bar{D}^s = B^{-1}(A^s - A^{k\sigma}), \quad (40)$$

и осуществим ведущую операцию, используя компоненту i_2 столбца $\bar{D}^s$ в качестве ведущего элемента. Заметим, что $\bar{D}^s$ уже вычислено по формуле (34). Конечно, все другие величины, используемые в следующем цикле, должны быть также вычислены заново. Это касается столбцов A^{ki} , которые появляются в уравнениях (21) для двойственных переменных u_i и индексов r_i , которые используются для определения множеств $R(i)$ уравнений, предшествующих (36). Единственное изменение в этом множестве заключается в том, что индекс r_{i_2} становится равным σ , так что столбец i_2 матрицы C становится σ -м единичным вектором. Запись этого изменения завершает случай 1.

Случай 2. A^{ir} является ключевым столбцом.

Подслучай а). A^{ir} и A^s принадлежат различным множествам, т. е. $\rho \neq \sigma$. В этом подслучае мы замещаем ключевой столбец в S_ρ столбцом A^s , не принадлежащим S_ρ . Значит, должен быть найден

новый ключевой столбец из S_p . По теореме 1 S_p должно содержать по крайней мере одну базисную переменную после того, как будет выведен столбец A^{ir} . Следовательно, один из неключевых столбцов B принадлежит S_p . Выберем любой из таких столбцов, например, столбец $p + i_2$, и назовем его A^k , где $k = p + i_2$. Он становится ключевым, и мы полагаем $A^{ir} \equiv A^{kp}$. Если эта замена проведена, и результирующая матрица B трансформирована к блочно-треугольной форме вида (8), то столбцы B , имеющие вид $A^{ji} - A^{kp}$, $j_i \in S_p$, заменяются на следующие:

$$A^{ji} - A^{kp} \leftarrow A^{ji} - A^k, \quad j_i \in S_p, \quad j_i \neq k, \quad (41)$$

$$A^k - A^{kp} \leftarrow A^{kp} - A^k. \quad (42)$$

Эти замены могут быть выполнены путем умножения $A^k - A^{kp}$ на -1 и добавления результата к каждому столбцу $A^{ji} - A^{kp}$, $j_i \in S_p$. В матричной форме имеем

$$B \leftarrow BT_1, \quad (43)$$

где

$$T_1 = \begin{bmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & \ddots & & & & & & \\ & & & & 1 & & & & & \\ 0 & \dots & -1 & \dots & 0 & \dots & -1 & \dots & 0 & \dots & -1 & \dots & 0 \\ & & & & & 1 & & & & & & \\ & & & & & & \ddots & & & & & \\ & & & & & & & \ddots & & & & \\ & & & & & & & & 1 & & & \end{bmatrix} \leftarrow \text{строка } i_2. \quad (44)$$

Элементы -1 появятся в столбцах, соответствующих $A^{ji} \in S_p$. Так как в p -й строке матрицы C также появятся -1 в этих положениях, строка i_2 в T_1 равна с минусом строке p матрицы C . Матрица B^{-1} , таким образом, замещается матрицей

$$B^{-1} \leftarrow T_1^{-1} B^{-1}. \quad (45)$$

Поскольку применение преобразования столбцов, задаваемого матрицей T_1 , дважды приводит вновь к исходной матрице B , то

$$T_1^{-1} = T_1, \quad (46)$$

поэтому

$$B^{-1} \leftarrow T_1 B^{-1}. \quad (47)$$

Замена A^{ir} на A^k также требует, чтобы вектор A^{ir} был заменен на вектор A^k . Если эти замены проведены, то ситуация окажется такой же, какая бы имела место в случае 1. Осуществление процедур, указанных в конце описания случая 1, завершает и здесь проведение итерации.

Подслучай б) $\rho = \sigma$. Здесь ключевой столбец в S_ρ заменяется другим столбцом из S_ρ . Если имеются неключевые столбцы из S_ρ , то заменяем один из них на A^{ir} так, как описано в подслучае а). Затем применяем операции случая 1. Если не имеется никаких неключевых столбцов из S_ρ , то A^* заменяется на A^{ir} в B . Пусть полученная матрица преобразована к блочно-треугольному виду (8). Так как последние m столбцов B не содержат ни одного столбца из S_ρ , то подматрица B (8) остается неизменной, т. е. рабочий базис остается тем же самым. Единственное изменение состоит в том, что A^* замещается A^{ir} в верхнем левом блоке B . Это завершает рассмотрение подслучая б).

Проведение описанной выше замены в тех случаях, когда она возможна, значительно упрощает этот подслучай. Когда замена возможна, мы переходим к предшествующим случаям, а когда нет, то рабочий базис остается неизменным.

Сводка алгоритма. После выполнения вычисления новой обратной матрицы B^{-1} может начаться новая итерация. Блок-схема процедуры показана на рис. 5.3.

Вычислительный опыт. Данциг и Ван Слайк сообщают [2], что в задаче, содержащей 2813 переменных, $p=780$, $m=39$, эта процедура, осуществляемая на ЦВМ IBM-7094, потребовала 15 минут для решения. Проводилось примерно 50 итераций в минуту. По оценке, вычисления с помощью стандартной общей программы симплекс-метода, применяемого к этой задаче, потребовали бы 150 минут, так что уменьшение времени вычисления достигает десятикратного.

Модификации на случай, когда в последних p уравнениях содержатся отрицательные коэффициенты. Как уже указывалось, может представлять практический интерес случай, когда последние p уравнений задачи (1) — (2) могут содержать не только коэффициенты 1, но и -1 . При этом теоремы 1 и 2 сохраняют свою силу. Каждый из ключевых столбцов в базисной матрице B может быть выбран так, чтобы он имел коэффициент 1, поскольку каждое множество S_i должно иметь по крайней мере один такой столбец в любом базисе. При формировании рабочего базиса, если окажется, что неключевой базисный столбец имеет коэффициент -1 , то ключевой столбец к нему добавляется, а не вычитается. Теорема 3 сохраняет свою силу для базиса, сформированного таким образом, равно как и соотношение (19) — (21) для двойственных переменных. При вычислении характеристических разно-

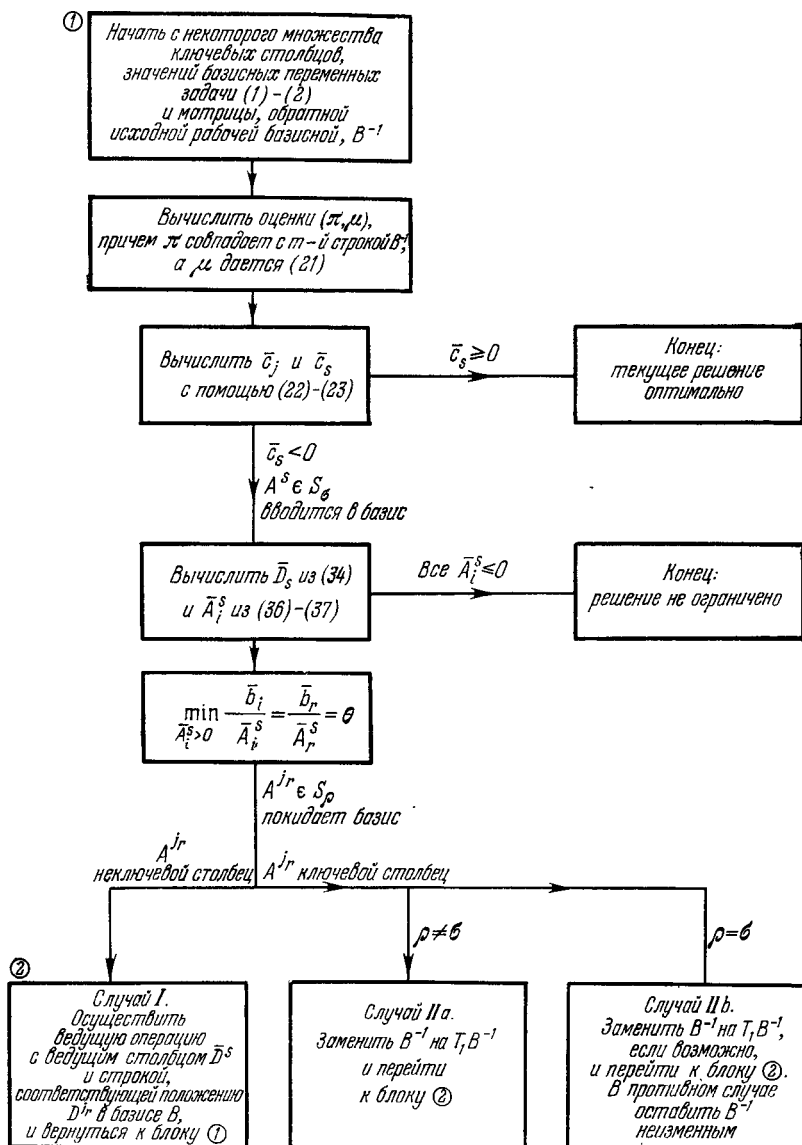


Рис. 5.3. Блок-схема алгоритма.

стей для столбца, содержащего коэффициент -1 , μ_i добавляется к $-\mu A^i$, а не вычитается, как в исходной формуле (22). Сходные малые модификации должны быть также введены и в формулы для вычисления $\bar{A}^i$ и для преобразования обратной матрицы рабочего базиса (см. задачи в конце этой главы). За исключением этих изменений алгоритм действует так же, как и ранее.

5.4.2. Пример применения метода учета обобщенных двусторонних ограничений на переменные

Рассмотрим задачу, заимствованную из [2]:

$$\max x_0$$

при ограничениях, представленных в виде таблицы

S_0		S_1		S_2		S_3		S_4		S_5	
A^0	A^1	A^2	A^3	A^4	A^5	A^6	A^7	A^8	A^9	b	
1	0	2	0	3	4	5	1	-1	-12	15	
1	1	-1	0	2	1	4	2	-3	6	7	
0	0	0	1	0	0	0	0	0	0	0	
	1	1	1							1	
				1						1	
					1					1	
						1	1			1	
								1	1	1	
*	*	*	*	*	*	*		*			
	✓			✓	✓	✓		✓			

причем все переменные являются неотрицательными. Начальный базис образуется из столбцов, помеченных *. Выберем в качестве ключевых столбцов столбцы, помеченные ✓. Начальный базис в блочной форме имеет вид

$$B = \begin{array}{c} \text{Ключевые столбцы} \\ \begin{array}{ccccc|ccc} A^1 & A^2 & A^3 & A^4 & A^5 & A^6 & A^7 & A^8 & A^9 \\ \hline 0 & 3 & 4 & 5 & -1 & 2 & 0 & 1 \\ 1 & 2 & 1 & 4 & -3 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 1 & & & & & 1 & 1 & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \end{array} \end{array},$$

$$A^{k_1} \quad A^{k_2} \quad A^{k_3} \quad A^{k_4} \quad A^{k_5}$$

Величины μ_i даются формулами

$$\mu_i = -\pi A^{k_i}, \quad i = 1, \dots, 5,$$

откуда следует

$$\mu_1 = -1/2, \quad \mu_2 = -5/2, \quad \mu_3 = -5/2, \quad \mu_4 = -9/2, \quad \mu_5 = 2.$$

Вычисление характеристических разностей для столбцов A^7, A^9 . В силу (22) имеем

$$\bar{c}_j = -\pi A_j - \mu_i, \quad \text{если } A^j \in S_i,$$

так что

$$\begin{aligned} \bar{c}_7 &= -\pi A^7 - \mu_4 = -3/2 - (-9/2) = 3, \\ \bar{c}_9 &= -\pi A^9 - \mu_5 = -3 - 2 = -5. \end{aligned}$$

Так как $\bar{c}_7 = \max \bar{c}_j > 0$, то $A^7 \in S_4$ вводится в базис.

Вычисление $\bar{A}^7$. В силу (34) получаем

$$\bar{D}^7 = B^{-1}(A^7 - A^6) = \begin{bmatrix} 1/4 & -1/4 & -1/4 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} -4 \\ -2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1/2 \\ 0 \\ -3 \end{bmatrix} = \begin{bmatrix} \bar{D}_1^7 \\ \bar{D}_2^7 \\ \bar{D}_3^7 \end{bmatrix}.$$

Используя (36) — (37) при $p=5$, $\sigma=4$, $r_1=r_2=1$, $r_3=0$, вычисляем

$$\begin{aligned} \bar{A}_1^7 &= -(\bar{D}_1^7 + \bar{D}_2^7) = 1/2, & \bar{A}_2^7 &= \bar{A}_3^7 = \bar{A}_5^7 = 0, & \bar{A}_4^7 &= 1, \\ \bar{A}_6^7 &= \bar{D}_1^7 = -1/2, & \bar{A}_7^7 &= \bar{D}_2^7 = 0, & \bar{A}_8^7 &= \bar{D}_3^7 = -3. \end{aligned}$$

Определение переменной, выводимой из базиса.

$$\theta = \min_{\bar{A}_i^7 > 0} \frac{\bar{b}_i}{\bar{A}_i^7} = \min \left(\frac{\bar{b}_1}{\bar{A}_1^7}, \frac{\bar{b}_4}{\bar{A}_4^7} \right) = \min(1, 1) = 1.$$

Таким образом, либо первый, либо четвертый столбец матрицы B , т. е. A^1 или A^6 , выводится из базиса. Мы выбираем $A^6 \in S_4$, так как это оставляет B^{-1} неизменной для следующей итерации.

Новые значения базисных переменных. Поскольку A^6 есть четвертый столбец B , то

$$\begin{aligned} \bar{b}_i &\leftarrow \bar{b}_i - \theta \bar{A}_i^7, \quad i \neq 4, \\ \bar{b}_4 &= \theta, \end{aligned}$$

так что новый базис и новые значения базисных переменных таковы:

$$\begin{aligned} B &= [A^1 A^4 A^5 A^7 A^8 | A^2 A^3 A^0], \\ x_B &= (0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 6). \end{aligned}$$

Вычисление обратной матрицы для нового рабочего базиса. Поскольку A^6 , столбец, который выводится из базиса, является ключевым и столбцы, вводимые и выводимый, принадлежат одному и тому же множеству, то имеет место случай 2а. Рабочий базис не изменяется, поскольку нет неключевых столбцов из S_4 .

Итерация 2

Двойственные переменные. Поскольку B^{-1} не изменяется, то

$$\pi = (1/2, 1/2, 1/2), \\ \mu_1 = -1/2, \mu_2 = -5/2, \mu_3 = -5/2, \mu_4 = -3/2, \mu_5 = 2.$$

Вычисление характеристических разностей для A^6 и A^9 :

$$\bar{c}_6 = -\pi A^6 - \mu_4 = -3, \quad \bar{c}_9 = -\pi A^9 - \mu_5 = 1.$$

Поскольку $\bar{c}_9 > 0$, то $A^9 \in S_5$ входит в базис.

Вычисление $\bar{A}^9$. В силу (34) имеем

$$\bar{D}^9 = B^{-1}(A^9 - A^8) = \begin{bmatrix} 1/4 & -1/4 & -1/4 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} -11 \\ 9 \\ 0 \end{bmatrix} = \begin{bmatrix} -5 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} \bar{D}_1^9 \\ \bar{D}_2^9 \\ \bar{D}_3^9 \end{bmatrix}.$$

Используем (36) — (37) при $p=5, \sigma=5, r_1=r_2=1, r_3=0$:

$$\bar{A}_1^9 = -(\bar{D}_1^9 + \bar{D}_2^9) = 5, \quad \bar{A}_2^9 = \bar{A}_3^9 = \bar{A}_4^9 = 0, \quad \bar{A}_5^9 = 1, \\ \bar{A}_6^9 = \bar{D}_1^9 = -5, \quad \bar{A}_7^9 = \bar{D}_2^9 = 0, \quad \bar{A}_8^9 = \bar{D}_3^9 = -1.$$

Определение переменной, выводимой из базиса:

$$\theta = \min \left(\frac{\bar{b}_1}{\bar{A}_1^9}, \frac{\bar{b}_5}{\bar{A}_5^9} \right) = \min(0, 1) = 0,$$

поэтому $A^1 \in S_1$ покидает базис.

Вычисление новой B^{-1} . Поскольку столбец, выводимый из базиса, является ключевым и принадлежит другому множеству, чем столбец, вводимый в него, мы должны заменить A^1 на один из трех последних столбцов в матрице B , которые принадлежат S_1 . В данном случае A^1 может быть заменен на вектор A^2 либо на вектор A^3 . Пусть выбран вектор A^2 . Тогда текущий рабочий базис

$$B_{\text{ср}} = [A^2 - A^1, A^3 - A^1, A^9]$$

заменяется на

$$\tilde{B} = [A^1 - A^2, A^3 - A^2, A^0].$$

Для того чтобы осуществить это преобразование, умножим $A^2 - A^1$ на -1 и добавим результат к $A^3 - A^1$, т. е. умножим $B_{ст}$ справа на

$$T_1 = \begin{bmatrix} -1 & -1 & \\ & 1 & \\ & & 1 \end{bmatrix}.$$

Тогда

$$\tilde{B}_1 = B_{ст} T_1,$$

так что

$$\tilde{B}^{-1} = T_1^{-1} B_{ст}^{-1} = T_1 B_{ст}^{-1} = \begin{bmatrix} -1/4 & 1/4 & -3/4 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 1/2 \end{bmatrix}.$$

После замены соответствующий базис исходной системы примет вид

$$B = [A^2 A^4 A^5 A^7 A^8 | A^1 A^3 A^0].$$

Пусть теперь A^1 выводится из этого базиса, а A^9 вводится. Так как A^1 по-прежнему не является ключевым столбцом, то имеет место случай 1 общей процедуры, причем $A^9 - A^8$ заменяется столбцом 1 матрицы $\tilde{B}$. Для того чтобы найти новую обратную матрицу, умножим $A^9 - A^8$ на $\tilde{B}^{-1}$:

$$\tilde{B}^{-1}(A^9 - A^8) = \begin{bmatrix} -1/4 & 1/4 & -3/4 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} -11 \\ 9 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ -1 \end{bmatrix}.$$

Ведущим является элемент в строке 1, так что ведущая операция приводит к новой обратной матрице

$$B_{нов}^{-1} = \begin{bmatrix} -1/20 & 1/20 & -3/20 \\ 0 & 0 & 1 \\ 9/20 & 11/20 & 7/20 \end{bmatrix}.$$

Новый базис исходной задачи таков:

$$B = [A^2 A^4 A^5 A^7 A^8 | A^9 A^3 A^0]$$

с базисными переменными

$$x_B = [1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 6].$$

Итерация 3

Двойственные переменные. Новый вектор π есть третья строка матрицы $B_{нов}^{-1}$:

$$\pi = (9/20, 11/20, 7/20),$$

$$\mu_1 = -7/20, \mu_2 = -49/20, \mu_3 = -47/20, \mu_4 = -31/20, \mu_5 = 42/20.$$

Вычисление характеристических разностей:

$$\begin{aligned}\bar{c}_1 &= -\pi A^1 - \mu_1 = -11/20 + 7/20 = -4/20, \\ \bar{c}_6 &= -\pi A^6 - \mu_4 = -99/20 - 31/20 = -130/20.\end{aligned}$$

Поскольку все $\bar{c}_i \leq 0$, то текущее решение является оптимальным.

§ 5.5. ОБОБЩЕНИЕ НА БЛОЧНО-ДИАГОНАЛЬНУЮ СТРУКТУРУ

Описанная выше методика допускает естественное обобщение на задачи с матрицей ограничений, имеющей блочно-диагональную структуру. Последующее изложение представляет собой обобщение работы, проведенной в [3]. Пусть задача такова:

$$\begin{aligned}& \min (x_0)_1, \\& \left. \begin{array}{l} m_0 \text{ строк} \left\{ \overbrace{A_0 x_0}^{S_0} + \overbrace{A_1 x_1}^{S_1} + \overbrace{A_2 x_2}^{S_2} + \dots + \overbrace{A_p x_p}^{S_p} = b, \right. \\ m_i \text{ строк} \left\{ \begin{array}{ccc} B_1 x_1 & & = b_1, \\ & B_2 x_2 & = b_2, \\ & & \vdots \\ & & B_p x_p = b_p, \end{array} \right\} \\ m_p \text{ строк} \left\{ \begin{array}{ccc} & & \vdots \\ & & \vdots \\ & & \vdots \end{array} \right\} \end{array} \right\} \quad (1) \\& x_i \geq 0, \quad i = 0, \dots, p, \quad (2)\end{aligned}$$

где A_i есть $m_0 \times n_i$ -матрицы, B_i есть $m_i \times n_i$ -матрицы, $(x_0)_1$ — первая компонента x_0 . Излагаемые далее методы позволяют проводить симплексную процедуру так, чтобы на всех итерациях сохранять рабочий базис, имеющий размерность только $m_0 \times m_0$, а также базисные матрицы для каждой из B_i . Как и в предшествующем параграфе, процедура может быть построена с помощью преобразования произвольной базисной матрицы исходной задачи к блочно-треугольной форме.

В течение всего этого параграфа мы сделаем следующее предположение:

Предположение. Система (1)—(2) имеет полный ранг.

Заметим, что из этого следует, что каждая B_i имеет полный ранг (в противном случае некоторые из строк системы (2) были бы линейно зависимы). Следовательно, $n_i \geq m_i$, и мы можем также предположить $n_i > m_i$, поскольку в противном случае переменные x_i могли бы быть исключены. Как и ранее, удобно определить S_i как множество столбцов или переменных (в зависимости от контекста) блочной задачи, имеющей компоненты, содержащиеся в B_i . Множество S_0 содержит те столбцы, которые имеют нулевые компоненты всюду, кроме первых m_0 строк.

Метод основывается на следующей теореме.

Теорема 1. Любая базисная матрица $\mathbf{B}$ задачи (1)–(2) может быть представлена в виде следующей блочной структуры:

$$\mathbf{B} = \begin{array}{c} \begin{array}{c} \text{Неключевые столбцы:} \\ t\text{-й столбец из множества} \end{array} \\ \begin{array}{c} \text{Ключевые столбцы} \end{array} \\ \begin{array}{c} S_{r_i} \\ \overbrace{\mathbf{B}}^{m_0 \text{ строк}} \\ \mathbf{C} \\ \underbrace{\phantom{\mathbf{C}}}_{m_0 \text{ столбцов}} \end{array} \end{array}, \quad (3)$$

$$\mathbf{B} = \begin{bmatrix} \overbrace{A_{11} \ A_{21} \ \dots \ A_{p1}}^{m_0 \text{ строк}} & \overbrace{\mathbf{B}}^{S_{r_i}} \\ B_{11} & \mathbf{C} \\ & \underbrace{\phantom{\mathbf{C}}}_{m_0 \text{ столбцов}} \\ & B_{p1} \end{bmatrix},$$

где каждая B_{ii} имеет размерность $m_i \times m_i$ и является неособой.

Доказательство. Рассмотрим множество столбцов $\mathbf{B}$, принадлежащих множеству S_i . Собранные вместе, они образуют матрицу следующей структуры:

$$m_0 \text{ строк} \left\{ \begin{bmatrix} A \\ \dots \\ 0 \\ \dots \\ D \\ \dots \\ 0 \end{bmatrix} \right.$$

$$\text{блок } i \left\{ \begin{bmatrix} A \\ \dots \\ 0 \\ \dots \\ D \\ \dots \\ 0 \end{bmatrix} \right.$$

Все другие базисные столбцы имеют нулевые компоненты в блоке i . Если бы D имела ранг меньше, чем m_i , то некоторые из строк $\mathbf{B}$ были бы зависимы, что невозможно. Следовательно, D имеет ранг m_i и содержит квадратную неособую подматрицу, которая обозначена B_{ii} .

Далее повсюду мы будем предполагать, что столбец, ассоциированный с $(x_0)_i$, всегда является базисным, и что он является последним столбцом $\mathbf{B}$, т. е. m_0 -м неключевым столбцом.

Заметим, что единичная матрица $I_{p \times p}$ в базисе $\mathbf{B}$ предшествовавшего параграфа (см. уравнение (4) этого параграфа) теперь замещена блочно-диагональной матрицей с неособыми блоками. Поэтому необходимо будет запоминать и преобразовывать обратные матрицы для этих блоков. В предшествовавшем параграфе каждый из этих блоков сводился к единичному элементу, так что никаких таких операций не требовалось.

В дальнейшем наша цель будет состоять в том, чтобы, используя тот факт, что матрицы B_{ii} являются неособыми, построить преобразующую матрицу $\mathbf{T}$ такую, чтобы $\mathbf{BT}$ оказалась блочно-треугольной. В частности, подматрица $\mathbf{C}$ в (3) должна быть преобразована в нулевую матрицу; при этом $\hat{\mathbf{B}}$ преобразуется в рабочий базис. Так как ключевые столбцы должны быть

оставлены неизменными, первые $m_1 + m_2 + \dots + m_p$ столбцов T должны быть единичными векторами. Таким образом, T должно иметь вид

$$T = \left[\begin{array}{c|c} I_1 & V \\ \hline 0 & E \end{array} \right], \quad (4)$$

где I_1 — единичная матрица, а V и E должны быть выбраны так, чтобы удовлетворить матричному уравнению

$$\left[\begin{array}{c|c} A_1 & \hat{B} \\ \hline \hat{B}_1 & C \end{array} \right] \left[\begin{array}{c|c} I_1 & V \\ \hline 0 & E \end{array} \right] = \left[\begin{array}{c|c} A_1 & B \\ \hline \hat{B}_1 & 0 \end{array} \right], \quad (5)$$

причем A_1 и $\hat{B}_1$ представляют собой соответствующие блоки матрицы B :

$$[A_1] = [A_{11} A_{21} \dots A_{p1}], \quad (6)$$

$$[\hat{B}_1] = \begin{bmatrix} B_{11} & & & \\ & B_{21} & & \\ & & \ddots & \\ & & & B_{p1} \end{bmatrix}. \quad (7)$$

Приравнявая соответствующие блоки в правых и левых частях тождества (5), получаем

$$\hat{B}_1 V + CE = 0, \quad (8)$$

$$A_1 V + \hat{B} E = B. \quad (9)$$

Эти уравнения достаточно просто удовлетворить. Выберем

$$E = I, \quad (10)$$

$$V = -\hat{B}_1^{-1} C, \quad (11)$$

так что матрица преобразования T приобретает вид

$$T = \left[\begin{array}{c|c} I_1 & V \\ \hline 0 & I_2 \end{array} \right]. \quad (12)$$

Подставляя (10) в (9), получим следующее выражение для матрицы B :

$$B = \hat{B} - A_1 V. \quad (13)$$

Так как T , а следовательно, и подматрица V играют в дальнейшем ключевую роль, то мы должны изучить структуру матрицы V . Матрица $\hat{B}_1$ блочно-диагональна с неособыми блоками, так

что (11) может быть записана в виде

$$[V] = - \begin{bmatrix} B_{11}^{-1} & & & \\ & B_{21}^{-1} & & \\ & & \ddots & \\ & & & B_{p1}^{-1} \end{bmatrix} [C]. \quad (14)$$

Так как t -й неключевой столбец B — из S_{r_t} , то при $r_t=0$ столбец t матрицы C является нулевым вектором, равно как и столбец t матрицы V . Если $r_t \neq 0$, то столбец t матрицы C имеет вид

$$\left\{ \begin{bmatrix} 0 \\ \hline R^t \\ \hline 0 \end{bmatrix} \right\} \text{ блок } r_t,$$

так что столбец t матрицы V таков:

$$\left\{ \begin{bmatrix} 0 \\ \hline -\bar{R}^t \\ \hline 0 \end{bmatrix} \right\} \text{ блок } r_t, \quad \bar{R}^t = B_{r_t}^{-1} R^t. \quad (15)$$

Таким образом, столбцы V могут запоминаться путем записи только индексов r_i и соответствующих векторов $\bar{R}^i$.

Матрица B в (5) и (13) является рабочим базисом процедуры. Однако мы должны еще доказать следующее утверждение:

Теорема 2. *B является неособой матрицей.*

Доказательство. Если B — особая, то особой является и матрица BT , так как подматрица, лежащая ниже B , целиком состоит из нулей. Однако $\det(BT) = \det(B) \cdot \det(T) = \det(B)$, так как $\det(T) = 1$. Таким образом, $\det(BT) = 0 \Rightarrow \det(B) = 0$, что и приводит к противоречию.

Для того чтобы пояснить предшествующий материал, рассмотрим задачу, ранее рассматривавшуюся и в § 4.4:

$$\begin{array}{cccccccccccccc} & & & & & \min x_0, & & & & & & & \\ & A^0 & A^1 & A^2 & A^3 & A^4 & A^5 & A^6 & A^7 & A^8 & A^9 & A^{10} & A^{11} \\ - & x_0 & -x_1 & -x_2 & & & -2x_5 & -x_6 & & & & & =0, \\ & & x_1 & +2x_2 & & & +2x_5 & +x_6 & & & +x_{10} & & =40, \\ & & x_1 & +x_2 & & & +4x_5 & +2x_6 & & & & +x_{11} & =50, \\ & & x_1 & +3x_2 & +x_3 & & & & & & & & =30, \\ & & 2x_1 & +x_2 & & +x_4 & & & & & & & =20, \\ & & & & & & x_5 & & +x_7 & & & & =10, \\ & & & & & & & x_6 & & +x_8 & & & =10, \\ & & & & & & x_5 & +x_6 & & & +x_9 & & =15, \\ & & & & & & x_i \geq 0, & i \in 1, \dots, 11. \end{array}$$

Выделим следующую подматрицу:

$$\begin{array}{c}
 \begin{array}{ccccc}
 A^1 & A^4 & A^7 & A^8 & A^9 \\
 \begin{bmatrix}
 -1 & & & & \\
 1 & & & & \\
 1 & & & & \\
 \hline
 1 & & & & \\
 2 & 1 & & & \\
 \hline
 & & 1 & & \\
 & & & 1 & \\
 & & & & 1
 \end{bmatrix}
 \end{array}
 \end{array}
 \begin{array}{c}
 \begin{array}{ccc}
 A^{10} & A^{11} & A^{12} \\
 \begin{bmatrix}
 -1 & -2 & -1 \\
 2 & 2 & \\
 1 & 4 & \\
 \hline
 3 & & \\
 1 & & \\
 \hline
 & 1 & \\
 & & 1
 \end{bmatrix}
 \end{array}
 \end{array}
 \end{array}
 = B.$$

Легко доказать, что B является неособой, так что она является (возможно недопустимой) базисной матрицей исходной задачи. Вычислим

$$B_{11}^{-1} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}, \quad B_{21}^{-1} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}.$$

Тогда

$$\bar{R}^1 = B_{11}^{-1} R^1 = \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ -5 \end{bmatrix},$$

$$\bar{R}^2 = B_{21}^{-1} R^2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix},$$

так что

$$\begin{array}{c}
 T = \begin{bmatrix}
 1 & & & & -3 \\
 & 1 & & & 5 \\
 & & 1 & & -1 \\
 & & & 1 & -1 \\
 \hline
 & & & & 1 \\
 & 0 & & & 1 \\
 & & & & & 1
 \end{bmatrix} \\
 \\
 BT = \begin{bmatrix}
 -1 & & & & 2 & -2 & -1 \\
 1 & & & & -1 & 2 & \\
 1 & & & & -2 & 4 & \\
 \hline
 1 & & & & & & \\
 2 & 1 & & & & & 0 \\
 \hline
 & & 1 & & & & \\
 & & & 1 & & & \\
 & & & & 1 & &
 \end{bmatrix}.
 \end{array}$$

Таким образом, рабочий базис имеет вид

$$B = \begin{bmatrix} 2 & -2 & -1 \\ -1 & 2 & 0 \\ -2 & 4 & 0 \end{bmatrix}.$$

Так же как в предшествовавшем параграфе, матрицу B можно интерпретировать как базисную матрицу для сокращенной системы с m_0 уравнениями. Пусть уравнения (2) переписаны в виде

$$B_{i1}x_{i1} + B_{i2}x_{i2} = b_i, \quad i = 1, \dots, p, \quad (16)$$

причем (1) расчленено сходным образом:

$$A_0x_0 + \sum_{i=1}^p (A_{i1}x_{i1} + A_{i2}x_{i2}) = b. \quad (17)$$

Так как каждая из B_{i1} является неособой, то (16) может быть использовано для того, чтобы исключить переменные x_{i1} :

$$x_{i1} = B_{i1}^{-1}b_i - B_{i1}^{-1}B_{i2}x_{i2}. \quad (18)$$

Сокращенная система получается путем подстановки (18) в (17):

$$A_0x_0 + \sum_{i=1}^p (A_{i2} - A_{i1}B_{i1}^{-1}B_{i2})x_{i2} = b - \sum_{i=1}^p A_{i1}B_{i1}^{-1}b_i. \quad (19)$$

Нетрудно убедиться, что это та же самая сокращенная система, что и используемая в методе расчленения Розена (глава IV). В силу (13), столбец t матрицы B равен столбцу t матрицы $\hat{B}$ (назовем его Q^t) минус A_{i1} , умноженное на столбец t матрицы V . Этот столбец V состоит из нулей за исключением элементов r_{ti} , где содержится вектор $-B_{r_{ti}}^{-1}R^t$, так что столбец t матрицы B имеет вид

$$Q^t - A_{r_{t1}}B_{r_{t1}}^{-1}R^t. \quad (20)$$

Так как Q^t есть столбец $A_{r_{t2}}$, а R^t — столбец $B_{r_{t2}}$, то это выражение ассоциировано с некоторыми компонентами $x_{r_{t2}}$ в (19). Следовательно, B есть базисная матрица сокращенной системы.

Проверим далее, как могут быть осуществлены операции модифицированного симплекс-метода с использованием только величин, ассоциированных с рабочим базисом. Эти операции требуют только решения двух систем линейных уравнений с коэффициентными матрицами B' и B (одна для двойственных переменных, другая — для вычисления разложения вводимого вектора по старому базису). Представление B в виде треугольной матрицы весьма упрощает эти решения.

Определение двойственных переменных. Вектор двойственных переменных $\pi = (\pi_0, \pi_1, \dots, \pi_p)$ удовлетворяет системе

$$\pi B = c_B \quad (21)$$

или, поскольку только $(x_0)_1$ имеет ненулевой коэффициент в целевой функции, и соответствующий ему столбец есть самый правый столбец матрицы $\mathbf{B}$, это соотношение можно записать в виде

$$\pi \mathbf{B} = (0, 0, \dots, 1). \quad (22)$$

Умножая его справа на $\mathbf{T}$, получаем

$$\pi (\mathbf{B}\mathbf{T}) = (0, 0, \dots, 1) \mathbf{T} = (0, 0, \dots, 1). \quad (23)$$

В блочной форме оно имеет вид

$$(\pi_0, \pi_1, \dots, \pi_p) \left[\begin{array}{cccc|c} A_{11} & A_{21} & \dots & A_{p1} & B \\ \hline B_{11} & B_{21} & & & \\ & & \ddots & & \\ & & & B_{p1} & 0 \end{array} \right] = (0, 0, \dots, 1). \quad (24)$$

Перемножение по блокам дает:

$$\pi_0 A_{i1} + \pi_i B_{i1} = 0, \quad i \in 1, \dots, p, \quad (25)$$

$$\pi_0 B = (0, 0, \dots, 1) = u_{m_0}. \quad (26)$$

Эти уравнения легко разрешимы:

$$\pi_0 = m_0\text{-я строка } B^{-1}, \quad (27)$$

$$\pi_i = -\pi_0 A_{i1} B_{i1}^{-1}, \quad i = 1, \dots, p. \quad (28)$$

Таким образом, если B^{-1} и B_{i1}^{-1} сохраняются в памяти либо в явной, либо в мультипликативной форме, то векторы π_0 и π_i легко вычисляются.

Определение столбца, вводимого в базис. Как и прежде, это может быть сделано путем вычисления величин:

$$\bar{c}_j = -\pi A^j, \quad j \text{ — небазисный}, \quad (29)$$

$$\bar{c}_j = \begin{cases} -\pi_0 A_i^{kj} - \pi_i B_i^{kj}, & \text{если } A^j \in S_i, i = 1, \dots, p, \\ -\pi_0 A_0^{kj}, & \text{если } A^j \in S_0, \end{cases} \quad (30)$$

где A_i^{kj} и B_i^{kj} — столбцы k_j матриц A_i и B_i соответственно.

Если

$$\min \bar{c}_j \geq 0, \quad (31)$$

то текущее решение оптимально. Если нет, то пусть

$$\min \bar{c}_j = \bar{c}_s < 0. \quad (32)$$

Тогда A^s входит в базис. Пусть $A^s \in S_\sigma$.

Вычисление $\bar{A}^s = B^{-1}A^s$. Для этой цели необходимо решить систему

$$B\bar{A}^s = A^s. \quad (33)$$

Пусть

$$\bar{A}^s = TZ^s. \quad (34)$$

Подставляя (34) в (33), получаем

$$(BT)Z^s = A^s, \quad (35)$$

что в блочной форме имеет такой вид:

$$\left[\begin{array}{cccc|c} A_{11} & A_{21} & \dots & A_{p1} & B \\ \hline B_{11} & & & & \\ & B_{21} & & & \\ & & \ddots & & \\ & & & & 0 \\ & & & & B_{p1} \end{array} \right] \begin{bmatrix} Z_1^s \\ \vdots \\ Z_p^s \\ Z^s \end{bmatrix} = \begin{bmatrix} Q^s \\ 0 \\ \vdots \\ R^s \\ 0 \end{bmatrix} \left. \vphantom{\begin{bmatrix} Z_1^s \\ \vdots \\ Z_p^s \\ Z^s \end{bmatrix}} \right\} \text{ блок } \sigma.$$

Записывая явно эти уравнения, получаем

$$\sum_{i=1}^p A_{i1}Z_i^s + BZ^s = Q^s, \quad (36)$$

$$B_{i1}Z_i^s = 0, \quad i = 1, \dots, p, \quad i \neq \sigma, \quad (37)$$

$$B_{\sigma 1}Z_\sigma^s = R^s. \quad (38)$$

Эти уравнения имеют следующее решение:

$$Z_i^s = 0, \quad i = 1, \dots, p, \quad i \neq \sigma, \quad (39)$$

$$Z_\sigma^s = B_{\sigma 1}^{-1}R^s, \quad (40)$$

$$Z^s = B^{-1}(Q^s - A_{\sigma 1}Z_\sigma^s). \quad (41)$$

Таким образом, Z^s и Z_σ^s могут быть вычислены, если $B_{\sigma 1}^{-1}$ и B^{-1} известны. После этого желаемый вектор $\bar{A}^s$ вычисляется

с помощью (34):

$$\begin{bmatrix} \bar{A}_1^s \\ \bar{A}_\sigma^s \\ \vdots \\ \bar{A}_p^s \\ \vdots \\ \bar{A}^s \end{bmatrix} = \begin{bmatrix} & \text{подматрица } V \\ & V_1 \\ & \vdots \\ I_1 & V_2 \\ & \vdots \\ & V_p \\ & \vdots \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} 0 \\ Z_\sigma^s \\ 0 \\ \vdots \\ Z^s \end{bmatrix} \left. \vphantom{\begin{bmatrix} \bar{A}_1^s \\ \bar{A}_\sigma^s \\ \vdots \\ \bar{A}_p^s \\ \vdots \\ \bar{A}^s \end{bmatrix}} \right\} \text{ блок } \sigma$$

$$\bar{A}_i^s = V_i Z^s, \quad i = 1, \dots, p, \quad i \neq \sigma, \quad (42)$$

$$\bar{A}_\sigma^s = Z_\sigma^s + V_\sigma Z^s, \quad (43)$$

$$\bar{A}^s = Z^s. \quad (44)$$

Тот факт, что каждый столбец V имеет по крайней мере одну ненулевую компоненту, может быть использован для того, чтобы записать (42)—(44) в более удобной форме. Напомним, что ненулевая часть столбца t матрицы V есть часть r_t , и она содержит вектор $-\bar{R}^t$, определенный в силу (15). Таким образом, столбец t матрицы V равен нулю, если r_t не равно i ; если же $r_t = i$, то он равен $-\bar{R}^t$. Как и ранее, введем в рассмотрение множества

$$R(i) = \{t/E \{1, \dots, m_0\}, r_t = i\}. \quad (45)$$

Тогда, обозначая через $(Z^s)_t$ компоненту t вектора Z^s , преобразуем (42), (43) к виду

$$\bar{A}_i^s = - \sum_{t \in R(i)} (Z^s)_t \bar{R}^t, \quad i = 1, \dots, p, \quad i \neq \sigma, \quad (46)$$

$$\bar{A}_\sigma^s = Z_\sigma^s - \sum_{t \in R(\sigma)} (Z^s)_t \bar{R}^t. \quad (47)$$

Заметим, что для задачи, рассмотренной в § 5.4, векторы $\bar{R}^t$ и Z_σ^s превращаются в скалярные единичные величины, и соотношения (44), (46) и (47) сводятся к уравнениям (5.4—36) и (5.4—37).

Выбор столбца, выводимого из базиса. Это делается согласно стандартным симплексным формулам, как указано в предшествующем параграфе. Если решение не является неограниченным, то столбец матрицы B , обозначаемый A^{j_r} , покидает базис. Предположим, что $A^{j_r} \in S_p$. Поскольку вычисление новых значений базисных переменных производится так же, как и в предшествующем параграфе, перейдем теперь к рассмотрению вопроса о пересчете матрицы B^{-1} , матриц

B_{ii}^{-1} и некоторых других величин, необходимых для осуществления последующих итераций.

Пересчет матрицы, обратной рабочему базису. Эффективная процедура пересчета B^{-1} может быть выведена путем анализа того, как изменяется большая базисная матрица B от цикла к циклу. Напомним, что столбец $A^{ir} \in S_p$ замещается столбцом $A^s \in S_\sigma$. Как и в предшествующем параграфе, A^s не всегда замещает A^{ir} непосредственно. В некоторых случаях сначала должна быть сделана замена A^{ir} на некоторый другой базисный столбец. Пусть *B есть новая базисная матрица, которая возникает после того, как либо сделаны эти замены, либо произведена непосредственная замена A^{ir} на A^s . Обратные матрицы к B и *B связаны уравнением

$${}^*B^{-1} = EB^{-1}, \quad (48)$$

где E — элементарная матрица либо матрица перестановок, чья точная структура вскоре будет построена. Пусть *T есть преобразующая матрица, которая приводит *B к блочно-треугольной форме. Обозначим

$$D = BT, \quad (49)$$

$${}^*D = {}^*B^*T. \quad (50)$$

Запишем *D в блочной форме:

$${}^*D = \left[\begin{array}{c|c} {}^*A_1 & {}^*B \\ \hline {}^*B_1 & 0 \end{array} \right], \quad (51)$$

где *B — новый рабочий базис. Обратная к *D матрица такова:

$${}^*D^{-1} = \left[\begin{array}{c|c} 0 & {}^*\hat{B}_1^{-1} \\ \hline {}^*B^{-1} & -{}^*B^{-1}{}^*A_1{}^*\hat{B}_1^{-1} \end{array} \right]. \quad (52)$$

В силу (50) и (48) имеем

$${}^*D^{-1} = {}^*T^{-1}{}^*B^{-1} = {}^*T^{-1}EB^{-1}. \quad (53)$$

Используя (49), получаем

$$B = DT^{-1}, \quad (54)$$

так что

$$B^{-1} = TD^{-1}. \quad (55)$$

Подставляя (55) в (53), получим

$${}^*D^{-1} = {}^*T^{-1}ETD^{-1}. \quad (56)$$

Обратная к *T матрица такова:

$${}^*T^{-1} = \left[\begin{array}{c|c} I_1 & {}^*V \\ \hline 0 & I_2 \end{array} \right]^{-1} = \left[\begin{array}{c|c} I_1 & -{}^*V \\ \hline 0 & I_2 \end{array} \right]. \quad (57)$$

Таким образом, в блочной форме уравнение (56) имеет вид

$$\left[\begin{array}{c|cc} 0 & & \\ \hline & \hat{B}_1^{-1} & \\ \hline *B^{-1} & -*B^{-1} & *A_1 \quad *\hat{B}_1^{-1} \end{array} \right] = \left[\begin{array}{c|c} I_1 & -*V \\ \hline 0 & I_2 \end{array} \right] \left[\begin{array}{c|c} E_1 & E_2 \\ \hline E_3 & E_4 \end{array} \right] \left[\begin{array}{c|c} I_1 & V \\ \hline 0 & I_2 \end{array} \right] \left[\begin{array}{c|cc} 0 & & \hat{B}_1^{-1} \\ \hline & B^{-1} & -B^{-1}A_1\hat{B}_1^{-1} \end{array} \right]. \quad (58)$$

Для того, чтобы теперь найти уравнение для $*B^{-1}$, вытекающее из (58), мы рассмотрим только первые m_0 столбцов последней матрицы и последние m_0 строк первой:

$$*B^{-1} = [0; I_2] \left[\begin{array}{c|c} E_1 & E_2 \\ \hline E_3 & E_4 \end{array} \right] \left[\begin{array}{c|c} I_1 & V \\ \hline 0 & I_2 \end{array} \right] \left[\begin{array}{c} 0 \\ B^{-1} \end{array} \right]. \quad (59)$$

Умножая первые две и последние две матрицы друг на друга, получаем

$$*B^{-1} = [E_3; E_4] \left[\begin{array}{c} VB^{-1} \\ B^{-1} \end{array} \right] \quad (60)$$

или

$$*B^{-1} = (E_4 + E_3V)B^{-1}. \quad (61)$$

Построенное уравнение (61) и задает основное преобразование для построения $*B^{-1}$. Далее мы выясним структуру $E_3V + E_4$ для различных типов изменения базиса, которые могут иметь место. Рассматриваемые подслучаи те же самые, что и в § 5.4, но требуемые операции несколько более сложные.

Случай 1. A^{lr} не является ключевым столбцом. В этом случае A^{lr} есть один из последних m_0 столбцов B , и блочно-диагональная структура подматрицы $\hat{B}_1$ не изменяется, если A^s заменяет его. Матрица E — обычная элементарная матрица, построенная в § 5.2. Пусть A^{lr} есть $(m+l)$ -й столбец B , где

$$m = \sum_{i=1}^p m_i, \quad (62)$$

и пусть $\bar{a}_{is}$ — i -я компонента $\bar{A}^s = B^{-1}A^s$. Тогда E отличается от единичной матрицы только в столбце $m+l$. Этот столбец является η -вектором с компонентами

$$\eta_i = \begin{cases} -\bar{a}_{is}/\bar{a}_{m+l,s}, & i = 1, \dots, m+m_0; i \neq m+l, \\ 1/\bar{a}_{m+l,s}, & i = m+l. \end{cases} \quad (63)$$

Таким образом, блоки матрицы E , участвующие в выражении

(59), имеют вид

$$E_3 = [0], \quad (64)$$

$$E_4 = \begin{bmatrix} 1 & & & & \eta_{m+1} \\ & \ddots & & & \vdots \\ & & 1 & & \eta_{m+l} \\ & & & \ddots & \vdots \\ & & & & 1 \end{bmatrix}. \quad (65)$$

Отсюда следует, что в этом случае основное уравнение (61) приобретает простую форму:

$$*B^{-1} = E_4 B^{-1}, \quad (66)$$

так что соотношение между новой и старой обратной матрицами очень просто и включает только элементарную столбцовую матрицу.

Для того чтобы завершить рассмотрение этого случая, необходимо показать, как вычисляются все другие величины, требуемые в следующем цикле. Единственное, что требуется проделать, это внести изменения в l -й столбец матрицы V , который в силу (46), (47) содержит вектор $-\bar{R}^l$. Ранее в него входил вектор с ненулевыми элементами из блока ρ . Теперь в него следует ввести ненулевые элементы из блока σ , равные $-B_{\sigma 1}^{-1}$, умноженной на блок σ вектора A^s . Никакой дополнительной работы не требуется, так как этот вектор уже был вычислен в виде Z^s в (40). Запись элементов этого вектора и индекса σ как l -го столбца V завершает вычисления в этом случае.

Случай 2. A^{lr} есть ключевой столбец.

Подслучай а): A^{lr} и A^s принадлежат различным множествам, т. е. $\rho \neq \sigma$. В этом случае прямая замена A^{lr} на A^s изменила бы блочно-диагональную структуру $\hat{B}_1$, которая существенна для реализации метода. Однако теорема 1 приложима к базисной матрице и после вывода A^{lr} , так что один из последних m_0 столбцов B должен быть таким, что когда вектор в его ρ -м строчном блоке замещает соответствующий подвектор A^{lr} в $B_{\rho 1}$, новая подматрица $B_{\rho 1}$ оказывается неособой.

Как и в предшествующем параграфе, один из таких столбцов обменивается местами с A^{lr} , делая A^{lr} неключевым столбцом. В этом случае применима процедура случая 1.

Пусть A^{lr} — столбец i_1 матрицы B и пусть он заменяется столбцом $m + i_2$. Преобразования, соответствующие такой замене столб-

цов, формально определяются как

$${}^*B = BE, \quad (67)$$

где E — единичная матрица, столбцы i_1 и $m+i_2$ которой обменялись местами, т. е.

$$E = \left[\begin{array}{ccccccc} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & 0 & \dots & 1 \\ & & & & \vdots & \ddots & \vdots \\ & & & & 1 & & \vdots \\ & & & & \vdots & & \ddots \\ & & & & & & 1 \\ & & & & & & & \vdots \\ & & & & & & & 1 \\ & & & & & & & & \vdots \\ & & & & & & & & 0 \\ & & & & & & & & \vdots \\ & & & & & & & & 1 \end{array} \right] \quad \begin{array}{l} \leftarrow \text{строка } i_1, \\ \\ \\ \\ \leftarrow \text{строка } m+i_2. \end{array}$$

↑ столбец i_1 ↑ столбец $m+i_2$

(68)

Поскольку проведение дважды такого преобразования соответствует единичной матрице, то

$$E^{-1} = E, \quad (69)$$

поэтому в силу (67)

$${}^*B^{-1} = E^{-1}B^{-1} \quad (70)$$

и, используя (69), получаем

$${}^*B^{-1} = EB^{-1}. \quad (71)$$

В силу (68) убеждаемся, что нижние блоки матрицы E таковы:

$$E_3 = \left[\begin{array}{ccc} \dots & 1 & \dots \end{array} \right] \leftarrow \text{строка } i_2, \quad (72)$$

↑ столбец i_1

$$E_4 = \left[\begin{array}{ccccccc} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & 0 & \dots & & \\ & & & \vdots & \ddots & & \\ & & & 1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & \end{array} \right] \leftarrow \text{строка } i_2, \quad (73)$$

↑ столбец i_1

Если v есть i_1 -я строка матрицы V , то E_3V есть квадратная матрица размерности $m_0 \times m_0$ со строкой i_2 , равной v , и всеми

другими равными нулю. Следовательно,

$${}^*B^{-1} = (E_4 + E_3 V) B^{-1} = \left[\begin{array}{ccccccc} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ \hline & & & & v & \text{(строка } i_2) & \\ \hline & & & & & 1 & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{array} \right] B^{-1}. \quad (74)$$

Таким образом, старая и новая обратные матрицы связаны элементарной строчной матрицей.

В приведенных выше выражениях ${}^*B^{-1}$ — неособенная матрица, если и только если $E_4 + E_3 V$ неособенна, что в свою очередь справедливо, если и только если компонента i_2 вектора v не равна нулю. То, что такой выбор i_2 всегда осуществим, можно видеть из следующего. Строка i_1 матрицы V , v взята из ρ -го строчного блока V . Единственные ненулевые столбцы в этом блоке имеют вид $B_{\rho 1}^{-1}(\cdot)$, т. е. $B_{\rho 1}^{-1}$, умноженная на некоторый другой вектор. Для конкретности, пусть

$$i_1 = \sum_{j=1}^{\rho-1} m_j + 3, \quad (75)$$

так что столбец 3 матрицы $B_{\rho 1}$ выводится из $B_{\rho 1}$, и v — строка 3 из ρ -го строчного блока. Если все вектора вида $B_{\rho 1}^{-1}(\cdot)$ имеют нули в положении 3, то невозможно заместить столбец 3 в $B_{\rho 1}$ и сохранить $B_{\rho 1}$ неособой. Но это невозможно в силу теоремы 1. Таким образом, найдется некоторый столбец i_2 матрицы V , в котором компонента в строке i_1 отлична от нуля и этот столбец может заместить A^{j_r} .

Для того чтобы завершить рассмотрение этого подслучая, заметим, что замена столбцов вызывает изменения в матрицах $B_{\rho 1}$ и $A_{\rho 1}$, так что $B_{\rho 1}^{-1}$ должна быть вычислена заново с помощью ведущей операции. Столбцы V , содержащие векторы вида $B_{\rho 1}^{-1}(\cdot)$, также изменяются, и они должны быть заново вычислены с помощью той же самой операции, использующей новое значение $B_{\rho 1}^{-1}$.

Подслучай б): $\rho = \sigma$. Вновь пусть A^{j_r} — столбец с номером i_1 матрицы B . Рассмотрим строку i_1 матрицы V , v и выясним, содержит ли она ненулевой элемент. Если это так, и элемент находится в столбце i_2 , заменим столбец i_1 матрицы B на столбец $m + i_2$ так, как это описано в подслучае а). Затем перейдем к операциям случая 1. Если $v = 0$, заменим A^{j_r} через A^s , непосредственно осуществляя ведущую операцию. Здесь E — элементар-

ная матрица со столбцом i_1 в качестве η -вектора, так же как и в (63). Таким образом, E_3 и E_4 имеют вид

$$E_3 = \begin{bmatrix} & \eta_{m+1} & \\ 0 & \vdots & 0 \\ & \eta_{m+m_0} & \end{bmatrix} \quad (76)$$

↑
столбец i_1

$$E_4 = I, \quad (77)$$

поэтому

$$E_4 + E_3 V = I + \begin{bmatrix} \eta_{m+1}^v \\ \vdots \\ \eta_{m+m_0}^v \end{bmatrix}. \quad (78)$$

Поскольку $v=0$, имеем

$$E_4 + E_3 V = I. \quad (79)$$

В силу (61) получаем, что в этом случае нет никаких изменений в рабочем базисе

$$^*B^{-1} = IB^{-1} = B^{-1}. \quad (80)$$

Для того чтобы завершить рассмотрение этого последнего подслучая заметим, что вычисление $B_{\rho 1}^{-1}$, $A_{\rho 1}$ и векторов в V вида $B_{\rho 1}^{-1}(\cdot)$ осуществляется так же, как в подслучае а). Вновь требуется только одна ведущая операция.

Подводя итоги, отметим, что каждый случай сводится к преобразованию B^{-1} путем умножения на элементарную строчную или столбцовую матрицу и либо переходу к случаю 1, либо к завершению этапа. Следовательно, на любой итерации k текущая обратная матрица представляет собой произведение элементарных строчных и столбцовых матриц, каждая размерности m_0 . Обновление обратных матриц B_{i1}^{-1} требует по крайней мере одной ведущей операции на цикле. Следовательно, вычислительная процедура в целом требует приблизительно такого же труда, как и при решении двух меньших задач ЛП, одной, имеющей матрицу ограничений с m_0 строками, и другой, имеющей матрицу с числом строк, равным размеру некоторой матрицы B_{i1} .

Задачи

1. Вывести формулы (5.3-30) — (5.3-32).
2. Решить задачу

$$\begin{aligned} & \min z, \\ & x_5 \quad - 1/4 x_2 + 3x_3 - 3/4 x_4 = 5, \\ & x_1 \quad - 3/4 x_2 + 2x_3 - 1/4 x_4 = 3, \\ & \quad \quad -z + 8x_2 - 24x_3 + 5x_4 = -28, \end{aligned}$$

$$0 \leq x_1 \leq 4, \quad 0 \leq x_2 \leq 1/2, \quad 0 \leq x_3 \leq 1, \quad x_4 \geq 0, \quad x_5 \geq 0.$$

3. Вывести формулы, сходные с (5.4-36)—(5.4-37), для новых значений базисных переменных через компоненты вектора $\bar{d}=B^{-1}d$.

4. Показать детально, какие изменения требуются в алгоритме § 5.4, когда последние p уравнений содержат как положительные, так и отрицательные коэффициенты.

5. Показать детально, как алгоритм § 5.4 мог бы быть применен к фазе I симплекс-метода.

6. Решить пример из § 2.5, используя алгоритм § 5.5. Сравнить результаты на каждой итерации с полученными в § 2.5.

7. Рассмотреть обобщение алгоритма § 5.5 на блочно-диагональные задачи, содержащие связывающие строки и связывающие столбцы. Какие важные свойства, использованные в § 5.5, в этом случае уже не будут иметь место? Предложить алгоритм, который обходит эти трудности и использует только рабочий базис и обратные матрицы к каждому блоку.

8. Показать, как алгоритм § 5.5 приводится к алгоритму § 5.4 и алгоритму § 5.3, когда решаемая задача приобретает частный вид, рассмотренный в этих параграфах.

ЛИТЕРАТУРА

1. G. B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, N. J., 1963, chap. 18. [Русский перевод: Дж. Данциг, Линейное программирование, его обобщения и применения, «Прогресс», М., 1966.]
2. G. B. Dantzig and R. M. Van Slyke, Generalized Upper Bounding Techniques, J. Computer System Sci., 1, 1967, pp. 213—226.
3. R. N. Kaul, An Extension of Generalized Upper Bounding Techniques for Linear Programming, Rept. ORC 65-27, Operations Research Center, University of California at Berkeley, 1965.
4. P. Wolfe and L. Cutler, Experiments in Linear Programming, in «Recent Advances in Mathematical Programming»; R. L. Graves and P. Wolfe, eds., McGraw-Hill, Inc., New York, 1963, pp. 177—200.
5. D. Smith and W. Orchard-Hays, Computational Efficiency in Product Form LP Codes, in «Recent Advances in Mathematical Programming»; R. L. Graves and P. Wolfe, eds., McGraw-Hill, Inc., New York, 1963, pp. 211—218.
6. Mathematical Programming System/360 (360A—CO—14X), Linear and Separable Programming — Users Manual, IBM Rept. H20—0476—0.
7. L. J. Larsen, A Modified Inversion Procedure for Product Form of the Inverse Linear Programming Codes. Commun. ACM, July 1962.

ГЛАВА VI

ПРОЦЕДУРЫ РАСЧЛЕНЕНИЯ В НЕЛИНЕЙНОМ ПРОГРАММИРОВАНИИ

§ 6.1. ВВЕДЕНИЕ

Так же как в линейном программировании, термин «процедура расчленения» используется здесь для обозначения алгоритма, в котором переменные задачи разделяются на два подмножества, причем сначала изменяются переменные в первом подмножестве, а потом в другом. Конечно, каждое из подмножеств может быть расчленено и далее, как, например, в случае, когда одно из них содержит переменные, которые входят линейно, а соответствующая матрица ограничений является блочно-диагональной. Используемое расчленение для задач, рассматриваемых далее, очевидным образом определяется структурой проблемы.

Ниже рассматриваются два связанных класса задач математического программирования. Первый содержит задачи вида

$$\min \sum_{i=1}^p c_i'(y) x_i + c_0(y) \quad (1)$$

при ограничениях

$$A_i(y)x_i \geq b_i(y), \quad i = 1, \dots, p. \quad (2)$$

Здесь матрицы A_i , коэффициенты целевой функции c_i и векторы правых частей b_i — все являются непрерывными функциями вектора связывающих переменных, y . Если y фиксировано, то задача становится линейной относительно x_i , так что естественно настраивать переменные y и x_i отдельно. Проблемы этого типа представляют собой нелинейное обобщение блочных задач ЛП. Розен распространил свою процедуру расчленения для линейных задач, описанную в § 4.4, на случай задач типа (1) — (2). Это обобщение рассматривается в первой части данной главы.

Второй класс задач, рассматриваемых далее, описывается соотношениями

$$\min [c'x + f(y)] \quad (3)$$

при ограничениях

$$Ax + F(y) \geq b, \quad (4)$$

$$x \geq 0, \quad y \in S, \quad (5)$$

где f и F — скалярная и векторнозначная функции y соответственно, а S — произвольное подмножество E^n . Свобода в доопределении S позволяет включать в этот класс некоторые важные новые задачи.

Наиболее интересен случай, когда S состоит из множества всех векторов в E^n с целочисленными компонентами. Если в дополнение предположить F и f линейными, то (3) — (5) будет представлять собой хорошо известную задачу ЛП частично-целочисленного типа. Если A блочно-диагональна и $S = E^n$, то (3) — (5) есть специальный случай (1) — (2), где A_i и c_i не являются функциями y .

Алгоритм расчленения, предложенный Бендерсом и приложимый к задачам типа (3) — (5), обсуждается во второй части этой главы. Так же как в методе Розена, сначала варьируется y , затем решается линейная задача для получения новых значений x и т. д. Однако на этом сходство заканчивается. Алгоритм Бендерса настраивает переменные y с помощью решения задачи математического программирования на множестве S , причем на каждом цикле добавляется одно или более ограничений. Эти ограничения конструируются с помощью решения задачи, двойственной к (3) — (5) для фиксированного y . Когда (3) — (5) является частично-целочисленной задачей ЛП, задача, в ходе которой происходит изменение переменных y , является чисто целочисленной задачей ЛП. Таким образом, в этом случае (3) — (5) решается путем раздельного изменения дискретных и непрерывных переменных. Это особенно удобно в случае, когда A имеет специальную структуру, например, является блочно-диагональной или является матрицей транспортного типа.

Как и для задач ЛП, оба эти алгоритма укладываются в единую схему двойственной концепции релаксации ограничений и закрепления переменных. В процедуре Бендерса задача, в которой определяются переменные y , имеет много ограничений, так что к ней приложима релаксация. Так как эта задача может не быть выпуклой, ограничения, вообще говоря, могут только добавляться, но не могут опускаться. Метод Розена использует закрепление, когда подмножество линейных неравенств с неособенной коэффициентной матрицей учитывается в виде равенств. Это подмножество записывается, а затем используется для выражения переменных x_i через y , что позволяет сконструировать координирующую задачу относительно переменных y .

§ 6.2. АЛГОРИТМ РАСЧЛЕНЕНИЯ РОЗЕНА ДЛЯ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Алгоритм расчленения Розена для линейного программирования, описанный в § 4.4, был распространен на определенный тип нелинейных задач [1]. Процедура приложима к задачам, имеющим вид

$$\min \left\{ \sum_{i=1}^p c'_i(y) x_i + c_0(y) \right\} \quad (1)$$

при ограничениях

$$A_i(y) x_i \geq b_i(y), \quad i = 1, \dots, p, \quad p \geq 1. \quad (2)$$

Здесь переменные задачи естественно разделяются на два подмножества: связывающие переменные y и «линейные» переменные x_i . Замечательной особенностью этой структуры является то, что для фиксированных y задача разделяется на p независимых линейных подзадач относительно переменных x_i . Следовательно, естественна итеративная процедура, в которой сначала изменяются y , а затем x_i . Как и в процедуре § 4.4 для линейных задач, базисные матрицы и двойственные оценки, полученные при решении подзадач, используются для формирования координирующей задачи относительно переменных y . В отличие же от линейных алгоритмов, координирующая задача имеет нелинейную целевую функцию.

Математическое доказательство сходимости к глобальному минимуму имеет место, только когда в (1)–(2) коэффициенты c_i и матрицы A_i не зависят от y , так что проблема приобретает вид

$$\min \sum_{i=1}^p c'_i x_i, \quad (3)$$

$$A_i x_i \geq b_i(y), \quad i = 1, \dots, p. \quad (4)$$

Первоначально рассматривается этот тип задач, для которого свойства алгоритма будут полностью установлены и обоснованы, а затем кратко рассмотрена более общая проблема (1)–(2). Делаются следующие дополнительные предположения:

Предположение 1. Для $i=1, \dots, p$ каждая компонента $b_i(y)$ является дифференцируемой выпуклой функцией y . Из этого следует, что (3)–(4) является задачей выпуклого программирования, для которой любой локальный минимум является глобальным.

Предположение 2. Существуют допустимые векторы (x_i, y) , внутренние ко всем нелинейным ограничениям, т. е. такие, что $A_i x_i \geq b_i(y)$, где выполняется строгое неравенство для

нелинейных компонент $b_i(y)$. Это обеспечивает существование седловой точки функции Лагранжа (см. [2], [3]). Совместно с предположением 1, предположение 2 обеспечивает, что условия Куна — Таккера являются необходимыми и достаточными для оптимального решения.

Предположение 3. Для $i = 1, \dots, p$ ни одна из строк A_i не является нулевым вектором. Если исходно сформулированная задача имеет нулевые строчки, т. е. присутствуют ограничения вида $b_{ij}(y) \leq 0$, то вводятся искусственные переменные x_{ij}^a с большим положительным коэффициентом в целевой функции, так что ограничение преобразуется к виду $b_{ij}(y) \leq x_{ij}^a$. Если коэффициенты достаточно велики, $x_{ij}^a = 0$ в оптимальном решении.

Предположение 4. Допустимый вектор y_0 существует и известен, т. е. известно такое y_0 , при котором существуют x_i , удовлетворяющие (4).

6.2.1. Изложение алгоритма

Задача (3) — (4) может быть решена путем минимизации по x_i при всех фиксированных y , а затем путем минимизации результата по y . Таким образом, можно записать так:

$$\min_{y \in S} \left\{ \sum_{i=1}^p \min \{c_i' x_i / A_i x_i \geq b_i(y)\} \right\}, \quad (1)$$

где S — множество значений y , для которых каждая из внутренних минимизаций допустима. Если

$$S_i = \{y / \exists x_i : A_i x_i \geq b_i(y)\}, \quad (2)$$

тогда

$$S = \bigcap_{i=1}^p S_i. \quad (3)$$

Векторы $y \in S$ будут называться *допустимыми*. Определим

$$\psi_i(y) = \min_{x_i} \{c_i' x_i / A_i x_i \geq b_i(y)\}. \quad (4)$$

Тогда задача (1) записывается в виде

$$\min \psi(y) = \sum_{i=1}^p \psi_i(y) \quad (5)$$

при ограничениях

$$y \in S. \quad (5)$$

Следующая теорема устанавливает, что при предположении 1 задача (5) — (6) есть задача выпуклого программирования.

Теорема 1. Если каждая функция $b_i(y)$ выпукла, то множество S выпукло и каждая функция ψ_i выпукла на S .

Доказательство предлагается в качестве упражнения 1. В силу теоремы 1 исходная задача приводится к решению задачи выпуклого программирования относительно y . Однако эта задача затрудняется тем обстоятельством, что для оценки любой из функций $\psi_i(y)$ требуется решить линейную подзадачу вида

$$\left. \begin{aligned} \min c_i' x_i, \\ A_i x_i \geq b_i(y). \end{aligned} \right\} \begin{array}{l} \text{подзадача} \\ i \end{array} \quad \begin{array}{l} (7) \\ (8) \end{array}$$

Напомним также, что ограничения на y также не заданы явно через некоторые функции. Для того чтобы преодолеть эти трудности, мы попытаемся извлечь как можно больше информации из решения линейных подзадач (7) — (8). Эта информация будет определять функции ψ_i и ограничения S в окрестности текущего значения вектора y .

Пусть y_0 — допустимый вектор. Предположим, что каждая из линейных задач (7) — (8) разрешима для y , равного y_0 . С помощью этого решения можно расчленить матрицы A_i , которые имеют больше строк, чем столбцов, на квадратные неособенные базисные матрицы A_{i1} и небазисные матрицы A_{i2} . Матрицы A_{i1} определяют векторы решения x_i^0 с помощью уравнений

$$A_{i1} x_i^0 = b_{i1}(y_0). \quad (9)$$

Если мы (времененно) предположим, что соотношение (9) определяет связь между x_i и y , то подзадачи (7) — (8) могут быть записаны в виде

$$\min c_i' x_i, \quad (10)$$

$$A_{i1} x_i = b_{i1}(y), \quad (11)$$

$$A_{i2} x_i \geq b_{i2}(y). \quad (12)$$

Тогда (11) может быть разрешено относительно x_i :

$$x_i = A_{i1}^{-1} b_{i1}(y). \quad (13)$$

Уравнение (13) используем для исключения x_i из (10) и (12), что дает

$$c_i' x_i = c_i' A_{i1}^{-1} b_{i1}(y) = (u_{i1}^0)' b_{i1}(y), \quad (14)$$

$$A_{i2} A_{i1}^{-1} b_{i1}(y) \geq b_{i2}(y), \quad i = 1, \dots, p, \quad (15)$$

где

$$(u_{i1}^0)' = c_i' A_{i1}^{-1} \geq 0 \quad (16)$$

— вектор двойственных переменных, ассоциированных с оптимальным базисом A_{i1} . В предположении, что связь (13) сохра-

няет свою силу, определим функции $\psi_i(y)$ соотношением (14); при этом ограничения (15) определяют множество S_i в (2). Таким образом, координирующая проблема в пространстве y имеет целевую функцию

$$f(y) = \sum_{i=1}^p (u_{i1}^0)' b_{i1}(y) \quad (17)$$

и ограничения (15). Проведенную операцию можно рассматривать как приложение процедуры закрепления, описанной в § 4.2, где некоторые ограничения исходной задачи рассматриваются как равенства. Целевая функция (17) выпукла, поскольку b_{i1} выпуклы и $u_{i1}^0 \geq 0$. Однако ограничения (15) нелинейны и могут не определять выпуклое множество, так как элементы $A_{i2}A_{i1}^{-1}$ могут иметь произвольный знак. Решение задач с нелинейными ограничениями значительно более сложно, чем решение задач с линейными ограничениями, и отсутствие гарантированной выпуклости означает, что не может быть гарантировано глобально-оптимальное решение. Однако любая координирующая программа служит только для того, чтобы провести улучшение допустимого вектора y , если таковое улучшение существует. Имея это в виду, мы линеаризуем ограничения (15) относительно текущей точки y_0 . Свойства полученной координирующей задачи определяются доказанной ниже теоремой 3. Пусть

$$Q_i = A_{i2}A_{i1}^{-1}. \quad (18)$$

Введем в рассмотрение якобиан $J_{ij}(y)$ ($j=1, 2$), где $J_{ij}(y)$ есть матрица, строки которой являются градиентами компонент b_{ij} , вычисленными в точке y . Тогда линеаризация (15) в точке y_0 дает

$$[Q_i J_{i1}(y_0) - J_{i2}(y_0)](y - y_0) \geq b_{i2}(y_0) - Q_i b_{i1}(y_0). \quad (19)$$

Координирующая задача имеет выпуклую целевую функцию (17) и линейные ограничения (19):

$$\min f(y) = \sum_{i=1}^p (u_{i1}^0)' b_{i1}(y), \quad (20)$$

$$[Q_i J_{i1}(y_0) - J_{i2}(y_0)](y - y_0) \geq b_{i2}(y_0) - Q_i b_{i1}(y_0), \quad i = 1, \dots, p. \quad (21)$$

Пусть y^1 — решение этой задачи. Так как y_0 допустимо для нее, то

$$f(y^1) \leq f(y_0). \quad (22)$$

Конечно, возможно, что в (22) имеет место равенство. В этом

случае текущее решение $(\{x_i^0\}, y_0)$ может быть проверено на оптимальность.

Теорема 2 (критерий оптимальности). Пусть $x_i^0, i=1, \dots, p$ — решение линейной задачи (7) — (8) для $y = y_0$. Тогда необходимые и достаточные условия того, чтобы $(\{x_i^0\}, y_0)$ были решением исходной задачи (6.2-3) — (6.2-4), заключаются в том, что y_0 — решение координирующей задачи и выполнены условия

$$u_{i1}^* = u_{i1}^0 - Q_i u_{i2}^1 \geq 0, \quad i = 1, \dots, p. \quad (23)$$

В (23) u_{i2}^1 есть вектор двойственных переменных, соответствующих i -му множеству ограничений (21) в оптимуме координирующей задачи.

Доказательство. Достаточность. При выполнении предположений 1—4 данное решение оптимально для исходной задачи, если и только если удовлетворяются условия Куна — Таккера. Пусть эта задача записана в расчлененном виде

$$\min \sum_{i=1}^p c_i' x_i \quad (24)$$

при ограничениях

$$\begin{cases} A_{i1} x_i \geq b_{i1}(y), \\ \end{cases} \quad i = 1, \dots, p. \quad (25)$$

$$\begin{cases} A_{i2} x_i \geq b_{i2}(y), \\ \end{cases} \quad (26)$$

Пусть u_{i1}, u_{i2} — двойственные переменные, соответствующие ограничениям (25), (26) соответственно. Запишем функцию Лагранжа

$$L = \sum_{i=1}^p (c_i' x_i + u_{i1}' (b_{i1}(y) - A_{i1} x_i) + u_{i2}' (b_{i2}(y) - A_{i2} x_i)). \quad (27)$$

Условия Куна — Таккера в точке $(\{x_i^0\}, y_0)$ таковы для $i = 1, \dots, p$:

$$A_{i1}' u_{i1} + A_{i2}' u_{i2} = c_i, \quad (28)$$

$$J_{i1}'(y_0) u_{i1} + J_{i2}'(y_0) u_{i2} = 0, \quad (29)$$

$$u_{i1}' (b_{i1}(y_0) - A_{i1} x_i^0) = u_{i2}' (b_{i2}(y_0) - A_{i2} x_i^0) = 0, \quad (30)$$

$$A_{i1} x_i^0 \geq b_{i1}(y_0), \quad A_{i2} x_i^0 \geq b_{i2}(y_0), \quad (31)$$

$$u_{i2} \geq 0, \quad (32)$$

$$u_{i1} \geq 0. \quad (33)$$

Условия оптимальности для линейных подзадач таковы:

$$u_{i1}^0 = (A_{i1}^{-1})' c_i \geq 0, \quad (34)$$

$$A_{i1} x_i^0 = b_{i1}(y_0), \quad A_{i2} x_i^0 \geq b_{i2}(y_0). \quad (35)$$

Условия Куна — Таккера для координирующей задачи в точке $y = y_0$ находятся из лагранжиана

$$\hat{L} = \sum_{i=1}^p \{ (u_{i1}^0)' b_{i1}(y) + (u_{i2}^1)' [b_{i2}(y_0) - Q_i b_{i1}(y_0) - (Q_i J_{i1}(y_0) - J_{i2}(y_0))(y - y_0)] \} \quad (36)$$

и имеют для $i = 1, \dots, p$ вид

$$J'_{i1}(y_0) [u_{i1}^0 - Q'_i u_{i2}^1] + J'_{i2}(y_0) u_{i2}^1 = 0, \quad (37)$$

$$(u_{i2}^1)' (b_{i2}(y_0) - Q_i b_{i1}(y_0)) = 0, \quad (38)$$

$$u_{i2}^1 \geq 0. \quad (39)$$

Сравнивая (28) — (33) и (34) — (35), (37) — (39), находим, что могут не удовлетворяться только условия (28), (29) и (33). Сравнение (37) и (29) предполагает, что мы приняли

$$u_{i1} = u_{i1}^* = u_{i1}^0 - Q'_i u_{i2}^1, \quad (40)$$

следовательно, (29) удовлетворяется. Умножая (40) на A'_{i1} , получаем

$$A'_{i1} u_{i1}^* + A'_{i1} Q'_i u_{i2}^1 = A'_{i1} u_{i1}^0, \quad (41)$$

так как, в силу (18), $Q'_i = (A_{i1}^{-1})' A'_{i1}$ и $u_{i1}^0 = (A_{i1}^{-1})' c_i$, (41) переходит в (28). Таким образом, если

$$u_{i1}^* \geq 0, \quad i = 1, \dots, p, \quad (42)$$

все множество условий Куна — Таккера удовлетворяется в $(\{x_i^0\}, y_0)$ причем u_{i1}^* и u_{i2}^1 являются оптимальными значениями множителей Лагранжа. Следовательно, это решение оптимально.

Необходимость доказывается сходным образом. Детали оставляем в качестве упражнения 2.

Следствие. Достаточное условие того, что $(\{x_i^0\}, y_0)$ является оптимальным, состоит в том, что y_0 решает координирующую задачу, причем ни одно из линейных ограничений не выполняется как равенство.

Доказательство. Если y_0 есть внутренний минимум, тогда, в силу (38), $u_{i2}^1 = 0$ и, в силу (40), $u_{i1}^* = u_{i1}^0$. Так как $u_{i1}^0 \geq 0$, то $(\{x_i^0\}, y_0)$ оптимально.

Если критерий оптимальности нарушается, то либо $f(y^1) < f(y_0)$ либо $f(y^1) = f(y_0)$ и некоторое $u_{i1}^* \not\geq 0$. Рассмотрим сначала случай $f(y^1) < f(y_0)$. Так как координирующая задача была образована в предположении $A_{ii} x_i = b_{ii}(y)$ и линеаризации ограничений (15), то целевая функция $f(y)$ координирующей задачи равна $\psi(y)$ только для таких y , что A_{ii} является оптимальным

базисом линейной подзадачи (7) — (8). В действительности мы желаем уменьшать ψ , а не f . Далее, благодаря линеаризации, y^1 может не удовлетворять (15) и, значит, может не быть в S . Следующая теорема касается этих трудностей. Она показывает что либо y^1 принадлежит S и удовлетворяет $\psi(y^1) < \psi(y_0)$, либо, если может быть сделан ненулевой шаг θ_m вдоль линии от y_0 к y^1 , то может быть построен новый вектор $y_0 + \theta_m(y^1 - y_0)$, который обладает этим свойством.

Теорема 3 (обоснование линеаризации ограничений). Пусть оптимальное решение координирующей задачи удовлетворяет условию

$$f(y^1) < f(y_0). \quad (43)$$

Если y^1 удовлетворяет

$$h_i(y^1) = Qb_{i1}(y^1) - b_{i2}(y^1) \geq 0, \quad i = 1, \dots, p, \quad (44)$$

то

$$\psi(y^1) \leq f(y^1) < f(y_0) = \psi(y_0). \quad (45)$$

Если y^1 не удовлетворяет (44), введем θ_m как наибольший шаг от y_0 к y^1 , при котором сохраняется допустимость:

$$\theta_m = \max \{ \theta / h_i(y_0 + \theta(y^1 - y_0)) \geq 0, \quad i = 1, \dots, p, 0 \leq \theta \leq 1 \}. \quad (46)$$

Тогда, если $\theta_m > 0$, то

$$\psi(y_0 + \theta_m(y^1 - y_0)) < \psi(y_0). \quad (47)$$

Доказательство. Неравенства (44) являются, в силу (11) — (15), просто необходимыми и достаточными условиями того, что базис A_{ii} является допустимым для i -й линейной подзадачи при $y = y^1$. Так как A_{ii} может не быть оптимальным базисом, когда $y = y^1$, то

$$\psi(y^1) \leq f(y^1). \quad (48)$$

Поскольку A_{ii} является оптимальным базисом при $y = y_0$, то

$$f(y_0) = \psi(y_0) \quad (49)$$

и, комбинируя (48), (49) и (43), получаем доказательство (45).

Для того чтобы доказать (47), положим

$$\hat{y} = y_0 + \theta_m(y^1 - y_0). \quad (50)$$

Поскольку f выпукла, то

$$f(\hat{y}) \leq f(y_0) + \theta_m(f(y^1) - f(y_0)). \quad (51)$$

Базисные матрицы A_{ii} могут не быть оптимальными в линейных подзадачах (7) — (8), когда $y = \hat{y}$, так что

$$\psi(\hat{y}) \leq f(\hat{y}). \quad (52)$$

Таким образом, используя (49), получаем

$$\psi(\hat{y}) \leq \psi(y_0) + \theta_m (f(y^1) - f(y_0)). \quad (53)$$

По условиям теоремы $f(y^1) < f(y_0)$ и $\theta_m > 0$, так что

$$\psi(\hat{y}) < \psi(y_0), \quad (54)$$

а это и есть (47).

Не исследована еще ситуация, когда $f(y^1) = f(y_0)$, и некоторая $u_{i1}^* \not\geq 0$. Здесь результат очень сходен с тем, который имел место в линейном случае § 4.4. Существует альтернативный оптимальный базис для i -й линейной подзадачи, в котором по крайней мере одна из строк A_{i1} , соответствующая отрицательной компоненте u_{i1}^* , отсутствует.

Теорема 4 (альтернативный оптимальный базис). *Если $f(y^1) = f(y^0)$ и для некоторого i , u_{i1}^* имеет некоторые отрицательные компоненты, существует альтернативное оптимальное решение для i -й линейной подзадачи ($y = y_0$) такое, что по крайней мере одна строка исходного базиса A_{i1} , соответствующая отрицательной компоненте u_{i1}^* , отсутствует.*

Доказательство. При $y = y_0$ базис A_{i1} оптимален для линейной подзадачи (7) — (8), если и только если он является оптимальным для двойственной к ней, которая записывается в виде

$$\begin{aligned} \max [b'_{i1}(y_0)u_{i1} + b'_{i2}(y_0)u_{i2}], \\ A'_{i1}u_{i1} + A'_{i2}u_{i2} = c_i, \quad u_{i1} \geq 0, u_{i2} \geq 0. \end{aligned} \quad (55)$$

Текущий базис A'_{i1} допустим, т. е. $u_{i1}^0 = (A'_{i1})^{-1}c_i \geq 0$, и оптимален, т. е. характеристические разности для u_{i2} удовлетворяют условиям

$$b_{i2}(y_0) - A_{i2}x_i^0 \leq 0. \quad (56)$$

Записанное условие есть, конечно, просто требование того, что (x_i^0, y_0) удовлетворяет нижней части ограничений i -го линейного блока (12). Вектор u_{i1}^* дается формулой

$$u_{i1}^* = u_{i1}^0 - (A_{i1}^{-1})' A'_{i2} u_{i2}^1. \quad (57)$$

Так как u_{i1}^* имеет некоторые отрицательные компоненты и $u_{i1}^0 \geq 0$, некоторые компоненты u_{i2}^1 должны быть положительны. В силу условий дополняющей нежесткости, ограничения координирующей задачи, соответствующие этим компонентам, должны быть связывающими при $y = y_0$, т. е. некоторые из характеристических разностей (56) должны быть равны нулю. Если один или более небазисных двойственных столбцов (т. е. исходных строк) с ну-

левыми характеристическими разностями могут заместить некоторые из текущих базисных столбцов, приводя к новому допустимому базису, то этот новый базис является оптимальным. То, что такая замена возможна, было уже показано в теореме 4.4—2, так что теорема доказана.

Таким образом, алгоритм состоит в следующем:

1. Выберем вектор $y_0 \in S$.

2. Решим линейные подзадачи (7) — (8) при $y = y_0$, получая оптимальные решения x_i^0 , базисные матрицы A_{i1} и двойственные переменные u_{i1}^0 .

3. Используя величины, найденные в шаге 2, построим координирующую задачу (20) — (21) и решим ее, получая оптимальное решение y^1 .

4. Если $f(y^1) = f(y^0)$ и все u_{i1}^* в (23) являются неотрицательными, решение $(\{x_i^0\}, y_0)$ является оптимальным.

5. Если $f(y^1) = f(y^0)$ и $u_{i1}^* \not\geq 0$ для $i \in I$, то в силу теоремы 4 существует альтернативный оптимальный базис для i -й линейной подзадачи, в котором по крайней мере одна строка A_{i1} , соответствующая отрицательной компоненте u_{i1}^* , отсутствует. Такой базис может быть найден путем решения линейной задачи (32) § 4.4. Проведем по меньшей мере одно изменение базиса и перейдем к шагу 3 с новой базисной матрицей, двойственными переменными и т. д.

6. Если $f(y^1) < f(y_0)$ и все $h_i(y^1) \geq 0$, то в силу теоремы 3 $\psi(y^1) < \psi(y_0)$. Возвращаемся к шагу 2 с заменой y_0 на y^1 .

7. Если $f(y^1) < f(y_0)$ и некоторые $h_i(y^1) \not\geq 0$, то вычисляем

$$\theta_m = \max \{ \theta / h_i(y_0 + \theta(y^1 - y_0)) \geq 0, i = 1, \dots, p, 0 \leq \theta \leq 1 \}.$$

а) Если $\theta_m > 0$, то в силу теоремы 3 $\psi(y_0 + \theta_m(y^1 - y_0)) < \psi(y_0)$. Возвращаемся к шагу 2 с заменой y_0 на $y_0 + \theta_m(y^1 - y_0)$.

б) Если $\theta_m = 0$, то для некоторых i и j имеет место $h_{ij}(y_0) = 0$ и $h_{ij}(y^1) < 0$. Тогда может быть показано, что строка j матрицы A_{i2} может быть заменена некоторой строкой A_{i1} , что приводит к новому оптимальному базису в i -й линейной подзадаче. Используя этот новый базис, возвращаемся к шагу 3.

Для того чтобы доказать утверждение, сделанное в шаге 7б, заметим, что из $h_{ij}(y_0) = 0$ следует, что выполняются такие соотношения:

$$A_{i1} x_i^0 = b_{i1}(y_0), \quad (58)$$

$$A_{i2}^j x_i^0 = b_{i2}^j(y_0), \quad (59)$$

где A_{i2}^j и b_{i2}^j являются j -ми строками A_{i2} и b_{i2} соответственно. В силу предположения [3] $A_{i2}^j \neq 0$, так что $A_{i2}^j A_{i1}^{-1} \neq 0$. Таким образом, A_{i2}^j

может быть заменено на некоторую строку A_{i1} , что приводит к неособой матрице $\tilde{A}_{i1}$. Соотношение

$$\tilde{A}_{i1}x_i^0 = b_{i1}(y_0) \quad (60)$$

должно сохранять силу, и, так как $\tilde{A}_{i1}$ неособая, x_i^0 является единственным решением (60). Целевая функция двойственной задачи $c_i x_i$, следовательно, не изменяется, так что $\tilde{A}_{i1}$ является новым оптимальным базисом.

Поскольку в линейных подзадачах больше неравенств, чем неизвестных, для их решения наиболее эффективным является двойственный алгоритм. Сходный комментарий приложим и к координирующей задаче, поскольку ее главная трудность обычно состоит в наличии большого числа ограничений. Метод проекций градиента Розена [4] является, следовательно, весьма подходящим для решения координирующей задачи, так как он представляет собой двойственный метод, эффективность которого зависит в основном от числа переменных в задаче и только косвенным образом от числа ограничений. Этот метод может быть также использован для решения линейных подзадач. В [4] показано, что метод проекций градиента будет сходиться к глобальному решению задачи с выпуклой целевой функцией и линейными ограничениями при стремлении числа итераций к бесконечности. В работе [5] описывается модификация этого метода, которая продемонстрировала в численных экспериментах более быструю сходимость.

Важным вычислительным аспектом алгоритма является то, что на каждой итерации находится допустимое решение исходной задачи с невозрастающей целевой функцией. Таким образом, вычисления могут быть закончены в любой точке с получением решения, в котором целевая функция не выше, чем во всех предшествующих точках.

Конечная сходимость описанной схемы расчленения доказывается следующей теоремой.

Т е о р е м а 5 (конечная сходимость). *Если на шагах 5 и 7 не повторяются оптимальные базисы, то процедура будет находить оптимальное решение за конечное число итераций.*

Д о к а з а т е л ь с т в о. На каждой итерации, на которой не удовлетворяется критерий оптимальности, уменьшается целевая функция координирующей задачи $f(y)$, либо обнаруживается альтернативный оптимальный базис в одной или более подзадачах. Каждый раз, когда $f(y)$ уменьшается, мы должны переходить к новому множеству оптимальных базисов $\{A_{i1}\}$, отличному от использовавшихся на любом из предшествовавших циклов, поскольку повторение множества будет вызывать повторение координирующей задачи, а следовательно, и минимального значения f . Рассмотрим цикл, на котором $f(y)$ остается тем же

самым. Имеется конечное число оптимальных базисов для каждого значения y . Если ни один из этих оптимальных базисов не повторяется на шагах 5 и 7, то на любой итерации текущее множество оптимальных базисов не может совпадать с множеством, имевшимся на предшествующей итерации. Поскольку полное число возможных множеств $\{A_{ii}\}$ конечно, оптимальное множество должно быть найдено за конечное число итераций.

6.2.2. Использование алгоритма для оптимизации работы нефтехимических установок

На базе описанного выше алгоритма Розеном и его сотрудниками [6], [7] была создана вычислительная система для решения проблем оптимизации производства и распределения в нефтехимической промышленности. Входом для этой системы являются такие величины, как качественные характеристики продуктов, интенсивности и свойства входных потоков, требования потребителей, цены и т. д. Все данные приведены к форме, удобной для инженеров и других специалистов, формулирующих задачу. При использовании этих данных строится задача математического программирования в общей форме (6.2—1) — (6.2—2), где коэффициенты целевой функции, вектор правых частей и матрица ограничений — все предполагаются функциями y . Эта задача затем решается с помощью метода расчленения. В конечном счете, результаты представляются в форме наиболее удобной для тех, кто формулировал задачу, и даются на выход. Вследствие такой удобной формализации пользователи программы могут быть совершенно незнакомы с математическим программированием.

Эта система, кратко называется SYMROS (заглавные буквы английских слов в наименовании «система программирования для операций нефтепереработки»), использует в качестве блоков имитаторы процессов в установках, т. е. независимые программы моделирования процессов вводятся в систему в качестве подпрограмм. Их выходами являются свойства выходных потоков различных нефтехимических установок, а входами значения управляющих переменных, характерных для каждой установки (некоторые из переменных y), и величины других параметров, характерных для установки. Выходы используются для того, чтобы образовать коэффициенты матрицы $A_i(y)$, векторов $b_i(y)$ и коэффициентов $c_i(y)$ в (6.2—1) — (6.2—2). Такие имитирующие подпрограммы могут быть весьма большими для того, чтобы дать реалистическое описание установок, являющихся зачастую весьма сложными. SYMROS построена так, что имитирующие подпрограммы могут значительно изменяться без введения изменений в программы для остальной части системы.

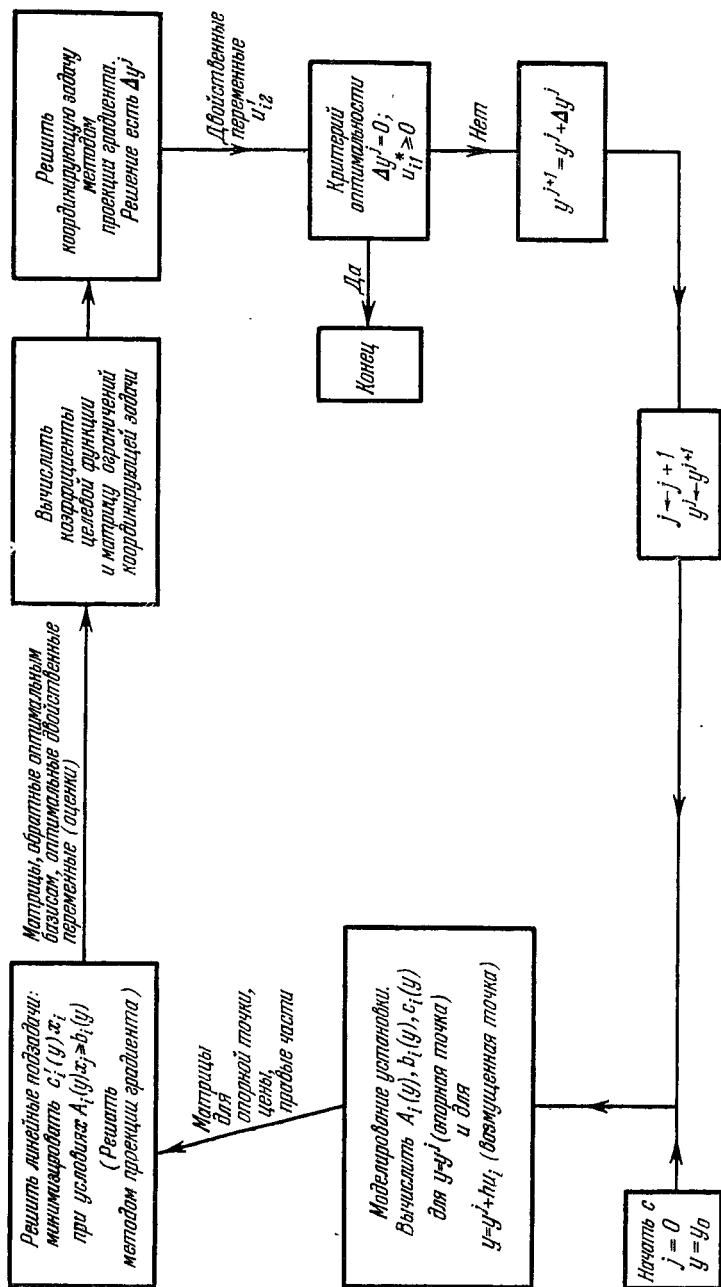


Рис. 6.1. Блок-схема подпрограммы процедуры расчленения.

Так как значения функций $A_i(y)$, $b_i(y)$ и $c_i(y)$ могут быть определены только с помощью больших вычислительных программ, аналитическое определение производных этих функций нецелесообразно. Координирующая задача (6.2.1—20) — (6.2.1—21) требует знания таких производных, поэтому их приходится оценивать по конечно-разностной схеме, вводя на вход имитирующих программ не только значения в базовой точке y , но и значения в возмущенных точках $y + hu_i$, где h положительно, а u_i есть i -й единичный вектор. При этом должно быть обеспечено, чтобы величина пробного шага h не была бы слишком малой, ввиду ограниченной точности программ, но и не была бы слишком большой, для того чтобы обеспечить эффективную аппроксимацию производных. Так как SYMROS является слишком большой системой, чтобы быть размещенной в МОЗУ, повторное обращение к имитирующим программам требует большого числа переносов подпрограмм с магнитной ленты в МОЗУ. В работе [7] указывается, что для больших задач большая часть времени тратится именно на эти переносы информации. Как указано в [6], [7], подпрограмма процедуры расчленения использует метод проектирования градиента Розена [4] для решения как линейных подзадач, так и нелинейной координирующей задачи. Блок-схема подпрограммы процедуры расчленения показана на рис. 6.1 (для простоты мы предполагаем, что $\Delta y^j \neq 0$, если критерий оптимальности не проходит). Более детальное описание системы читатель может найти в работах [6], [7].

§ 6.3. АЛГОРИТМ РАСЧЛЕНЕНИЯ БЕНДЕРСА ДЛЯ ЧАСТИЧНО-ЦЕЛОЧИСЛЕННЫХ ЗАДАЧ

6.3.1. Описание алгоритма

В этом параграфе мы опишем метод, представленный Бендерсом [8], для решения задач математического программирования следующего вида:

$$\left. \begin{aligned} \min \{c'x + f(y)\}, \\ Ax + F(y) \geq b, \\ x \geq 0, y \in S \end{aligned} \right\} \text{ задача } P1. \quad \begin{matrix} (1) \\ (2) \\ (3) \end{matrix}$$

Здесь матрица A имеет размерность $m \times n$, x и c являются n -векторами, y — p -вектор, f — скалярная функция от y , F — m -вектор, компоненты которого являются функциями y , b — m -вектор, а S — произвольное подмножество E^p . Функции f и F необязательно линейны, а S может быть, например, множеством векторов из E^p с целочисленными компонентами. В последнем случае задача $P1$ является частично-целочисленной, где

некоторые переменные являются непрерывными, а некоторые — дискретными. Пример такой задачи был дан в § 1.8.

Поскольку $P1$ линейна по x при фиксированных значениях y , естественно попытаться решить ее путем фиксации y , решения линейной задачи относительно x , получения «лучшего» значения y и т. д. Конечно, могут рассматриваться только такие значения y , для которых существует x , удовлетворяющее полученным линейным ограничениям, т. е. y должно лежать в множестве

$$R = \{y/\exists x \geq 0: Ax \geq b - F(y), y \in S\}. \quad (4)$$

Векторы $y \in R$ будут называться *допустимыми*.

Для того чтобы в явной форме получить ограничения, определяющие R , используем лемму Фаркаша [9].

Л е м м а Ф а р к а ш а. *Существует вектор $x \geq 0$, удовлетворяющий условиям $Bx = a \Leftrightarrow a'u \geq 0$ для всех u , удовлетворяющих $B'u \geq 0$.*

Применяя лемму к линейной задаче (y — фиксировано)

$$Ax - s = b - F(y), \quad (5)$$

$$x \geq 0, \quad s \geq 0, \quad (6)$$

приходим к выводу, что y допустимо, если и только если выполнено условие

$$(b - F(y))'u \leq 0 \quad (7)$$

для всех u , удовлетворяющих

$$A'u \leq 0, \quad u \geq 0. \quad (8)$$

Поскольку конус

$$C = \{u/A'u \leq 0, u \geq 0\} \quad (9)$$

является многогранным, то он определяется конечным числом образующих, т. е. существуют векторы $u_i^r, i = 1, \dots, n_r$ такие, что любой элемент $u \in C$ может быть представлен в виде

$$u = \sum_{i=1}^{n_r} \lambda_i u_i^r, \quad \lambda_i \geq 0. \quad (10)$$

Подставляя (10) в (7), получим

$$\sum_{i=1}^{n_r} \lambda_i (b - F(y))'u_i^r \leq 0, \quad (11)$$

что справедливо для всех $\lambda_i \geq 0$, если и только если

$$(b - F(y))'u_i^r \leq 0, \quad i = 1, \dots, n_r. \quad (12)$$

Таким образом, вектор $y \in S$ допустим, если и только если он удовлетворяет конечной системе ограничений (12). Множество R в (4) может быть, следовательно, представлено в виде

$$R = \{y / (b - F(y))'u_i' \leq 0, i = 1, \dots, n_r, y \in S\}. \quad (13)$$

Если R пусто, то исходная задача $P1$ не имеет допустимого решения. Предполагая R непустым, мы перепишем $P1$ в виде

$$\min_{y \in R} \{f(y) + \min \{c'x / Ax \geq b - F(y), x \geq 0\}\}. \quad (14)$$

Это описание соответствует нашему исходному подходу, т. е. фиксации y , решению линейной задачи относительно x , выбору лучшего значения y и т. д. Для фиксированного y внутренняя минимизация в (14) есть задача ЛП. Запишем ее вместе с двойственной к ней:

Прямая

$$\min c'x \text{ при } Ax \geq b - F(y), \quad x \geq 0. \quad (15)$$

Двойственная

$$\max (b - F(y))'u \text{ при } A'u \leq c, \quad u \geq 0. \quad (16)$$

Если $y \in R$, то прямая задача допустима и в силу теоремы двойственности имеем

$$\min \{c'x / Ax \geq b - F(y), x \geq 0\} = \max \{(b - F(y))'u / A'u \leq c, u \geq 0\}, \quad (17)$$

где значение максимума принимается равным $-\infty$, если двойственная задача недопустима. Соотношение (17), таким образом, суммирует утверждение, что если прямая имеет неограниченное решение, то двойственная недопустима, в то время как если обе задачи допустимы, то они имеют конечное оптимальное решение с равными значениями целевых функций.

Подставляя (17) в (14), приходим к новой форме задачи $P1$:

$$\min_{y \in R} \{f(y) + \max \{(b - F(y))'u / A'u \leq c, u \geq 0\}\}. \quad (18)$$

Рассмотрим множество ограничений двойственной задачи

$$P = \{u / A'u \leq c, u \geq 0\}. \quad (19)$$

Эти ограничения не зависят от y , и величины u_i' в (9) являются крайними лучами множества P . Если P пусто, то двойственная задача (15) имеет неограниченное решение, равно, как и исходная задача $P1$. Если P непусто, то внутренний максимум (18) достигается в одной из крайних точек P или стремится к ∞ при движении по крайнему лучу P . В последнем случае прямая задача (15) недопустима, что противоречит нашим исходным

предположениям, так что мы можем ограничиться рассмотрением только крайних точек P . Таких точек имеется конечное число. Обозначим их $u_i^p, i=1, \dots, n_p$. Таким образом, задача (18) может быть теперь переписана в виде

$$\min_{y \in R} \{f(y) + \max_{1 \leq i \leq n_p} (b - F(y))' u_i^p\}. \quad (20)$$

Однако эта проблема эквивалентна следующей:

$$\min z, \quad (21)$$

$$z \geq f(y) + (b - F(y))' u_i^p, \quad i = 1, \dots, n_p, \quad (22)$$

$$y \in R. \quad (23)$$

Используя определение R , данное ограничениями (13), мы можем переписать (21) — (23) и получить новую задачу $P2$:

$$\left. \begin{aligned} \min z, \\ z \geq f(y) + (b - F(y))' u_i^p, \quad i = 1, \dots, n_p, \\ (b - F(y))' u_i^r \leq 0, \quad i = 1, \dots, n_p, \\ y \in S \end{aligned} \right\} \text{задача } P2. \quad \begin{aligned} (24) \\ (25) \\ (26) \\ (27) \end{aligned}$$

Из предшествующего ясна связь между задачами $P1$ и $P2$. Суммируем эти результаты в виде следующей теоремы.

Теорема 1 (эквивалентность $P2$ и $P1$).

а) $P2$ имеет допустимое решение $\Leftrightarrow P1$ имеет допустимое решение;

б) $P2$ допустима, но не имеет оптимального решения $\Leftrightarrow P1$ допустима, но не имеет оптимального решения;

в) если (z^0, y^0) есть решение $P2$ и x^0 есть решение линейной задачи

$$\min c'x, \quad (28)$$

$$Ax \geq b - F(y^0), \quad x \geq 0, \quad (29)$$

то (x^0, y^0) является решением $P1$ и

$$z^0 = c'x^0 + f(y^0); \quad (30)$$

г) если (x^0, y^0) есть решение $P1$ и $z^0 = c'x^0 + f(y^0)$, то (z^0, y^0) есть решение $P2$.

Формальное доказательство теоремы оставим в качестве упражнения. Для краткости в дальнейшем изложении мы будем использовать следующее обозначение:

$$G = \{(z, y) / (z, y) \text{ удовлетворяет } (25) - (27)\}, \quad (31)$$

так что $P2$ может быть записана в виде

$$\left. \begin{array}{l} \min z, \\ (z, y) \in G \end{array} \right\} \text{ задача } P2.$$

Итерации путем последовательного добавления ограничений. Решение исходной задачи было приведено к решению задачи $P2$ с определением (z^0, y^0) , а также решению линейной задачи (28)—(29) для получения оптимального значения x . К сожалению, решение $P2$ затруднительно в силу того, что для каждой крайней точки и крайнего луча множества P имеется по одному ограничению, так что их общее число даже для проблемы умеренной размерности может быть очень большим. Однако лишь малая часть ограничений являются связывающими в оптимальном решении. Поэтому здесь естественно применить релаксационный подход § 4.2. Мы можем начать с учета только малого числа ограничений и решать модифицированные таким образом задачи $P2$, в дальнейшем обозначаемые $MP2$. Относительно просто проверить, удовлетворяет ли решение остающимся ограничениям. Если это так, то решение оптимально, поскольку целевая функция минимизируется на множестве, содержащем G . Если это не так, то добавляются ограничения, которые не удовлетворились на текущем решении, и задача решается заново. Результирующий алгоритм состоит в итеративном решении двух задач. Первая задача есть задача $MP2$ с переменными (x, y) , в которую последовательно добавляются ограничения. Вторая задача есть двойственная линейная задача (16) или прямая (15), которая служит для испытания на оптимальность решений $MP2$ и, если необходимо, дает новые ограничения. Заметим, что, поскольку мы до сих пор не предполагали вогнутости — $f(y)$ или $F(y)$, то в процедуре релаксации необоснованно опускать несвязывающие ограничения. Последнее следует из того, что теорема 4.2.1 уже оказывается несправедливой.

Для детального раскрытия процедуры необходимо сделать следующие предположения.

Предположение 1. S является замкнутым и ограниченным.

Предположение 2. $f(y)$ и компоненты $F(y)$ являются непрерывными на S .

Эти предположения удовлетворяются в большинстве приложений и позволяют избежать трудностей рассмотрения задач, имеющих неограниченное решение. Если S не является ограниченным, то могут быть добавлены покомпонентные ограничения сверху и снизу на y . При этом границы могут быть взяты столь большими, что они заведомо включают оптимальное решение, либо таковы, что любое решение, превышающее эти ограничения,

не может быть реально интерпретировано. Для дальнейшего необходимо следующая лемма.

Лемма 1. Если выполнены предположения 1 и 2 и P_2 является допустимой, то z не имеет конечной нижней границы на G , если и только если P в (19) является пустым.

Доказательство. *Достаточность.* По предположению, существует точка $(z^*, y^*) \in G$. Если P пусто, то она не содержит никаких крайних точек или крайних лучей и ограничения, определяющие G , сводятся просто к условию

$$y \in S \quad (32)$$

и $(z, y^*) \in G$ для любых значений z .

Необходимость. Если P непусто, то имеется по крайней мере одна крайняя точка $\hat{u}$, так что ограничения P_2 включают неравенство

$$z \geq (b - F(y))' \hat{u} + f(y). \quad (33)$$

Тогда для любых $(z, y) \in G$ должно выполняться условие

$$z \geq \min_{y \in S} \{(b - F(y))' \hat{u} + f(y)\}. \quad (34)$$

Но S замкнуто и ограничено, а F и f непрерывны на S . Следовательно, по теореме Вейерштрасса, правая часть в (34) конечна, следовательно, z имеет конечную нижнюю границу на G .

Рассмотрим теперь задачу MP_2 , в которой включено только некоторое подмножество ограничений, определяющих G ,

$$\min z, \quad (35)$$

$$z \geq f(y) + (b - F(y))' u_i^p, \quad i \in I_1, \quad (36)$$

$$(b - F(y))' u_i^r \leq 0, \quad i \in I_2, \quad (37)$$

$$y \in S,$$

где I_1 и I_2 — соответствующие подмножества индексов $1, \dots, n_p$ и $1, \dots, n_r$. Пусть G' — множество всех (z, y) , удовлетворяющих ограничениям MP_2 . Так как не все ограничения включены, $G' \supseteq G$. Таким образом, MP_2 сводится к минимизации целевой функции задачи P_2 на множестве, содержащем множество ограничений P_2 . Значит, решение MP_2 оптимально для P_2 , если и только если оно лежит в G , т. е. удовлетворяет всем ограничениям P_2 . Конечно, эти ограничения в явной форме недоступны, поскольку заранее не вычисляются все крайние точки и лучи множества P . Однако для проверки принадлежности к G или генерации новых ограничений могут быть использованы линейные задачи (15) — (16). Процедура генерации ограничений, или, что то же самое, строк в матрице ограничений, двойственна к

процедурам генерации столбцов, описанным ранее в § 1.6 и § 2.3 и в главе III.

При выполнении предположений 1 и 2, если $MP2$ допустима, то она имеет конечное оптимальное решение (z^0, y^0) . Это решение оптимально для $P2$, если и только если

$$(b - F(y^0))' u_i^p \leq z^0 - f(y^0), \quad i = 1, \dots, n_p, \quad (38)$$

$$(b - F(y^0))' u_i^r \leq 0, \quad i = 1, \dots, n_r. \quad (39)$$

Найдем наиболее неудовлетворенное ограничение (38), т. е. ограничение, на котором достигается

$$\max_{1 \leq i \leq n_p} (b - F(y^0))' u_i^p \leq z^0 - f(y^0).$$

Однако если линейная функция $(b - F(y^0))' u$ достигает конечного максимума на P , то она достигает его в крайней точке P , так что максимизация в приведенном неравенстве эквивалентна решению линейной задачи

$$\left. \begin{array}{l} \max (b - F(y^0))' u, \\ u \in P \end{array} \right\} \begin{array}{l} \text{линейная} \\ \text{подзадача.} \end{array} \quad (40)$$

Индекс крайней точки, в которой достигается решение (40), совпадает, очевидно, с индексом наиболее неудовлетворенного ограничения (38). Задача (40) совпадает с двойственной задачей (16) при $y = y^0$. Поскольку z^0 предполагается конечным, и

$$\min \{z/(z, y) \in G\} \geq z^0, \quad (41)$$

то, в силу леммы 1, P не пусто. Таким образом, эта задача имеет конечное оптимальное решение, либо неограниченное решение. Если оно не ограничено, то целевая функция стремится к $+\infty$ на полупрямой $u_i^p + \lambda u_i^r, \lambda \geq 0$. В этом случае

$$(b - F(y^0))' u_i^r > 0 \text{ для некоторого } i.$$

Из этого следует, что нарушается одно из ограничений (39). Таким образом, оба множества ограничений (38) и (39) удовлетворяются, если и только если

$$\max \{(b - F(y^0))' u \mid A'u \leq c, u \geq 0\} \leq z^0 - f(y^0). \quad (42)$$

Заметим, однако, что (38) должно выполняться как равенство для некоторых i . Иначе говоря, z^0 не может быть уменьшено без нарушения каких-нибудь ограничений. Следовательно, если (42) выполняется, то оно выполняется и как равенство, так что может быть сформулирован следующий критерий оптимальности.

Теорема 2 (критерий оптимальности). Если (z^0, y^0) оптимально для $MP2$, то оно является оптимальным для $P2$, если и

только если

$$\max \{(b - F(y^0))' u / A'u \leq c, u \geq 0\} = z^0 - f(y^0). \quad (43)$$

Если критерий оптимальности не проходит, то не удовлетворяются некоторые из ограничений (38) и (39), так что

$$\max \{(b - F(y^0))' u / A'u \leq c, u \geq 0\} > z^0 - f(y^0). \quad (44)$$

Если максимум в приведенном выше выражении достигается в крайней точке u^0 , то

$$(b - F(y^0))' u^0 > z^0 - f(y^0). \quad (45)$$

Возвращаясь к рассмотрению ограничений задачи $P2$, замечаем, что (45) устанавливает новые ограничения, которые должны быть введены в новую задачу $MP2$.

Если линейная подзадача (40) неограничена, то симплекс-метод приводит к крайней точке P , обозначаемой u^0 , и крайнему лучу P , обозначаемому v^0 , таким, что $(b - F(y^0))' u$ стремится к $+\infty$ вдоль полупрямой

$$u = u^0 + \lambda v^0, \quad \lambda \geq 0. \quad (46)$$

Для того чтобы это имело место, v^0 должно удовлетворять условиям

$$(b - F(y^0))' v^0 > 0, \quad (47)$$

что задает новое ограничение в форме (39), которое также должно быть добавлено к текущей задаче $MP2$ для формирования ее нового вида.

В случае неограниченности может случиться, что (45) выполняется в крайней точке u^0 , указанной в (46). В этом случае при формировании новой задачи $MP2$ должны быть учтены как ограничения (45), так и ограничение (47).

Сводное изложение алгоритма:

1. Начать процедуру с решения проблемы $MP2$, в которой только несколько (или ни одного) ограничений вида (35) — (36).

2. Решаем проблему $MP2$. Если $MP2$ недопустима, то недопустимы и $P2$ и $P1$. В противном случае мы получаем конечное оптимальное решение (z^0, y^0) , либо информацию о том, что решение является неограниченным. Если $z^0 = -\infty$, то принимаем в качестве y^0 произвольный элемент S и переходим к шагу 3.

3. Решаем двойственную линейную задачу (16) (или прямую (15), если она допустима). Если двойственная недопустима, то в силу леммы 1 исходная задача $P1$ имеет неограниченное решение. Если двойственная неограниченная, то переходим к шагу 6.

4. Если оптимальное значение целевой функции на шаге 3 равно $z^0 - f(y^0)$, то решение (z^0, y^0) является решением $P2$. Если x^0 — решение (15), то (x^0, y^0) решение $P1$.

5. Если на шаге 3 не проходит критерий оптимальности, и двойственная (16) имеет конечное оптимальное решение u^0 , то

$$z^0 < f(y^0) + (b - F(y^0))' u^0, \quad (48)$$

так что текущее решение $MP2$ не удовлетворяет ограничению

$$z \geq (b - F(y))' u^0 + f(y). \quad (49)$$

Добавим это ограничение к $MP2$ и вернемся к шагу 2.

6. Если двойственная (16) имеет неограниченное решение, то симплекс-метод позволяет найти луч v^0 и точку u^0 такие, что целевая функция двойственной задачи стремится к $+\infty$ на луче

$$u = u^0 + \lambda v^0, \quad \lambda \geq 0. \quad (50)$$

Неравенство

$$(b - F(y^0))' v^0 > 0 \quad (51)$$

удовлетворяется, так что y^0 не удовлетворяет ограничению

$$(b - F(y))' v^0 \leq 0. \quad (52)$$

Добавим это ограничение к $MP2$. Если также имеет место

$$z^0 < f(y^0) + (b - F(y^0))' u^0 \quad (53)$$

для крайней точки u^0 , указанной в (50), то добавим ограничение вида (49) к $MP2$. Возвращаемся к шагу 2.

Конечная сходимость алгоритма прямо следует из конечности числа ограничений в $P2$.

Теорема 3 (конечная сходимость). Описанная выше итеративная процедура заканчивается за конечное число итераций с получением оптимального решения $P1$, либо с получением информации о том, что $P1$ недопустима или неограничена.

Доказательство. Задача $P2$ имеет только конечное число ограничений. Если критерий оптимальности не проходит, то одно или более новых ограничений добавляется к $MP2$. Следовательно, за конечное число итераций выполняется критерий оптимальности, либо используется полное множество ограничений. $P1$ недопустима, если и только если $MP2$ недопустима на некотором шаге. $P1$ неограничена, если и только если двойственная линейная задача (16) недопустима, что будет обнаружено на первом шаге.

Верхние и нижние границы. Привлекательной чертой этого алгоритма является возможность вычисления верхней и нижней границ значения оптимальной целевой функции, причем таких, что они сходятся к оптимальному значению тогда, когда в процессе решения достигается оптимальность. Верхняя оценка генерируется последовательностью допустимых решений задачи $P1$, так что лучшая из этих оценок может быть взята в

качестве решения, если процедура заканчивается раньше достижения полного оптимума.

Для того чтобы получить верхнюю оценку, предположим, что (z^i, y^i) есть решение $MP2$ на шаге i , и предположим, что линейная задача (15) имеет допустимое решение при $y = y^i$. Если x^i — решение (15), то пара (x^i, y^i) допустима для $P1$ и имеет место неравенство

$$c'x^i + f(y^i) \geq \min \{z/(z, y) \in G\}. \quad (54)$$

Получение нижней границы следует из того факта, что ограничения добавляются к $MP2$, так что, если z^i оптимальное значение целевой функции для $MP2$ на шаге i , то

$$z^i \leq z^{i+1} \leq \min \{z/(z, y) \in G\}. \quad (55)$$

Уравнения (54) и (55) на шаге i дают следующие оценки:

$$z^i \leq \min \{z/(z, y) \in G\} \leq \min_{1 \leq l \leq 1} (c'x^l + f(y^l)). \quad (56)$$

По мере сходимости итеративной процедуры как правая, так и левая части неравенства (56) сходятся к величине, заключенной между ними. Действительно, критерий оптимальности (43) просто является условием того, что верхняя и нижняя границы равны.

Решение задачи $MP2$. Хотя процедура формально приложима к очень широкому классу задач, она практически эффективна только тогда, когда задача $MP2$ может быть эффективно решена. Случаи, когда это имеет место, таковы:

1. S является множеством векторов в E^p с неотрицательными целочисленными компонентами, $F(y) = By$ и $f(y) = d'y$. Задача $P1$ в этом случае является задачей ЛП с некоторыми целочисленными и с некоторыми непрерывными переменными. Задача $MP2$ является чисто целочисленной линейной программой, для которой применима различная известная техника решения [10]. Так как последовательные задачи $MP2$ отличаются только добавлением одного или двух ограничений, то для решения $MP2$ удобны методы отсечения типа метода Гомори. Алгоритм Бендера, очевидно, является наиболее обещающим именно для таких приложений [10].

2. S определено множеством линейных и нелинейных неравенств

$$S = \{y/g_i(y) \leq 0, i = 1, \dots, r, a_j' y \leq b_j, j = 1, \dots, s\},$$

где g_i нелинейные, но непрерывные и дифференцируемые функции. Функции $f(y)$ и $F(y)$ могут быть нелинейными, но так же непрерывными и дифференцируемыми. Тогда $MP2$ имеет линейную целевую функцию, но нелинейные ограничения, и для ее

решения кажется наиболее подходящим метод штрафных функций, точнее, смешанный метод внутренней и внешней точки, описанный в [11], [12]. Ограничения, которые удовлетворялись на предшествующих шагах, образуют в штрафной функции слагаемые внутреннего типа, в то время как новые ограничения вводятся слагаемыми внешнего типа. В этом случае ранее найденное оптимальное решение MP_2 может быть использовано в качестве начальной точки для вновь построенной штрафной функции. Если F вогнута и f и g выпуклы, MP_2 является задачей выпуклого программирования. Тогда при слабых дополнительных предположениях [12], алгоритм штрафных функций будет сходиться к оптимальному решению, впрочем, так же как и множество других методов. Заметим, что в этом случае несвязующие ограничения могут быть опущены в MP_2 , как объяснялось раньше в § 4.2.

При любых применениях процедура Бендерса обладает тем желательным свойством, что она сохраняет любую специальную структуру матрицы A , что, вообще говоря, несправедливо, если задача решается без какой-либо формы расчленения. Таким образом, если матрица A транспортного типа, то задача (15) будет также транспортной задачей, в которой правые части изменяются на каждом шаге. В этом случае желательно решать (15) равно как и двойственную (16). Так как на каждой итерации в (15) изменяются только правые части, оптимальное двойственное решение на итерации i остается допустимым и для новой двойственной на итерации $i+1$. Таким образом, двойственный симплекс-метод представляется наиболее подходящим для ее решения. Конечно, если (15) неразрешима, то в двойственной должны генерироваться крайние лучи, т. е. неограниченные ребра. Однако это легко сделать при применении как прямого, так и двойственного симплекс-метода (см. задачи 6 и 7).

6.3.2 Связь принципа декомпозиции и алгоритмов отсечения

Для того чтобы обнаружить связь между алгоритмом Бендерса и декомпозиционной процедурой Данцига — Вулфа, предположим, что исходная задача (6.3.1—1) — (6.3.1—3) является задачей линейного программирования, представленной ниже вместе с двойственной к ней.

$$\text{Прямая: } \min(c'x + d'y) \text{ при } Ax + By \geq b, x \geq 0, y \geq 0; \quad (1)$$

$$\text{Двойственная: } \max b'u \text{ при } B'u \leq d, A'u \leq c, u \geq 0. \quad (2)$$

Для простоты предположим, что многогранное множество

$$P = \{u/A'u \leq c, u \geq 0\} \quad (3)$$

является ограниченным. Тогда задача $P2$ приобретает такой вид:

$$\left. \begin{array}{l} \min z \\ \text{при ограничениях} \\ z + (By - b)' u_j \geq d' y, \forall j, \\ y \geq 0 \end{array} \right\} \text{задача Бендерса } P2, \quad (4)$$

где u_j есть j -я крайняя точка многогранника P . Задача $MP2$ имеет тот же самый вид, однако, учитывается только некоторое подмножество ограничений с номерами j . Критерий оптимальности таков:

$$\max \{(b - By^0)' u / A'u \leq c, u \geq 0\} = z^0 - d'y^0 \left\{ \begin{array}{l} \text{критерий оп-} \\ \text{тимальнос-} \\ \text{ти Бен-} \end{array} \right. \quad (5)$$

дерса,

где (z^0, y^0) — решение $MP2$. Для проверки этого критерия, очевидно, необходимо решить линейную подзадачу:

$$\left. \begin{array}{l} \max (b - By^0)' u \\ \text{при ограничениях} \\ A'u \leq c, \\ u \geq 0 \end{array} \right\} \begin{array}{l} \text{линейная} \\ \text{подзадача} \\ \text{Бендерса.} \end{array} \quad (6)$$

Теперь применим принцип декомпозиции к двойственной задаче (2). Запишем любое $u \in P$ в виде выпуклой комбинации крайних точек P :

$$u = \sum_i \lambda_i u_i, \quad \sum_i \lambda_i = 1, \quad \lambda_i \geq 0 \quad (7)$$

и подставим результат в остальные ограничения двойственной задачи и ее целевую функцию, что приводит к следующей координирующей задаче:

$$\left. \begin{array}{l} \max \sum_i (b' u_i) \lambda_i \\ \text{при ограничениях} \\ \sum_i (B' u_i) \lambda_i \leq d, \\ \sum_i \lambda_i = 1, \\ \lambda_i \geq 0 \end{array} \right\} \begin{array}{l} \text{координиру-} \\ \text{ющая задача.} \end{array} \quad (8)$$

Двойственная к координирующей задаче такова:

$$\min (d'y + v)$$

при ограничениях

$$\begin{aligned} v + (By)' u_j &\geq b' u_j, \\ y &\geq 0, \end{aligned}$$

или, полагая

$$z = d' y + v, \quad (9)$$

получим

$$\min z \quad \left\{ \begin{array}{l} \text{двойственная} \\ \text{при ограничениях} \\ z + (By - b)' u_j \geq d' y, \forall j, \\ y \geq 0 \end{array} \right. \quad \left\{ \begin{array}{l} \text{к координи-} \\ \text{рующей задаче} \end{array} \right. = \text{задача Бендерса } P2. \quad (10)$$

Сравнение с (4) показывает, что задача $P2$ в процедуре Бендерса и двойственная к координирующей задаче в схеме декомпозиции являются идентичными. Задача $MP2$ двойственная к сокращенной координирующей задаче, содержащей только подмножество допустимых столбцов. Пусть (y^0, v^0) — двойственные переменные, соответствующие оптимальному решению некоторой сокращенной координирующей задачи. Для того чтобы выяснить, является ли это решение одновременно и решением координирующей задачи, вычислим максимальную из характеристических разностей

$$\bar{c}_j = b' u_j - (y^0)' B' u_j - v^0. \quad (11)$$

Так как u_j крайняя точка многогранника P , то поиск максимальной $\bar{c}_j$ эквивалентен решению подзадачи:

$$\left. \begin{array}{l} \max (b - By^0)' u \\ \text{при ограничениях} \\ A' u \leq c, \\ u \geq 0 \end{array} \right\} \begin{array}{l} \text{подзадача в} \\ \text{принципе} \\ \text{декомпозиции} \end{array} = \text{подзадача Бендерса}, \quad (12)$$

которая совпадает с подзадачей (6) в алгоритме Бендерса.

Критерий оптимальности координирующей задачи таков:

$$\max \bar{c}_j \leq 0 \quad (13)$$

или, в силу (11),

$$\max_j (b - By^0)' u_j \leq v^0. \quad (14)$$

В § 2.6 было показано, что это неравенство в действительности выполняется как равенство. Из (9) следует

$$z^0 = d' y^0 + v^0, \quad (15)$$

и (14) преобразуется к виду

$$\max \{(b - By^0)' u / A'u \leq c, u \geq 0\} = z^0 - d'y^0. \quad (16)$$

Сравнение с условием (5) показывает, что (16) эквивалентно критерию оптимальности в процедуре Бендерса.

Суммируя, можно утверждать, что алгоритм Бендерса и принцип декомпозиции Данцига—Вулфа в действительности являются двойственными друг к другу. Если рассмотреть задачи, двойственные к паре (1)—(2), рассмотренной при описании алгоритма Бендерса, то их решение с помощью принципа декомпозиции в точности соответствует алгоритму Бендерса. Конечно, когда рассматриваются нелинейные задачи, существуют важные различия между двумя процедурами. Наиболее важное из них заключается в том, что алгоритм Бендерса может быть применен для значительно более широкого класса задач, чем существующие обобщения принципа декомпозиции.

Процедура Бендерса также весьма сходна с методами отсечения целочисленного программирования [10] и выпуклого программирования [13]. Задача P_2 решается с помощью последовательного отсечения решений, которые не удовлетворяют всем

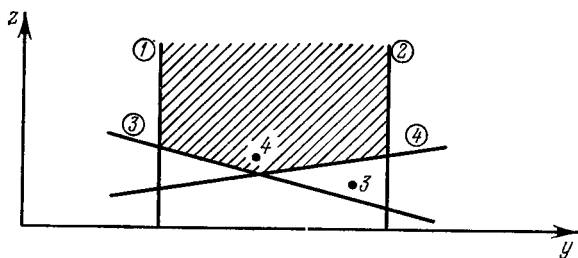


Рис. 6.2.

ограничениям, как показано на рис. 6.2. Ограничения, отмеченные индексами 1 и 2, соответствуют крайним лучам P . Для случая скалярной переменной y эти ограничения попросту ограничивают диапазон изменения y . Ограничение 3 соответствует крайней точке многогранника P и в силу предположений 1 и 2 обеспечивает то, что минимум z является конечным. При задании множества S требуется, чтобы y было неотрицательным целочисленным, так что точка 3 является оптимальным решением для ограничений 1—3. Ограничение 4 отсекает это решение, приводя к тому, что ограничивающим множеством в задаче MP_2 является заштрихованная область. Тогда точка 4 будет новым оптимальным решением, в котором значение z больше, чем в точке 3.

6.3.3. Приложения к задаче о размещении складов

Балинский и Вулф [10, 14] применили алгоритм Бендерса к задаче о размещении складов с учетом фиксированных затрат. Общая задача может быть сформулирована так. Задано предприятие, производящее единственный продукт, конечное множество мест, в которых могут быть построены склады, и конечное множество потребителей с известным спросом. Требуется определить, где, в каких точках следует разместить склады, чтобы общие затраты на перевозку к складам и хранение в них, а также затраты при перевозке продукта к потребителям были минимальны.

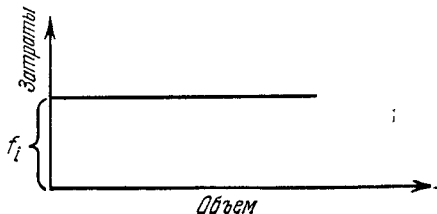


Рис. 6.3.

Задача, рассмотренная Балинским и Вулфом, является специализированным вариантом указанной выше задачи, причем используются следующие дополнительные предположения:

1. Предприятие и все склады имеют бесконечную емкость.
2. Стоимость перевозки и хранения на складах является фиксированной положительной величиной, если перевозится положительное количество, или нулем в противном случае, как показано на рис. 6.3 для склада i . Фиксированные затраты представляют собой капиталовложения, требуемые для строительства склада.

3. Затраты на доставку к потребителю j из склада i являются линейными функциями объема продукции с ценой за единицу, равной $\hat{c}_{ij} \geq 0$.

Задача может быть сформулирована, как частично-целочисленная задача линейного программирования, в которой переменные типа 0, 1 представляют альтернативы строительства или нестроительства данного склада. Если $\hat{x}_{ij}$ есть объем продукции, перевозимой со склада i к потребителю j , то должно быть выполнено условие

$$\sum_i \hat{x}_{ij} \geq d_j, \quad j=1, \dots, n \quad \hat{x}_{ij} \geq 0. \quad (1)$$

Введем в рассмотрение обозначение

$$x_{ij} = \frac{\hat{x}_{ij}}{d_j}, \quad c_{ij} = d_j \cdot \hat{c}_{ij}. \quad (2)$$

Тогда задача может быть представлена в следующем виде:

$$\min \left(\sum_i f_i y_i + \sum_{i,j} c_{ij} x_{ij} \right), \quad (3)$$

$$\sum_i x_{ij} \geq 1, \quad j = 1, \dots, n, \quad \left. \begin{array}{l} \text{прямая} \\ \text{задача.} \end{array} \right\} \quad (4)$$

$$\left. \begin{array}{l} 0 \leq x_{ij} \leq y_i, \quad \forall i, j, \\ y_i = 0 \text{ либо } 1 \end{array} \right\} \quad (5) \quad (6)$$

Если $y_i = 1$, тогда используется склад i , так как условие (5) не налагает дополнительных ограничений на выбор x_{ij} , и величина f_i может не учитываться. Если $y_i = 0$, то $x_{ij} = 0$ для всех j , и капитальные затраты равны нулю. Пусть $y = (y_1, \dots, y_m)$ фиксировано; тогда (3)—(5) представляет собой обычную задачу ЛП. Алгоритм Бендерса добавляет ограничения к чисто целочисленной задаче $MP2$ путем решения двойственной к (3)—(5), которая имеет вид

$$\max \left(\sum_j v_j - \sum_{i,j} y_i u_{ij} \right), \quad \left. \begin{array}{l} \text{двойственная} \\ \text{задача} \end{array} \right\} \quad (7)$$

$$v_j - u_{ij} \leq c_{ij}, \quad (8)$$

$$v_j \geq 0, \quad u_{ij} \geq 0 \quad (9)$$

Многогранное множество P в этой задаче представляет собой множество всех переменных, удовлетворяющих (8) и (9), а задача $P2$ имеет вид

$$\begin{aligned} & \min z, \\ & z \geq \sum_i v_i^p - \sum_{i,j} u_{ij}^p y_i + \sum_i f_i y_i, \end{aligned} \quad (10)$$

$$\sum_i v_i^r - \sum_{i,j} u_{ij}^r y_i \leq 0, \quad (11)$$

$$y_i = 0 \text{ либо } 1, \quad (12)$$

где (10) должно выполняться для всех крайних точек $\{v_i^p, u_{ij}^p\}$ множества P , а (11)—для всех крайних лучей. Как было показано ранее, неравенства (11) являются необходимыми и достаточными условиями того, чтобы y было допустимым, т. е. для того, чтобы существовали допустимые значения x_{ij} в (3)—(5). Однако проверка (3)—(5) показывает, что любое $y \neq 0$ допустимо, так что неравенства (11) могут быть опущены, если условие $y \neq 0$ выполняется. Алгоритм может быть начат с учета нескольких или даже ни одного из ограничений вида (10) задачи $P2$. Пусть решение этой ограниченной задачи $MP2$ есть y^0 . После этого должна быть решена двойственная пара задач (3)—(5) и (7)—(9), причем y фиксируется на значении y^0 . Оптимальное решение может быть получено в явной форме. Пусть множество

складов, которые предполагается открыть на данной итерации, обозначено так:

$$I = \{i/y_i^0 = 1\}. \quad (13)$$

Мы предполагаем, что условие $y \neq 0$ выполнено, так что I не пусто. Задача (3)—(5) приобретает вид

$$\min \sum_{i \in I} \sum_j c_{ij} x_{ij}, \quad (14)$$

$$\sum_{i \in I} x_{ij} \geq 1, \quad j = 1, \dots, n, \quad x_{ij} \geq 0. \quad (15)$$

Поскольку $c_{ij} \geq 0$, то ограничения (15) на оптимальном решении выполняются, как равенства. Так как ограничения (15) независимы, то решение находится непосредственно, если положить $x_{ij} = 1$ для минимальных c_{ij} и нулем в противоположном случае, т. е.

$$x_{k_j j}^0 = 1, \quad \text{где } c_{k_j j} = \min_{i \in I} c_{ij}, \quad j = 1, \dots, n, \quad (16)$$

$$x_{ij}^0 = 0 \quad \text{в остальных случаях.} \quad (17)$$

Это решение связывает потребителей с наиболее выгодным складом k_j . Для того чтобы получить соответствующее двойственное решение, используем условие дополняющей нежесткости:

$$v_j = u_{k_j j} + c_{k_j j}, \quad j = 1, \dots, n. \quad (18)$$

Остальные двойственные ограничения таковы:

$$u_{ij} \geq 0, \quad (19)$$

$$u_{ij} \geq v_j - c_{ij}, \quad i \neq k_j. \quad (20)$$

Подставляя (18) в целевую функцию (7), мы убеждаемся, что ее значение не зависит от $u_{k_j j}$, поскольку $\sum_j u_{k_j j}$ появляется дважды с противоположными знаками. Таким образом, мы можем положить $u_{k_j j} = 0$. Для того чтобы максимизировать остающиеся слагаемые, u_{ij} должны быть взяты настолько малыми, насколько позволяют ограничения (19)—(20), т. е. должны быть приняты равными максимуму из значений правой части. Это приводит к следующему оптимальному двойственному решению:

$$v_j^0 = c_{k_j j}, \quad j = 1, \dots, n, \quad (21)$$

$$u_{ij}^0 = \max \{0, v_j^0 - c_{ij}\}, \quad i \neq k_j. \quad (22)$$

Пусть d^0 — оптимальное значение целевой функции двойственной задачи, а z^0 оптимальное значение, полученное при

решении $MP2$. Если

$$d^0 = z^0 - \sum_i f_i y_i^0, \quad (23)$$

то по теореме 6.3.1—2 текущее решение оптимально; если — нет, то образуется новое ограничение с помощью $\{v_j^0, u_{ij}^0\}$,

$$z \geq \sum_i v_i^0 - \sum_{i,j} u_{ij}^0 y_i + \sum_i f_i y_i. \quad (24)$$

Это ограничение добавляется к $MP2$. Интерпретацию этого ограничения можно найти в [10].

6.3.4. Численный пример

Рассмотрим следующий пример, взятый из [10, стр. 293], представляющий собой проблему размещения вида (6.3.3-3) — (6.3.3-6) с четырьмя пунктами размещения и потребителями, причем $f_i = 7$ для всех i , а матрица затрат на перевозки

$$(c_{ij}) = \begin{bmatrix} 0 & 12 & 20 & 18 \\ 12 & 0 & 8 & 6 \\ 20 & 8 & 0 & 6 \\ 18 & 6 & 6 & 0 \end{bmatrix}.$$

Цикл. 1. Первоначально в $MP2$ не вводится никаких ограничений. Выберем $y^0 = (0 \ 1 \ 0 \ 0)$ и положим $z^0 = -\infty$. Решая пару линейных задач (3)—(5) и (7)—(9), получаем

$$\left. \begin{aligned} \sum_{i,j} c_{ij} x_{ij}^0 + \sum_i f_i y_i^0 &= 33, \\ v_1^0 &= 12, v_2^0 = 0, v_3^0 = 8, v_4^0 = 6, \\ u_{11}^0 &= 12, u_{33}^0 = 8, u_{44}^0 = 6, \text{ остальные } u_{ij} &= 0. \end{aligned} \right\} \quad (1)$$

Начальная задача $MP2$ такова:

$$\begin{aligned} \min z \\ z \geq 26 - 5y_1 + 7y_2 - y_3 - y_4, \end{aligned} \quad (2)$$

$$y_i = 0 \quad \text{либо} \quad 1. \quad (3)$$

Оптимальное решение, минимизирующее правую часть ограничения (2), состоит в том, чтобы положить $y_i = 1$ для отрицательных коэффициентов, и нулю в противном случае, так что

$$y^0 = (1 \ 0 \ 1 \ 1), \quad (4)$$

причем оптимальное значение целевой функции

$$z^0 = 19. \quad (5)$$

Значение 33 дает верхнюю границу оптимума, а значение 19 — нижнюю границу. Так как они не равны, то текущее решение не оптимально.

Цикл 2. Решая (3)—(5) и (7)—(9), при значениях y , указанных выше, получаем:

$$\begin{aligned} \sum_i f_i y_i^0 + \sum_{i,j} c_{ij} x_{ij}^0 &= 27, \\ v_1^0 = v_3^0 = v_4^0 &= 0, \quad v_2^0 = 6, \\ u_{22}^0 &= 6, \text{ остальные } u_{ij}^0 = 0. \end{aligned}$$

Далее добавляется новое ограничение

$$z \geq 6 + 7y_1 + y_2 + 7y_3 + 7y_4, \quad (6)$$

и решается новая задача $MP2$ минимизации z при ограничениях (2), (3) и (6). Эта задача эквивалентна следующей:

$$\min_{y_i=0 \text{ либо } 1} \max \{ \text{прав.ч. (2), прав.ч. (6)} \}. \quad (7)$$

Ее решение может быть найдено путем перебора: $y_0 = (1001)$ или (1010) , $z_0 = 20$. Поскольку вновь верхняя и нижняя границы не равны, то должен быть проделан следующий цикл.

В таблице 6.3.1 суммировано поведение решений этой задачи по циклам. В этой таблице y^0 — оптимальное решение задачи $MP2$ при ограничениях, указанных в выше стоящих строчках справа, причем соответствующее значение целевой функции дано в строке, лежащей непосредственно над y^0 (например, $y^0 = (1010)$ — оптимальное решение задачи при трех неравенствах и с минимальным значением целевой функции $z^0 = 20$). Столбец, озаглавленный $\sum_{i,j} c_{ij} x_{ij}^0 + \sum_i f_i y_i^0$, дает минимальные затраты

задачи (3)—(6) при $y = y^0$ и представляет собой верхнюю границу минимальных затрат. Значение z^0 задает нижнюю границу.

Таблица 6.3.1

$y_0 = (y_1^0, y_2^0, y_3^0, y_4^0)$	$\sum_{i,j} c_{ij} x_{ij}^0 + \sum_i f_i y_i^0$	$z \geq (\cdot) + (\cdot) y_1 + (\cdot) y_2 + (\cdot) y_3 + (\cdot) y_4 \quad z^0$
(0100)	33	26 -5 7 -1 -1 19
(1011)	27	6 7 1 7 7 20
(1001)	26	12 7 1 1 7 20
(1010)	28	14 7 -1 7 -1 21
(1000)	57	50 7 -29 -29 -31 24
(0011)	38	24 -11 -5 7 7 26
(1001)	26	

6.3.5. Вычислительный опыт

Базби и др. [15] применяли алгоритм Бендерса для многих задач распределения с кусочно-линейными и даже, возможно, разрывными функциями затрат. Задача состояла в том, чтобы произвести и перевезти единственный продукт из m предприятий — источников к n потребителям при минимальных затратах. Потребительский спрос предполагается известным, а мощности предприятий заданными. Транспортные затраты предполагаются линейными, однако затраты на производство считаются кусочно-линейными на некоторых предприятиях. Эта задача представляет собой несколько упрощенный вариант задачи, описанной в § 1.8, в которой не присутствовали склады, и некоторое обобщение задачи из 6.3.3 путем введения кусочно-линейных функций затрат.

Такие задачи могут быть решены путем решения конечного числа стандартных транспортных задач. При этом просто принимают наклон каждого линейного сегмента функции затрат в качестве общего коэффициента затрат для предприятия. Оптимальное решение исходной задачи может быть получено путем перебора всех возможных различных комбинаций таких сегментов (цен) и решения транспортной задачи для каждой комбинации. Пусть имеется N_T комбинаций. Если, например, характеристика одного предприятия имеет три участка линейности, а второго имеет четыре участка, то требуется решить $N_T = 12$ отдельных транспортных задач.

Задача может быть сформулирована как частично-целочисленная линейная задача (см. § 1.8), для которой приложим алгоритм Бендерса. Этот алгоритм решает также конечное число транспортных проблем, причем каждый раз решается новая целочисленная задача. Отношение числа N_a таких задач, которое требуется решать при применении этого алгоритма, к числу задач, возникающих при полном переборе, N_T , может рассматриваться, как мера эффективности алгоритма.

При вычислениях, произведенных Базби и другими [15], решалось 135 задач описанного выше типа. Пусть m — число предприятий, n — число потребителей, m_p — число предприятий с кусочно-линейными функциями затрат, r — полное число участков линейности в характеристиках всех m_p предприятий. В решенных задачах значения указанных выше параметров изменялись в следующих пределах:

$$7 \leq m \leq 19, \quad 7 \leq n \leq 30, \quad 2 \leq m_p \leq 6, \quad 5 \leq r \leq 24.$$

Другие данные, например, коэффициенты транспортных затрат, мощности предприятий и т. д., генерировались случайно, равномерно, выбираясь из фиксированного диапазона. Результаты показаны в таблице 6.3.2, где число альтернатив означает

Таблица 6.3.2

Диапазон изменения возможных альтернатив	Число проб	Диапазон изменения опробованных альтернатив	Среднее число опробованных альтернатив	Диапазон изменения эффективности (%)	Средняя эффективность (%)
0—20	14	2—12	5,4	13—100	43
21—40	18	2—20	7,8	6,3—56	32
41—60	16	2—22	7,6	4,4—46	16
61—100	15	2—23	12,7	2,1—32	16
101—150	15	2—30	12,0	1,4—21	9,1
151—300	19	2—45	14,6	0,93—23	6,7
300—600	20	2—78	17,7	0,37—20	3,9
600—1000	10	2—48	15,5	0,2—5,0	1,8
1000—2000	5	5—52	37,4	0,3—3,6	2,3
2000—4000	3	3—24	11,7	0,1—0,8	0,44

число различных возможных вариантов значений целочисленных переменных, каждый из которых приводит к решению одной из N_T транспортных задач. Эффективность равна $100 N_a/N_T$.

Легко видеть, что эффективность существенно улучшается с ростом N_T , т. е. алгоритм становится более эффективным с ростом сложности задачи. В [15] сообщается также, что в задачах, где многие из возможных вариантов предприятий являются неэкономичными, т. е. требуют больших капиталовложений, они быстро исключаются из рассмотрения с помощью ограничений, вводимых в целочисленную задачу. Таким образом, в таких задачах рассматривается только малое число альтернатив. В 135 решенных задач число допустимых альтернатив сокращалось в среднем приблизительно в 1,7 раза на каждом шаге. К аналогичным выводам приводит рассмотрение данных по отдельным задачам, приведенных Балинским в [10] и указанных в таблице 6.3.3. Однако, как отмечено в [10], действительная полезность алгоритма Бендерса в частично-целочисленных задачах сильно

Таблица 6.3.3

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
m_0	3	4	4	5	5	6	6	6
r	7	19	20	17	22	16	22	23
m	10	13	13	16	16	19	19	19
n	14	18	19	24	26	22	28	30
N_a	12	14	21	3	43	38	52	8
N_T	12	450	405	432	1200	288	1944	2592
$100 N_a/N_T$	100	3,1	5,2	0,7	3,6	13	2,7	0,3

зависит от эффективности алгоритма целочисленного программирования, используемого при решении задачи $MP2$. При решении указанных выше задач использовался один из вариантов схем частичного перебора.

Другой алгоритм в связи с алгоритмом Бендерса был предложен Зойтендейком [16]. Однако в настоящее время неизвестно, насколько хорошо та или другая схемы могут решать целочисленные задачи типа $MP2$.

Задачи

1. Покажите, что если $b_i(y)$ выпукла, то множество S в (6.2.1-3) выпукло, и все $\psi(y)$ выпуклы на S .

2. Докажите необходимость утверждения теоремы 6.2.1-2.

3. Покажите, что если $h_i(y)$ в (6.2.1-44) выпукла для всех i , то $y^1 \in S$ и $\psi(y^1) < \psi(y_0)$.

4. Покажите, что допустима иная формулировка задачи $P2$ в § 6.3.1 путем, замены целевой функции на $z + f(y)$ и ограничений на $z \geq (b - F(y))' u_i^p$, $i = 1, \dots, n_p$. Постройте итеративный алгоритм, базирующийся на этой формулировке, и сравните его с алгоритмом § 6.3.1. Запись $P2$ в такой форме позволяет поместить $f(y)$ в ее «естественное» положение в целевой функции, а не в ограничениях.

5. Рассмотрите возможность опускания ограничений в $MP2$, когда они не являются связующими в точке текущего оптимума. Задайте условия на F , f и S и стратегию изъятия ограничений, при которой гарантируется конечная сходимость.

6. Покажите, как построить двойственный крайний луч с помощью конечной таблицы двойственного симплекс-метода, если алгоритм показывает, что исходная задача (6.3.1-15) недопустима. Предложите также способ, при котором более, чем один такой луч, может быть генерирован при каждом применении двойственного симплекс-метода. Как это может быть использовано в алгоритме Бендерса и какие возможны преимущества?

7. Предположим, что симплекс-метод используется для решения задачи (6.3.1-15), и что эта задача является недопустимой для текущего вектора y . Покажите, как может быть найден двойственный крайний луч с помощью конечной фазы симплексной таблицы.

8. Преобразуйте задачу $P1$ к эквивалентной форме

$$\begin{aligned} \min & z, \\ Ax_i + F(y) & \geq b, \\ z & \geq c'x + f(y), \\ x & \geq 0, \quad y \in S. \end{aligned}$$

Примените лемму Фаркаша к этой системе и получите задачу $P2$, а также утверждение теоремы 6.3.1-1 (см. [8] по поводу выводов такого типа).

9. Как обобщение алгоритма Бендерса, рассмотрите задачу:

$$\begin{aligned} \min & (g(x) + f(y)), \\ G(x) + F(y) & \geq b, \\ x & \in S_1, \quad y \in S_2, \end{aligned}$$

где g — выпуклая функция, G — вектор таких функций, S_1 — выпуклое множество, f и F не обязательно выпуклы, а S_2 — произвольное подмножество E^p . Это нелинейная задача, в которой выпуклые и невыпуклые части разделимы.

Попытайтесь распространить алгоритм Бендерса на этот класс задач, применяя теорию двойственности нелинейного программирования.

10. Рассмотрите применение различных алгоритмов целочисленного программирования для задачи $MP2$ в случае, когда исходная задача $P1$ является задачей частично-целочисленного линейного программирования. Какой из этих алгоритмов является наиболее подходящим? Возможно ли построить специальный алгоритм, который будет наиболее эффективным?

11. Рассмотрите применение алгоритма Бендерса в случае, когда A является блочно-диагональной, т. е. имеет структуру

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_m \end{bmatrix}.$$

Покажите, что в этом случае пара линейных программ (6.3.1-15) — (6.3.1-16) распадается на m малых подзадач. Покажите также, что существует возможность добавления более, чем одного ограничения к $MP2$ на каждом шаге. Как изменить критерий оптимальности в этом случае?

Идея добавления более чем одного ограничения соответствует сокращенной координирующей задаче в принципе декомпозиции Данцига — Вулфа, в которой более чем одно из предложений подзадач может входить в базис. Можно ли ожидать увеличения эффективности алгоритма Бендерса в этом случае? Почему?

12. Постройте двойственную к задаче, использованной в качестве примера в § 2.5. Постройте проблему $P2$ и двойственную пару подзадач. Установите двойственную связь между $P2$ и координирующей программой в § 2.5.

13. Сравните процедуру Бендерса с процедурой расчленения Розена из главы V, если обе они применяются к линейным задачам с угловой структурой матрицы.

14. Рассмотрите многопродуктовую задачу распределения программы с «непрерывной» настройкой типа, обсуждавшегося в § 3.2, но такую, в которой имеется три вида продуктов, каждый из которых относится к своему типу машин. Машины типа j , не используемые для выпуска продукта типа j , могут быть использованы для выпуска продукта i , если $i < j$. Сформулируйте эту задачу как задачу математического программирования, имеющую структуру из трех отдельных многопродуктовых задач распределения, связанных некоторыми переменными. Примените алгоритм Бендерса к этой проблеме. Постройте $P2$ и двойственную к ней и покажите, что подзадачи являются обычными задачами распределения программы с переменными правыми частями. Могут ли эти подзадачи эффективно решаться с помощью модификаций двойственного симплекса-метода или должна использоваться процедура § 3.2, и почему?

15. Рассмотрите задачу того же типа, что была указана в пункте 14, но для случая, когда для выпуска продуктов i -го типа могут использоваться машины j -го типа, только если $i = j - 1$. Повторите анализ пункта 14 для этого случая.

ЛИТЕРАТУРА

1. J. B. Rosen, Convex Partition Programming, in «Recent Advances in Mathematical Programming», R. L. Graves and P. Wolfe, eds., McGraw-Hill, Inc., New York, 1963, pp. 159—176.
2. H. W. Kuhn and A. W. Tucker, Nonlinear Programming, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, Jerzy Neuman, ed., University of California Press, Berkeley, 1950, pp. 481—492.

3. S. Karlin, *Mathematical Methods and Theory in Games, Programming and Economics*, vol. 1, Addison — Wesley Publishing Company, Inc., Reading, Mass., 1959, chap. 7. [Русский перевод: С. Карлин, Математические методы в теории игр, программировании и экономике, «Мир», М., 1964.]
4. J. B. Rosen, *The Gradient Projection Method for Nonlinear Programming*, Parts I and II. *J. Soc. Ind. Appl. Math.*, 8, 1960, pp. 181—217; 9, 1961, pp. 514—532.
5. D. Goldfarb, *Extension of Davidons' Variable Metric Method to Maximization Under Linear Inequality and Equality Constraints*, *SIAM. J. Appl. Math.*, 17, No. 4, 1969, pp. 739—764.
6. J. B. Rosen and J. C. Ornea, *Solution of Nonlinear Programming Problems by Partitioning*, *Management Sci.*, 10, No. 1, 1963, pp. 160—173.
7. J. C. Ornea and G. G. Eldredge, *Nonlinear Partitioned Models for Plant Scheduling and Economic Evaluation*. *Proc. A. I. C. H. E. — Ind. Chem. Eng. Joint Meeting*, London, June 13—17, 1965, *Symposium on the Application of Mathematical Models in Chemical Engineering, Research, Design, and Production*, pp. 101—112.
8. J. F. Benders, *Partitioning Procedures for Solving Mixed Variables Programming Problems*. *Numerische Mathematik*, 4, 1962, pp. 238—252.
9. A. W. Tucker, «Dual Systems of Homogeneous Linear Relations», in «Linear Inequalities and Related Systems» (*Annals of Mathematics Studies*, 38), Princeton University Press, Princeton, N. J., 1956, pp. 3—18. [Русский перевод: А. У. Таккер, Двойственные системы однородных линейных соотношений, в сб. «Линейные неравенства и смежные вопросы», ИЛ, М., 1959.]
10. M. Balinski, *Integer Programming: Methods, Uses, Computation*, *Management Sci.*, 12, No. 3, 1965, pp. 253—313.
11. A. V. Fiacco and G. P. McCormick, *The Sequential Unconstrained Minimization Technique for Nonlinear Programming. A Primal — Dual Method*. *Management Sci.*, 10, Jan. 1964, pp. 360—366.
12. A. V. Fiacco and G. P. McCormick, *Sequential Unconstrained Minimization Techniques for Nonlinear Programming*, John Wiley and Sons, Inc., New York, 1969. [Русский перевод: А. Ф и а к к о, Г. М а к — К о р м и к, Нелинейное программирование: методы последовательной безусловной минимизации, «Мир», М., 1972.]
13. J. E. Kelley, *The Cutting Plane Method for Solving Convex Programs*. *J. Soc. Ind. Appl. Math.*, 8, 1960, pp. 703—712.
14. M. L. Balinski and P. Wolfe, *On Benders Decomposition and a Plant Location Problem*, *Mathematica Working Paper ARO — 27*, 1963.
15. B. R. Buzby, B. J. Stone, and R. L. Taylor, *Computational Experience with a Nonlinear Distribution Problem*, private communication, Jan. 1965.
16. G. Zoutendijk, *Enumeration Algorithms for the Pure and Mixed Integer Programming Problem*. *International Symposium on Mathematical Programming*, Princeton University, Aug. 14—18, 1967.

ГЛАВА VII

ДВОЙСТВЕННОСТЬ И ДЕКОМПОЗИЦИЯ В МАТЕМАТИЧЕСКОМ ПРОГРАММИРОВАНИИ

§ 7.1. ВВЕДЕНИЕ

Во многих алгоритмах, обсуждавшихся в предшествовавших главах, цены использовались в качестве координирующего механизма. Выдающимся примером такого рода является принцип декомпозиции Данцига — Вулфа (гл. II). Там задача включала в себя ряд независимых линейных подзадач, связанных ограничениями на распределяемые ресурсы. Эта задача разделялась путем установления цен на общие ресурсы и добавления затрат на эти ресурсы к целевой функции каждой подзадачи. Цены вычислялись как двойственные переменные в координирующей задаче. При соответствующем изменении этих цен подзадачи вырабатывали предложения для координирующей задачи, которые, будучи скомбинированы с предшествующими предложениями, позволяли сокращать полные затраты. Было отмечено в § 2.4, что в этой схеме координирующая задача (т. е. «центральное агентство») обладает полной ответственностью за принятие решений. Оно определяет оптимальное решение, как выпуклую комбинацию предложений, а также определяет новые цены, направляемые к подзадачам. Подзадачи только вырабатывают предложения, и данное предложение не обязательно должно быть включено в конечное решение. Таким образом, процесс принятия решений только частично децентрализован.

Принцип декомпозиции был распространен на нелинейные задачи в § 3.4.2. Там подзадачи включали минимизацию нелинейной выпуклой функции на выпуклых множествах. Ограничения на общие ресурсы сохранялись линейными. При использовании линеаризации на сетке был развит аналог принципа декомпозиции. Подзадачи оказывались задачами нелинейного выпуклого программирования, и алгоритм уже не сходил к конечному числу итераций. Как и в полностью линейном случае, подпроблемы играют ограниченную роль в принятии решений.

Идея децентрализации при помощи системы цен изучается далее в этом параграфе. Рассматривается общий класс нелинейных задач. Показано, что если функция Лагранжа такой задачи аддитивно-сепарабельна, то может быть произведена декомпозиция с помощью цен при некоторых дополнительных условиях. Как и ранее, цены представляют собой множители Лагранжа и являются решениями двойственной задачи. Однако имеется важное различие между схемами, изучаемыми здесь, и теми, которые рассматривались ранее. Здесь мы пытаемся осуществить полную децентрализацию в принятии решений. В данном случае это означает, что оптимальные значения решений должны быть получены непосредственно при решении подзадач. Координатор верхнего уровня несет ответственность только за значения цен. Как можно заранее ожидать, такая схема не будет работоспособна во всех случаях. Критическим моментом является вопрос о существовании седловой точки функции Лагранжа в задаче. Существование может быть гарантировано, грубо говоря, только для выпуклых задач. Однако полезные результаты иногда могут быть получены и для задач, не обладающих требуемой выпуклостью.

§ 7.2. ДЕКОМПОЗИЦИЯ, ИСПОЛЬЗУЮЩАЯ МЕХАНИЗМ ЦЕН

Для того чтобы ввести основные идеи, рассмотрим следующий простой пример. Пусть имеется n предприятий. Действия каждого предприятия сопровождаются затратами и используют некоторое количество единственного ресурса. Пусть $c_i(x_i)$ — затраты предприятия i при интенсивности работы на уровне x_i , а a_i — величина ресурса, используемая на единицу x_i и предполагаемая постоянной, b — доступное количество ресурса. Проблема выбора интенсивности работы всех предприятий при минимальных общих затратах, если они не используют больше никаких ресурсов, формулируется в виде

$$\min z = \sum_{i=1}^n c_i(x_i), \quad (1)$$

$$\sum_{i=1}^n a_i x_i \leq b, \quad x_i \geq 0. \quad (2)$$

Рассмотрим децентрализованную процедуру решения этой задачи с использованием системы цен в качестве координирующего механизма. Пусть $u \geq 0$ — цена ресурса, размерность которой (единицы z)/(единицы b). После задания u , естественным методом декомпозиции является привязка к каждому предприятию «менеджера», который полностью ответствен за выбор

интенсивности его работы x_i . Менеджер может покупать любое желаемое количество ресурса, но должен платить за то, что он использует. Разумный менеджер выбрал бы уровень интенсивности, который минимизирует полные затраты на операции, т. е. прямые затраты плюс затраты на ресурсы. Это приводит к подзадаче вида

$$\min_{x_i \geq 0} \tilde{f}_i(x_i, u) = c_i(x_i) + u(a_i x_i), \quad \left. \vphantom{\min} \right\} \text{ подзадача } i \quad (3)$$

Для фиксированного u все такие подзадачи независимы друг от друга. Предположим (временно), что все они имеют единственное конечное оптимальное решение для любой величины $u \geq 0$ и обозначим это решение $x_i(u)$. Чувствуется, что при увеличении u менеджеры будут стремиться использовать меньше ресурса, и наоборот, так что u оказывается средством координации их действий. Возникают естественные вопросы:

1. Существует ли значение u , равное u^0 и называемое оптимальным, для которого решения подзадач $\{x_1(u^0), \dots, x_n(u^0)\}$ решают исходную задачу (1) — (2)?

2. Если ответ на первый вопрос положителен, то как можно проверить оптимальность решения $\{x_1(u^0), \dots, x_n(u^0)\}$?

3. Как может быть вычислено оптимальное значение u , если оно существует?

В остающейся части этой главы будут даны частичные ответы на эти вопросы. Ответы на вопросы 1 и 2 связаны с изучением седловых точек функции Лагранжа, в то время как ответ на вопрос 3 ставит во главу угла понятие двойственности. Заметим, что подзадачи должны непосредственно приводить к оптимальному решению. Таким образом, процедура принятия решений является полностью децентрализованной.

Ключ к вопросу о существовании оптимальных цен дается теоремой Куна — Таккера (см. [1]). Пусть функции c_i являются выпуклыми и дифференцируемыми. Образует функцию Лагранжа

$$L(x, u) = \sum_i c_i(x_i) + u \left(\sum_i a_i x_i - b \right), \quad (4)$$

$$L(x, u) = \sum_i \{c_i(x_i) + u a_i x_i\} - ub. \quad (5)$$

Сравнивая (5) и (3), мы убеждаемся, что функция Лагранжа является аддитивно-сепарабельной и может быть записана в виде

$$L(x, u) = \sum_i \tilde{f}_i(x_i, u) - ub, \quad (6)$$

где f_i есть целевая функция в задачах (3). В силу теоремы Куна — Таккера, вектор $x^0 = (x_1^0, \dots, x_n^0)$ является решением задачи (1) — (2), если и только если существует множитель $u^0 \geq 0$ такой, что x^0 удовлетворяет условиям

$$\frac{\partial L}{\partial x_i} = \frac{\partial \tilde{f}_i(x_i, u^0)}{\partial x_i} \geq 0, \quad i = 1, \dots, n, \quad (7)$$

$$u^0 \left(\sum_i a_i x_i - b \right) = 0, \quad \sum_i a_i x_i \leq b, \quad (8)$$

$$x_i \frac{\partial \tilde{f}_i(x_i, u^0)}{\partial x_i} = 0, \quad x_i \geq 0, \quad i = 1, \dots, n. \quad (9)$$

Так как f_i выпукла по x_i для любого $u \geq 0$, то (7) и (9) являются необходимыми и достаточными для того, чтобы x_i^0 минимизировало $\tilde{f}_i(x_i, u^0)$ при условиях $x_i \geq 0$, т. е. чтобы x_i^0 решало i -ю подзадачу (3) при $u = u^0$. Таким образом, множители Лагранжа u^0 являются желаемой ценой. Условие (8) говорит, что u^0 должно быть достаточно большим для того, чтобы ограничить полное потребление ресурсов доступной величиной и что если оптимальное потребление меньше, чем b единиц, то цена u^0 равна 0. Следовательно, ресурс, имеющийся в избытке, не должен ничего стоить.

§ 7.3. СЕДЛОВЫЕ ТОЧКИ ФУНКЦИЙ ЛАГРАНЖА

7.3.1. Основные теоремы

В предшествующем обсуждении теорема Куна — Таккера использовалась для обоснования существования оптимальных цен. Условия Куна — Таккера являются условиями стационарности, в то время как подзадачи (3) сформулированы как задачи минимизации. Стационарность и минимальность эквивалентны только для выпуклых дифференцируемых функций, определенных на выпуклых множествах. Однако во многих ситуациях эти требования выпуклости и дифференцируемости не выполняются, например, в проблемах, включающих оптимизацию на конечных множествах.

Множители Лагранжа, однако, могут быть использованы и в таких задачах. Для того чтобы показать это, рассмотрим некоторые необходимые теоремы, касающиеся седловых точек функции Лагранжа. Эти результаты, необходимые определения и теоремы здесь устанавливаются, но не доказываются. Рассмотрим задачу математического программирования вида

$$\min f(x) \quad (1)$$

при ограничениях

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

$$x \in S. \quad (3)$$

Будем называть ее *исходной задачей* или *прямой задачей*. Здесь x — n -вектор, S —произвольное подмножество E^n , f и g_i —произвольные вещественнозначные функции, определенные на S . Введем функцию Лагранжа, ассоциированную с этой задачей:

$$L(x, u) = f(x) + \sum_{i=1}^m u_i g_i(x), \quad u_i \geq 0. \quad (4)$$

Определение. Точка (x^0, u^0) , где $u^0 \geq 0$, $x^0 \in S$, называется *седловой точкой* L , если она удовлетворяет условиям

$$L(x^0, u^0) \leq L(x, u^0), \quad \forall x \in S, \quad (5)$$

$$L(x^0, u^0) \geq L(x^0, u), \quad \forall u \geq 0. \quad (6)$$

Следующая теорема дает необходимые и достаточные условия для выделения седловой точки L .

Теорема 1. Пусть $u^0 \geq 0$ и $x^0 \in S$. Тогда (x^0, u^0) является седловой точкой L , если и только если:

$$a) \quad x^0 \text{ минимизирует } L(x, u^0) \text{ на } S, \quad (7)$$

$$b) \quad g_i(x^0) \leq 0, \quad i = 1, \dots, m, \quad (8)$$

$$c) \quad u_i^0 g_i(x^0) = 0, \quad i = 1, \dots, m. \quad (9)$$

Полезность седловой точки вытекает из следующей теоремы.

Теорема 2 (достаточность выделения седловой точки). Если (x^0, u^0) —седловая точка L , то x^0 есть решение прямой задачи (1)–(3).

Для того чтобы связать теоремы 1 и 2 с нашей предшествующей дискуссией, касавшейся проблемы декомпозиции, применим эти теоремы к задаче о распределении ресурсов, рассмотренной ранее. Чтобы записать ее в общей форме (1)–(3), положим

$$S = \{x/x \geq 0\}. \quad (10)$$

Тогда задача записывается в виде

$$\min \sum_{i=1}^n c_i(x_i) \quad (11)$$

при ограничениях

$$\sum_{i=1}^n a_i x_i - b \leq 0, \quad (12)$$

$$x \in S. \quad (13)$$

Функция Лагранжа этой задачи

$$L(x, u) = \sum_{i=1}^n (c_i(x_i) + u a_i x_i) - ub. \quad (14)$$

Условие (а) теоремы 1 требует, чтобы $L(x, u^0)$ минимизировалась на S для получения x^0 . Так как L аддитивно-сепарабельна по x_i при фиксированном u , имеем

$$\min_{x_i \geq 0, \dots, x_n \geq 0} \sum_{i=1}^n (c_i(x_i) + u a_i x_i) = \sum_{i=1}^n \{ \min_{x_i \geq 0} (c_i(x_i) + u a_i x_i) \}. \quad (15)$$

Величина

$$\min_{x_i \geq 0} (c_i(x_i) + u a_i x_i) \quad (16)$$

является оптимальным значением целевой функции i -й подзадачи (7.2—3). Таким образом, эти подзадачи возникают математически при попытке удовлетворить первому из необходимых и достаточных условий для седловой точки. Второе из этих условий (условие (b) в теореме 1) говорит, что значение оптимальной цены u^0 должно быть таким, чтобы решения подзадач совместно удовлетворяли ограничениям по наличию ресурсов. Условие (с) повторяет ранее сделанное замечание, что ресурс, имеющийся в избытке, должен иметь цену, равную нулю.

Процесс минимизации лагранжиана $L(x, u)$ на $x \in S$ при фиксированном u приводит к ряду независимых подзадач, когда L — аддитивно-сепарабельна по x при фиксированном u и когда S может быть записано в виде декартова произведения. В прямой задаче (1)—(3) вектор x может быть расчленен следующим образом:

$$x = (x_1 | x_2 | \dots | x_p), \quad p \leq n. \quad (17)$$

Предположим также, что f , S и вектор функций g в ограничениях могут быть записаны в виде

$$f(x) = \sum_{i=1}^p f_i(x_i), \quad (18)$$

$$g(x) = \sum_{i=1}^p g^i(x_i), \quad (19)$$

$$S = S_1 \times S_2 \times \dots \times S_p, \quad (20)$$

где $\times$ означает декартово умножение, и каждая из g^i есть вектор-функция размерности m . В этом важном специальном случае

$$L(x, u) = \sum_i f_i(x_i) + u \sum_i g^i(x_i), \quad (21)$$

$$L(x, u) = \sum_i (f_i(x_i) + u g^i(x_i)), \quad (22)$$

так что лагранжиан аддитивно-сепарабелен по x_i и

$$\min_{x_1 \in S_1, \dots, x_p \in S_p} L(x, u) = \sum_i \min_{x_i \in S_i} (f_i(x_i) + u g^i(x_i)). \quad (23)$$

Таким образом, лагранжиан может быть минимизирован путем решения p подзадач

$$\left. \begin{array}{l} \min (f_i(x_i) + u g^i(x_i)) \\ \text{при ограничениях} \\ x_i \in S_i. \end{array} \right\} \quad (24)$$

Эти подзадачи имеют ту же самую стоимостную интерпретацию, как и в приведенном ранее примере, где u_k — цена ресурса, доступность которого ограничена k -м ограничением.

При выделении в исходных ограничениях тех, которые определяют область S , и тех, которые явно записываются, как $g_i \leq 0$, а следовательно, входят в лагранжиан, мы должны стремиться к достижению сепарабельности L и S , т. е. этот выбор должен быть сделан так, чтобы, если возможно, условия (18)–(20) выполнялись для некоторого разбиения вектора x на подвекторы $(x_1 | x_2 | \dots | x_p)$. Все предшествующие результаты легко распространяются на случай, когда в числе ограничений есть равенства. Если k -е ограничение имеет вид $g_k(x) = 0$, то множитель u_k , соответствующий этому ограничению, может иметь любой знак, и условие (с) теоремы 1 окажется излишним, поскольку $g_k(x^0) = 0$.

7.3.2. Теорема Эверетта

Эверетт [3] показал, что, если удается выбрать некоторые множители $u \geq 0$, для которых $\min_{x \in S} L(x, u)$ существует, то любой вектор $x(u)$, который минимизирует $L(x, u)$ представляет собой решение задачи, тесно связанной с исходной прямой задачей.

Этот результат состоит в следующем:

Теорема 3 (Эверетт). Если $x(u)$ — решение лагранжевой проблемы

$$\left. \begin{array}{l} \min L(x, u) \\ \text{при ограничениях} \\ x \in S, \end{array} \right\} \quad (25)$$

где $u \geq 0$, то $x(u)$ представляет собой решение модифицированной исходной задачи

$$\min f(x) \quad (26)$$

при ограничениях

$$g_i(x) \leq u_i, \quad i = 1, \dots, m, \quad x \in S, \quad (27)$$

где

$$u_i = g_i(x(u)), \text{ если } u_i > 0; \quad u_i \geq g_i(x(u)), \text{ если } u_i = 0. \quad (28)$$

Модифицированная прямая задача имеет в правой части ограничений константы, равные или большие $g_i(x(u))$. Пусть ограничения вида $g_i \leq 0$ рассматриваются как отражающие ограниченность ресурсов, а u_i — как цены этих ресурсов. Тогда смысл теоремы заключается в том, что любой вектор, который минимизирует $L(x, u)$, решает исходную задачу, в которой допускается использование не большего количества имеющих цены ресурсов, чем затрачивается при выборе самого вектора $x(u)$, в то время как наличность свободных, не имеющих цены ресурсов, по крайней мере так велика, как это требуется при использовании самого вектора. Следовательно, множители u преобразуют исходную задачу минимизации с ограничениями в задачу минимизации без ограничений (если не считать ограничений вида $x \in S$), но уже для функции Лагранжа.

Как было указано раньше, это преобразование особенно привлекательно, если f , g_i и S имеют сепарабельную структуру, удовлетворяющую условиям (18)–(20). Тогда лагранжева задача распадается на меньшие независимые подзадачи (24). Отметим, что в теореме не налагается каких-либо ограничений на поведение функций f , g_i или на множество S . Доказательство теоремы 3 следует немедленно из того факта, что, в силу теоремы 1, $x(u)$ и u задают седловую точку лагранжиана

$$L = f(x) + \sum_{i=1}^m u_i (g_i(x) - u_i). \quad (29)$$

Теорема 3 может быть практически использована следующим образом. Можно выбрать некоторые значения множителей $u \geq 0$, решить лагранжеву проблему, а затем проверить, найдутся ли

значения констант y_i , при которых решается модифицированная исходная задача. Такой путь может оказаться ценным, если требуется решить целый набор исходных задач с различными правыми частями. Конечно, нельзя гарантировать, что эта процедура может генерировать любые векторы правых частей. Векторы y , для которых ассоциированные задачи не могут быть разрешены, названы в [3] *зазорами* (gaps). Геометрическое описание этих зазоров дано в § 7.4.

Хотя значения правых частей, являющиеся следствием выбора значений u , не могут быть предсказаны заранее, имеются некоторые результаты, которые облегчают решение. Так как u_k интерпретируется как цена ресурса k , то чувствуется, что если u_k увеличивается, в то время как все другие цены остаются фиксированными, то используемое количество k -го ресурса будет уменьшаться. Это справедливо, как показывает следующая теорема.

Теорема 4 (Эверетт). Пусть u^1, u^2 — неотрицательные m -векторы, такие, что

$$u_k^2 > u_k^1, \quad u_j^2 = u_j^1, \quad j \neq k. \quad (30)$$

Если $x(u^1)$ — решение лагранжевой проблемы, в которой $u = u^1$, то

$$g_k(x(u^2)) \leq g_k(x(u^1)). \quad (31)$$

Другой результат, полезный для вычислений, относится к почти оптимальным решениям лагранжевой проблемы. В случаях, когда эта проблема должна решаться численно, точные решения находятся редко. Следующий результат показывает, что «хорошее» решение лагранжевой проблемы является «хорошим» решением модифицированной прямой задачи (26) — (28).

Теорема 5 (Эверетт). Пусть x^* дает решение лагранжевой проблемы с точностью до ε , т. е.

$$L(x^*, u) \leq L(x, u) + \varepsilon \text{ для всех } x \in S.$$

Тогда x^* решает модифицированную прямую задачу также с точностью до ε .

Доказательство теорем 4 и 5 опущено и указано в качестве упражнений в конце этой главы.

7.3.3. Применение к линейным целочисленным задачам

Рассмотрим линейную целочисленную задачу, именуемую далее $IP(b)$:

$$\min \sum_{j=1}^n c_j x_j \quad (32)$$

при ограничениях

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i=1, \dots, m, \quad (33)$$

$$x_j = 0 \text{ или } 1, \quad j=1, \dots, n. \quad (34)$$

Так как задача сепарабельна, то естественно попытаться применить метод Лагранжа. Функция Лагранжа

$$\begin{aligned} L(x, u) &= \sum_{j=1}^n c_j x_j + \sum_{i=1}^m u_i \left(\sum_{j=1}^n a_{ij} x_j - b_i \right) = \\ &= \sum_{j=1}^n \left(c_j + \sum_{i=1}^m u_i a_{ij} \right) x_j - \sum_{i=1}^m u_i b_i. \end{aligned} \quad (35)$$

При введении обозначений

$$\beta_j = c_j + \sum_{i=1}^m u_i a_{ij}$$

лагранжева задача сводится к минимизации $\sum_{j=1}^n \beta_j x_j$ при ограничениях $x_j = 0$ или 1. Решение очевидно:

$$\beta_j > 0 \Rightarrow x_j = 0, \quad \beta_j < 0 \Rightarrow x_j = 1, \quad \beta_j = 0 \Rightarrow x_j = 0 \text{ или } 1.$$

Если обозначить такое решение через $x(u)$, то, в силу теоремы 3, $x(u)$ является решением и модифицированной задачи $IP(y(u))$, где коэффициенты в правых частях ограничений $y(u)$ строятся в силу (28). Как показано в [32], $x(u)$ является также оптимальным решением задачи $LP(y(u))$, отличающейся от задачи (32)–(34) заменой правых частей на $y(u)$, а целочисленных ограничений (34) — на линейные вида $0 \leq x_j \leq 1$. Этот результат является следствием того, что пара $(x(u), u)$ задает седловую точку для $L(x, u)$, если в (35) заменить b_i на $y_i(u)$, причем, x_j ограничено значениями 0 или 1. Если эти ограничения заменить на $0 \leq x_j \leq 1$, то минимум $L(x, u)$ вновь будет достигаться при $x = x(u)$. Таким образом, $(x(u), u)$ вновь окажется седловой точкой, и это может быть справедливо, если и только если $x(u)$ является решением $LP(y(u))$, а u — решением двойственной к ней. Это означает, что любая целочисленная линейная задача, которая может быть решена через решение лагранжевой задачи, может быть также разрешена симплекс-методом.

Однако, как указывалось ранее, даже если лагранжева процедура не может дать решения непосредственно рассматриваемой задачи, она все же может решить задачу, в которой правые

части $y(u)$ достаточно близки к правым частям исходной, и это может оказаться полезным результатом. Последнее часто имеет место и при решении задач распределения капиталовложений, рассмотренных, например, в [31]. В этих задачах максимизируется целевая функция, причем b_i представляют собой объемы капиталовложений, доступные на период i , a_{ij} — количества капиталовложений, используемые для проекта j в периоде i , c_j представляют собой значения всех будущих доходов от проекта j , и наконец, $x_j = 1$ означает, что проект j принимается для использования. Так как b_i редко известно точно, то решения, получаемые для $y(u)$, близких к b , могут быть столь же хороши, как и оптимальные решения исходной задачи.

Теорема 5 полезна для получения таких решений, поскольку она позволяет генерировать оптимальные решения, не получаемые непосредственно. Предположим, что все c_j целочисленные и предположим, что дано индексное множество J такое, что

$$\sum_{j \in J} |\beta_j| < 1. \text{ Положим}$$

$$x_j^* = 1 - x_j(u), \quad j \in J,$$

$$x_j^* = x_j(u), \quad j \notin J.$$

Тогда

$$L(x^*, u) - L(x(u), u) = \sum_{j \in J} |\beta_j| = \varepsilon,$$

так что x^* минимизирует $L(x, u)$ с точностью до ε . В силу теоремы 5, x^* дает ε -оптимальное решение $IP(y^*)$, где

$$y_i^* = \sum_{j=1}^n a_{ij} x_j^*, \text{ если } u_i > 0,$$

$$y_i^* \geq \sum_{j=1}^n a_{ij} x_j^*, \text{ если } u_i = 0.$$

Но $\varepsilon < 1$ и, так как c_j целочисленные, целевая функция $IP(y^*)$ может принимать только целочисленные значения. Таким образом, x^* решает задачу $IP(y^*)$. Векторы y^* , генерируемые таким путем, могут лежать в «зазорах», так что этот выбор действительно расширяет метод Лагранжа.

Существование седловых точек. Вопросы, касающиеся существования оптимальных множителей, и определение таких множителей, если они существуют, частично решаются предшествующими результатами. Любой вектор u^0 , который является u -компонентой седловой точки L , может служить в качестве желаемого вектора-множителя, и условия (а) — (с) теоремы 1

показывают, что такая точка может быть обнаружена. Следующая теорема относится к существованию седловой точки для задач выпуклого программирования.

Теорема 6 (Карлин [4]). Пусть S — выпуклое подмножество E^n , f — выпуклая функция, определенная на S , и $g(x) = \{g_1(x), \dots, g_m(x)\}$ — вектор выпуклых функций, определенных на S . Предположим, что существует точка $x \in S$, такая, что $g(x) < 0$. Если x^0 точка, в которой $f(x)$ достигает своего минимума при ограничениях $g(x) \leq 0$, $x \in S$, то существует вектор множителей $u^0 \geq 0$, такой, что (x^0, u^0) представляет собой седловую точку

$$L(x, u) = f(x) + ug(x)$$

при ограничениях $x \in S$, $u \geq 0$. Обратно, если (x^0, u^0) есть седловая точка $L(x, u)$ при указанных ограничениях, то x^0 минимизирует $f(x)$ при ограничениях $g(x) \leq 0$, $x \in S$.

Доказательство можно найти в [4].

§ 7.4. МИНИМАКСНЫЕ ДВОЙСТВЕННЫЕ ЗАДАЧИ

Хотя уже указывался ряд результатов, касающихся существования седловых точек, но ничего еще не было сказано относительно того, как находить такие точки. Для того чтобы ответить на этот вопрос, необходимо изучить поведение минимального значения $L(x, u)$ как функции u , т. е. вместо того, чтобы обращать внимание на единственное множество лагранжевых множителей u^0 , мы рассмотрим все $u \geq 0$.

Пусть

$$h(u) = \min_{x \in S} L(x, u), \quad (1)$$

$$X(u) = \{x/x \text{ минимизирует } L(x, u) \text{ на } S\}. \quad (2)$$

Функция h называется *двойственной функцией*. Ее область определения такова:

$$D = \{u/u \geq 0, \min_{x \in S} L(x, u) \text{ существует}\}, \quad (3)$$

т. е. D представляет собой множество неотрицательных векторов u , для которых $L(x, u)$ имеет конечный $\inf$ на S , достигаемый в некоторой точке S . Если для всех $u \geq 0$ $L(x, u)$ непрерывна по x для всех $x \in S$ и если S замкнуто и ограничено, то по теореме Вейерштрасса [5] имеем

$$D = (E^m)^+ = \{u/u \geq 0\}. \quad (4)$$

Однако D не всегда является выпуклой и может быть пустым множеством. Нетрудно построить примеры, где D состоит из единственной точки. Определим теперь пару двойственных задач

следующим образом:

$$\begin{array}{ll} \text{Прямая} & \text{Двойственная} \\ \min \{f(x)/g(x) \leq 0; x \in S\} \quad (5); & \max \{h(u)/u \in D\}. \quad (6) \end{array}$$

Мы назовем эту двойственную задачу *минимаксной двойственной* *), поскольку она может быть представлена в виде

$$\max_{u \in D} \min_{x \in S} L(x, u). \quad (7)$$

Легко показать, что если прямая задача является линейной, то (6) является двойственной к ней в обычном смысле. Пусть прямая задача записана в виде

$$\min c'x \quad (8)$$

при ограничениях

$$Ax \geq b, \quad x \geq 0. \quad (9)$$

Тогда

$$L(x, u) = c'x + u'(b - Ax), \quad u \geq 0, \quad (10)$$

и

$$h(u) = \min_{x \geq 0} (c'x + u'(b - Ax)), \quad (11)$$

$$h(u) = \min_{x \geq 0} (x'(c - A'u) + b'u). \quad (12)$$

Минимум существует, если и только если

$$c - A'u \geq 0, \quad (13)$$

так как в противном случае $h(u)$ может быть произвольно большим и отрицательным. Таким образом,

$$D = \{u/A'u \leq c; u \geq 0\}. \quad (14)$$

Для любого $u \in D$ минимум достигается при условии

$$x'(c - A'u) = 0, \quad (15)$$

которое выражает условие дополняющей нежесткости для линейных задач. Тогда

$$h(u) = b'u. \quad (16)$$

Задача максимизации $b'u$ на D является стандартной двойственной задачей линейного программирования для исходной (8) — (9). Если $S = E^n$ и f и g являются выпуклыми и дифференцируемыми, то двойственная (6) эквивалентна двойственной задаче по определению Вулфа [9]. Действительно, двойственная задача Вулфа состоит в следующем:

$$\max \{L(x, u) = f(x) + ug(x)\} \quad (17)$$

*) Такие двойственные задачи изучались Фалком [2], Мангасарьяном [6], Штоером [7], Рокафелларом [8] и др.

при ограничениях

$$\nabla_x L(x, u) = \nabla f(x) + \sum_i u_i \nabla g_i(x) = 0, \quad (18)$$

$$u \geq 0. \quad (19)$$

В этой двойственной задаче x и u рассматриваются как переменные. Допустимая область такова:

$$D_w = \{x, u / (x, u) \text{ удовлетворяют (18) — (19)}\}. \quad (20)$$

Двойственная задача по Вульффу эквивалентна минимаксной двойственной в следующем смысле:

$$D_w = \{(x, u) / u \in D, x \in X(u)\}, \quad (21)$$

$$L(x, u) = h(u) \text{ для каждого } (x, u) \in D_w. \quad (22)$$

Эти соотношения следуют из того, что если f и g_i выпуклы и дифференцируемы, то $L(x, u)$ выпукла и дифференцируема по x для всех $u \geq 0$ и достигает минимума по x , если и только если $u \in D$ и $\nabla_x L(x, u) = 0$. Таким образом, если $u \in D$ и x минимизируют $L(x, u)$, то $(x, u) \in D_w$, и обратно.

Дальнейшее изложение будет посвящено выяснению связи между прямой и двойственной задачами типа (5) — (6). Мы покажем, что седловая точка $L(x, u)$ существует, если и только если оптимальные значения прямой и двойственной целевых функций равны. В этом случае любая u компонента седловой точки является решением двойственной. Исследование дифференцируемости h приводит тогда к двухуровневому декомпозиционному алгоритму решения нелинейных задач.

Пример 1. Для иллюстрации рассмотрим прямую задачу вида

$$\min [(x_1 - 1)^2 + (x_2 - 2)^2]$$

при ограничениях

$$x_1 + x_2 \leq 2, \quad x_1 \geq 0, \quad x_2 \geq 0,$$

которая имеет оптимальное решение $x_1^0 = 0,5$; $x_2^0 = 1,5$; $\min f = 0,5$. Если ограничения неотрицательности определяют множество S , то двойственная функция формулируется следующим образом:

$$\begin{aligned} h(u) &= \min \{(x_1 - 1)^2 + (x_2 - 2)^2 + u(x_1 + x_2 - 2) / x_1 \geq 0, x_2 \geq 0\} = \\ &= \min \{(x_1 - 1)^2 + ux_1 / x_1 \geq 0\} + \min \{(x_2 - 2)^2 + ux_2 / x_2 \geq 0\} - 2u. \end{aligned}$$

Так как минимум здесь существует при любых u ,

$$D = \{u / u \geq 0\}.$$

Решим каждую из независимых задач минимизации по x_1 и x_2 :

$$x_1(u) = \begin{cases} 1 - \frac{u}{2}, & 0 \leq u \leq 2, \\ 0, & u > 2, \end{cases} \quad x_2(u) = \begin{cases} 2 - \frac{u}{2}, & 0 \leq u \leq 4, \\ 0, & u > 4. \end{cases}$$

Функция h такова:

$$h(u) = \begin{cases} -\frac{u^2}{2} + u, & 0 \leq u \leq 2, \\ -\frac{u^2}{4} + 1, & 2 < u \leq 4, \\ 5 - 2u, & 4 < u. \end{cases}$$

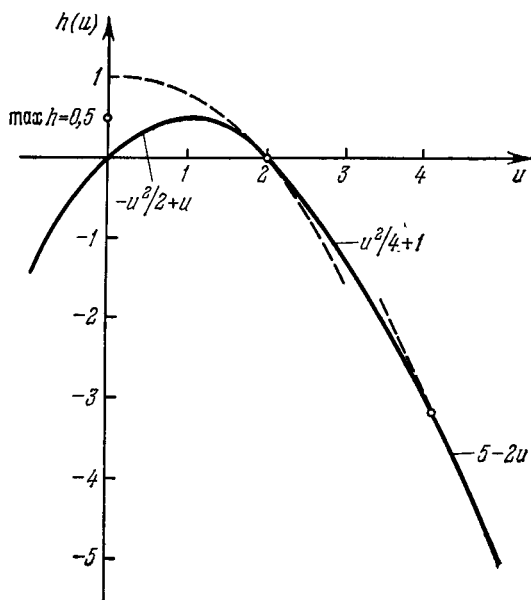


Рис. 7.1. График h из примера 1.

График h показан на рис. 7.1. Максимум h достигается при $u^0 = 1$. Заметим, что

$$\max h = \min f = \frac{1}{2}.$$

Итак,

$$x_1(1) = 0,5 = x_1^0, \quad x_2(1) = 1,5 = x_2^0.$$

Точка $(0,5, 1,5; 1)$ является седловой точкой лагранжевой функции L . Далее, h является нижней границей $\min f$ при любых $u \geq 0$:

$$h(u) \leq f(x), \quad \forall u \geq 0, \quad \forall x \in S. \quad (23)$$

Наконец, h вогнута для неотрицательных u и дифференцируема во всех точках $u \geq 0$, включая $u=2$ и $u=4$.

Условие (23) и вогнутость h на выпуклых подмножествах D будут доказаны и в общем случае. Равенство $\max h = \min f$ спра-

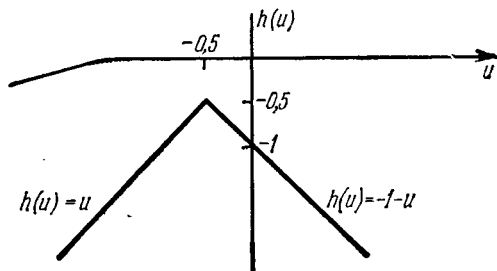


Рис. 7.2. График h из примера 2.

выпукла, все g_i выпуклы, а S замкнуто и выпукло, то h имеет непрерывные первые производные всюду внутри D .

Пример 2. Рассмотрим задачу вида

$$\min \{-x^2/1 - 2x = 0, \quad 0 \leq x \leq 1\},$$

где

$$h(u) = \min_{0 \leq x \leq 1} (-x^2 + u(1-2x)).$$

Производя минимизацию, получаем

$$h(u) = \begin{cases} u & u \leq -\frac{1}{2}, \\ -1-u, & u > -\frac{1}{2}. \end{cases}$$

График h показан на рис. 7.2. Заметим, что $\max h = -1/2 < \min f = -1/4$ и что $x^0 = 1/2 \notin X(u^0) = \{0, 1\}$. Более того, h недифференцируема в точке $u^0 = -1/2$, хотя и имеет конечные производные слева и справа. Однако $\max h$ является нижней границей для $\min f$, и h вогнута.

Перейдем теперь к рассмотрению вопроса, при каких дополнительных предположениях имеют место эти свойства. Прежде всего докажем, что $h(u)$ является нижней границей для $\min f$.

ведливо только в том случае, когда существует седловая точка. Функция h не является в общем случае дифференцируемой всюду, даже для выпуклых задач. Различные сегменты, образующие h , обычно связываются с разрывом производной. Однако, как будет показано в §§ 7.5 и 7.7, если f строго

Иначе говоря, покажем, что если взять точки, допустимые для прямой и соответственно двойственной задачи, то соответствующие им значения целевых функций ограничат сверху и снизу оптимум прямой задачи. Это условие может оказаться полезным для установления условий останковки вычислений в итеративных алгоритмах, позволяющих находить такие допустимые точки.

Теорема 1 (о нижней границе):

$$h(u) \leq f(x) \begin{cases} \text{для всех } x, \text{ удовлетворяющих (5),} \\ \text{для всех } u \in D. \end{cases} \quad (24)$$

Доказательство.

$$h(u) = \min_{x \in S} (f(x) + ug(x)), \quad (25)$$

$$h(u) \leq f(x) + ug(x), \quad \forall x \in S. \quad (26)$$

Для всех $x \in S$, удовлетворяющих $g(x) \leq 0$, имеет место

$$h(u) \leq f(x), \quad (27)$$

следовательно, теорема доказана.

Если допустимые области как прямой, так и двойственной задачи не пусты, то из (24) сразу следует, что $\sup h$ и $\inf f$, взятые на соответствующих ограничивающих множествах, являются конечными. Не обязательно, конечно, чтобы супремум или инфимум достигались в некоторой допустимой точке. Если же это имеет место, то из (24) вытекает, что если как прямая, так и двойственная имеют допустимые планы, то обе они имеют оптимальные решения. Это всегда справедливо для линейных задач и было бы справедливо и здесь, если бы мы захотели заменить на супремум максимум и на инфимум минимум в формулировке прямой и двойственной задач. Другие результаты, немедленно следующие из (24), сформулируем в виде следствий.

Следствие 1.

(а) Если $\sup \{h(u) / u \in D\} = +\infty$, то прямая задача не имеет допустимых планов.

(б) Если $\inf \{f(x) / x \text{ — допустимые в прямой задаче}\} = -\infty$, то двойственная задача недопустима.

Следствие 2. Если существует допустимое решение прямой задачи x^0 и двойственной u^0 , такие, что $f(x^0) = h(u^0)$, то x^0 является решением прямой, а u^0 — решением двойственной.

Теорема 2. Двойственная функция h является вогнутой на любом выпуклом подмножестве ее области определения D .

Доказательство. Пусть $\hat{D}$ является выпуклым подмножеством D и пусть $u_1 \in \hat{D}$, $u_2 \in \hat{D}$. Тогда.

$$h(\alpha u_1 + (1-\alpha)u_2) = \min_{x \in S} L(x, \alpha u_1 + (1-\alpha)u_2). \quad (28)$$

Так как L линейна по u , имеем

$$h(\alpha u_1 + (1-\alpha)u_2) = \min_{x \in S} [\alpha L(x, u_1) + (1-\alpha)L(x, u_2)], \quad (29)$$

$$h(\alpha u_1 + (1-\alpha)u_2) \geq \alpha \min_{x \in S} L(x, u_1) + (1-\alpha) \min_{x \in S} L(x, u_2), \quad (30)$$

$$h(\alpha u_1 + (1-\alpha)u_2) \geq \alpha h(u_1) + (1-\alpha)h(u_2). \quad (31)$$

Как было показано выше, если D — выпуклое множество, то двойственная задача состоит в максимизации вогнутой функции на выпуклом множестве, и, таким образом, не имеется локальных максимумов, отличных от глобального максимума. В частности, если f и g , непрерывны на S и если S замкнуто и ограничено, то $D = (E^m)^+$, т. е. выпукло. Если S замкнуто, но не ограничено, то его можно искусственно сделать ограниченным, вводя верхние границы для всех компонент x , достаточно большие для того, чтобы гарантировать, чтобы любые физически возможные значения векторов x удовлетворяли им. Таким образом, двойственная задача может быть, вообще говоря, сделана хорошо определенной. В этом случае единственный вопрос, который еще остается нерешенным, состоит в том, справедливо ли равенство $\max h = \min f$, т. е. достигается ли равенство в (24).

Покажем теперь, что равенство в (24) может достигаться, если и только если существует седловая точка L . Это утверждение тесно связано с установленными результатами. Если f и g непрерывны, а S компактно, то тогда легко показать (задача 6), что

$$\min f = \min_{x \in S} \sup_{u \geq 0} L(x, u). \quad (32)$$

Таким образом, прямая и двойственная задачи могут быть сформулированы в виде:

$$\begin{array}{l} \text{Прямая} \\ \min_{x \in S} \sup_{u \geq 0} L(x, u), \end{array} \quad (33)$$

$$\begin{array}{l} \text{Двойственная} \\ \max_{u \geq 0} \min_{x \in S} L(x, u). \end{array} \quad (34)$$

Если бы не было асимметрии из-за наличия внутреннего супремума (вместо максимума) в прямой задаче, то равенство целевых функций прямой и двойственной задач требовало бы совпадения минимаксного и максиминного значения функции Лагранжа. Хорошо известно (Карлин [4]), что эти значения равны, если и только если седловая точка существует. Доказательство, что такое же утверждение справедливо и для введенных нами прямой и двойственной задач, дается ниже.

Теорема 3. (x^0, u^0) является седловой точкой для L , если и только если x^0 является допустимой для прямой задачи, u^0 является допустимой для двойственной и $f(x^0) = h(u^0)$.

Доказательство. Необходимость. Так как (x^0, u^0) является седловой точкой, то выполняются условия теоремы 7.3.—1. Следовательно, x^0 является допустимой для прямой, а u^0 для двойственной и

$$u^0 g(x^0) = 0, \quad (35)$$

так что

$$h(u^0) = f(x^0) + u^0 g(x^0) = f(x^0). \quad (36)$$

Достаточность. Так как x^0 является допустимой точкой, то

$$g(x^0) \leq 0. \quad (37)$$

Если $x^0 \notin X(u^0)$, то существует точка $\tilde{x} \in X(u^0)$ такая, что

$$h(u^0) = f(\tilde{x}) + u^0 g(\tilde{x}) < f(x^0) + u^0 g(x^0). \quad (38)$$

Так как $h(u^0) = f(x^0)$, то из (38) следует

$$f(x^0) < f(x^0) + u^0 g(x^0) \quad (39)$$

или

$$u^0 g(x^0) > 0. \quad (40)$$

Но $u^0 \geq 0$ и $g(x^0) \leq 0$, что приводит к противоречию. Таким образом,

$$x^0 \in X(u^0), \quad (41)$$

так что

$$f(x^0) = h(u^0) = f(x^0) + u^0 g(x^0), \quad (42)$$

откуда следует

$$u^0 g(x^0) = 0. \quad (43)$$

В силу теоремы 7.3—1, соотношения (43), (41) и (37) доказывают, что (x^0, u^0) является седловой точкой функции Лагранжа L .

Рассмотрим вновь прямые задачи, для которых L является сепарабельной по x , что приводит к распадению лагранжевой проблемы на подпроблемы. Из предшествующих результатов следует, что возможный путь координации этих подзадач, т. е. обеспечение того, чтобы их решения в совокупности давали решение прямой задачи, состоит в выборе множителей u , являющихся решениями двойственной задачи. Таким образом, координатор должен решать задачу максимизации h при ограничениях $u \in D$. Схема решения дается ниже.

1. Выбрать начальные значения $u \in D$ и решить лагранжевую задачу (7.3—25). В сепарабельном случае это может быть сделано раздельным решением подзадач (7.3—24).

2. Оценить двойственную функцию $h(u)$ и выбрать новые множители так, чтобы эта функция возросла.

3. Вернуться к шагу 1, остановившись, когда h достигнет максимума.

Если при окончании этой процедуры u^0 дает решение двойственной для некоторого $x^0 \in X(u^0)$, являющегося допустимым планом прямой задачи, и $h(u^0) = f(x^0)$, то x^0 является решением прямой задачи. Выполнение шага 2 будет кратко обсуждено в дальнейшем.

Геометрическая интерпретация. Более полное понимание проблемы может быть получено с помощью следующей геометрической интерпретации [10, 11]. Погрузим исходную задачу в семейство возмущенных задач с возмущениями y :

$$\min f(x) \quad (44)$$

при ограничениях

$$g_i(x) \leq y_i, \quad i = 1, \dots, m, \quad (45)$$

$$x \in S, \quad (46)$$

т. е. рассмотрим семейство задач с переменными правыми частями $y = (y_1, \dots, y_m)$. Исходная задача соответствует $y = 0$. Пусть

$$\omega(y) = \min \{f(x)/g(x) \leq y, x \in S\}. \quad (47)$$

Функция ω может не существовать для всех y , либо в силу того, что задача (44)–(46) может не иметь допустимого решения, либо в силу того, что инфимум f при ограничениях $g(x) \leq y, x \in S$ не достигается. Для того чтобы избежать этой трудности, предположим, что S замкнуто и ограничено, а f и g_i непрерывны на S . Тогда область определения ω такова

$$F = \{y / \exists x \in S \text{ такое, что } g(x) \leq y\}. \quad (48)$$

Функция ω — невозрастающая на F . Если f, g и S выпуклы, то может быть показано, что F является выпуклым множеством, а ω — выпуклая функция на F (см. задачу 5). Рассмотрим множество точек R в E^{m+1} , лежащих на и выше графика $\omega(y)$:

$$R = \{(y_0, y) / y \in F, y_0 \geq \omega(y)\}. \quad (49)$$

Если ω и F выпуклы, то это множество выпукло и может быть наглядно представлено так, как это сделано на рис. 7.3. Имеется тесная связь между двойственностью и опорными гиперплоскостями для этого множества, как показывает следующая теорема.

Теорема 4. Пусть $\hat{x} \in S$ и $\hat{u} \geq 0$. Тогда $\hat{x}$ минимизирует $L(x, \hat{u})$ на S , если и только если

$$H = \{(y_0, y) / y_0 + \hat{u}y = L(\hat{x}, \hat{u})\} \quad (50)$$

является опорной гиперплоскостью к R в точке $(f(\hat{x}), g(\hat{x}))$.

Доказательство. Необходимость. Мы должны показать, что

$$y_0 + \hat{u}y \geq L(\hat{x}, \hat{u}), \quad \forall (y_0, y) \in R.$$

Если для некоторого $(y_0^*, y^*) \in R$

$$y_0^* + \hat{u}y^* < L(\hat{x}, \hat{u}),$$

тогда, поскольку $y_0^* \geq \omega(y^*) = f(x^*)$, $y^* \geq g(x^*)$,

$$f(x^*) + \hat{u}g(x^*) < L(\hat{x}, \hat{u}).$$

Так как $x^* \in S$, то это противоречит утверждению, что $\hat{x}$ минимизирует $L(x, \hat{u})$ на S .

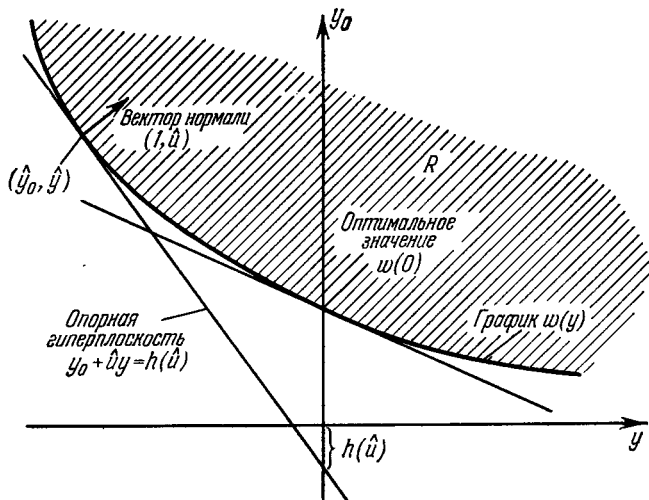


Рис. 7.3. Гиперплоскость, опорная к множеству R .

Достаточность. Пусть

$$y_0 + \hat{u}y \geq L(\hat{x}, \hat{u}), \quad \forall (y_0, y) \in R. \quad (51)$$

Рассмотрим множество

$$R_1 = \{(f(x), g(x)) / x \in S\}.$$

Выберем $\tilde{x} \in S$ и образуем вектор $(f(\tilde{x}), g(\tilde{x})) \in R_1$. Если положить $y = g(\tilde{x})$, то $y \in F$ и $f(\tilde{x}) \geq \omega(y)$. Таким образом, $(f(\tilde{x}), g(\tilde{x})) \in R$ и, так как $\tilde{x}$ является произвольным элементом S , то $R \supseteq R_1$.

При использовании этого факта, из (51) получаем

$$f(x) + \hat{u}g(x) \geq L(\hat{x}, \hat{u}), \quad \forall x \in S,$$

так что $\hat{x}$ минимизирует $L(x, \hat{u})$ на S .

Согласно полученному результату, процесс минимизации $L(x, \hat{u})$ на S эквивалентен нахождению величины

$$h(\hat{u}) = \min_{x \in S} L(x, \hat{u}) = L(\hat{x}, \hat{u}),$$

такой, что гиперплоскость H в (50) является опорной гиперплоскостью для множества R . Значение двойственной функции, $h(u)$, определяется пересечением этой опорной гиперплоскости с осью y_0 , и углы наклона этой гиперплоскости задают числа — $\hat{u}$. Следовательно, двойственная задача состоит в нахождении опорной плоскости с неположительными коэффициентами, имеющей самую высокую точку пересечения y_0 , в то время как прямая задача состоит в нахождении точки $(y_0, 0)$ в R с минимальным уровнем пересечения y_0 , равным числу $w(0)$.

Так как w является невозрастающей, то все опорные плоскости к R имеют углы наклона, удовлетворяющие условию $-u \leq 0$. Из рис. 7.3 очевидно, что любая такая опорная плоскость имеет значение уровня пересечения y_0 равное или более низкое, чем оптимум исходной задачи $w(0)$. Это утверждение соответствует неравенству $h(u) \leq \min f$. Если R замкнуто и выпукло, то опорные гиперплоскости существуют во всех граничных точках, в частности, в точке $(w(0), 0)$. Пересечение этой гиперплоскости с осью y_0 равно $w(0)$, что и приводит к соотношению $\min f = \max h$.

Теорема 4 приводит к интерпретации двойственной задачи в терминах сопряженных функций. Используя определение 9 из приложения 1 и комментарии, следующие после него, мы убеждаемся, что

$$h(u) = -w^*(-u),$$

где w^* является сопряженной к w . Рокафеллар [10] использовал понятие сопряженности для установления элегантной и симметричной теории двойственности для выпуклых задач.

Геометрическая точка зрения показывает также, в каких ситуациях может быть успешным лагранжев подход. Рассмотрим исходную задачу с оптимальной целевой функцией $w(0) = f(x^0)$. Результат, формулируемый ниже, прямо следует из теоремы 4.

Следствие. Пусть x^0 — решение исходной задачи. Тогда существуют множители $u^0 \geq 0$, такие, что x^0 минимизирует $L(x, u^0)$ на S , если и только если множество R в (49) имеет опорную гиперплоскость в точке $(f(x^0), g(x^0))$ с углами наклона — u^0 .

Это следствие приложимо к прямым задачам с произвольным вектором правых частей y . Таким образом, мы получили характеристику «зазоров», рассматривавшихся в предшествующем параграфе.

Любой вектор y , для которого R имеет опорную гиперплоскость в некоторой точке $(w(y), v)$ с $v \leq y$, может быть найден

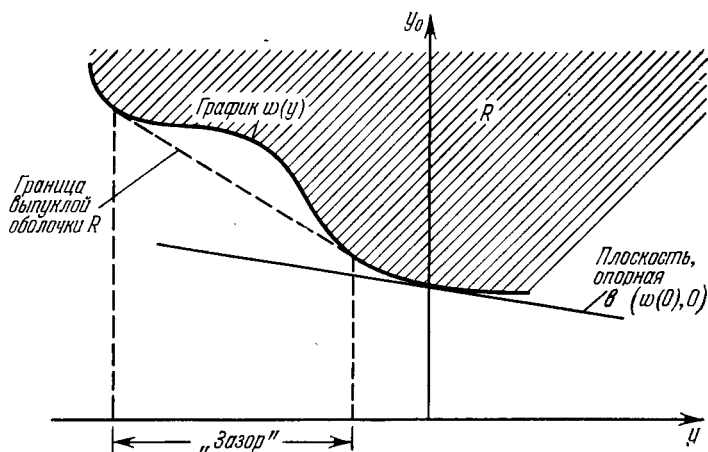


Рис. 7.4. Плоскость, опорная в оптимальной точке.

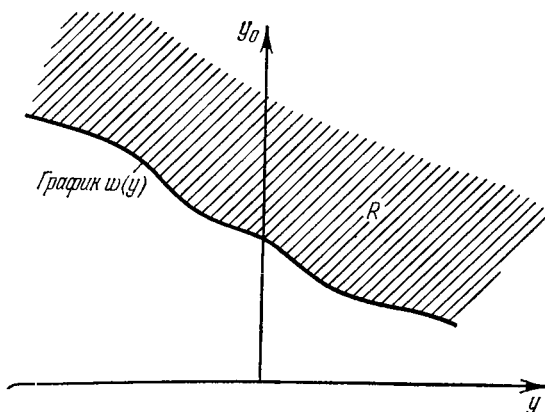


Рис. 7.5. В оптимальной точке не существует опорной плоскости.

путем решения лагранжевой задачи при некотором $u \geq 0$. Таковыми являются векторы y , для которых $(w(y), v)$ лежит в пересечении границы выпуклой оболочки R с границей R . Векторы y , не обладающие этим свойством, образуют «зазоры». На рис. 7.4 и 7.5 множества R невыпуклы, отражая невыпуклость

исходной задачи. Несмотря на это, на рис. 7.4 R имеет опорную плоскость $(w(0), 0)$, в то время как на рис. 7.5 в этой точке не имеется никакой опорной плоскости. Существует множество «зазоров», один из которых отмечен на рис. 7.4.

Предшествующее следствие показывает, что с помощью метода Лагранжа можно решить некоторые невыпуклые задачи. В частности, если можно выделить класс задач, для которых ассоциированное множество R всегда имеет опорную гиперплоскость в некоторой точке $(f(x^0), g(x^0))$, при $f(x^0) = w(0)$, $g(x^0) \leq 0$, то любая задача из этого класса может быть решена путем решения лагранжевой задачи для некоторого вектора $u^0 \geq 0$.

§ 7.5. ДИФФЕРЕНЦИРУЕМОСТЬ ДВОЙСТВЕННОЙ ЦЕЛЕВОЙ ФУНКЦИИ *)

При конструировании алгоритмов решения двойственной задачи существенна информация, касающаяся производных двойственной целевой функции, h . Например, когда допустимым множеством в двойственной задаче D , является $(E^m)^+$ и h дифференцируема, то можно с успехом использовать метод наискорейшего спуска. Особенно приятным свойством двойственной задачи является то, что частные производные $\partial h / \partial u_i$ легко получить, если таковые существуют. Однако, как было показано в примере 2 § 7.4, функция h не является, вообще говоря, дифференцируемой во всех точках. Однако, поскольку h вогнута на выпуклых подмножествах D , то она дифференцируема почти всюду внутри таких подмножеств [13] и обладает (односторонними) производными по любым направлениям во всех точках таких подмножеств.

Выведем необходимые и достаточные условия, при которых h является дифференцируемой в данной точке, и найдем выражение для ее частных производных и производных по направлению. Главным инструментом, используемым при выводе, является результат, принадлежащий Данскину [12], который справедлив для любой функции ϕ , определенной как минимум некоторой другой функции $F(x, u)$ на $x \in S$. Этот результат является вполне общим и остается справедливым, когда x представляет собой элемент произвольного компактного топологического пространства.

Пусть функция $F(x, u)$ определена для всех u в E^m и для всех x в некотором топологическом пространстве S .

Примем:

Предположение 1. S является компактным.

*) В этом параграфе используются результаты Приложений 1 и 2.

Предположение 2. F и частные производные $\partial F/\partial u_i$ непрерывны.

Введем обозначения

$$\varphi(u) = \min_{x \in S} F(x, u), \quad (1)$$

$$X(u) = \{x/x \text{ минимизирует } F(x, u) \text{ на } S\}. \quad (2)$$

Односторонняя производная φ по направлению s в точке u определяется следующим образом:

$$D\varphi(u; s) = \lim_{\alpha \rightarrow 0^+} \frac{\varphi(u + \alpha s) - \varphi(u)}{\alpha}. \quad (3)$$

Теорема 1 (Данскин [12]). *При предположениях 1 и 2 существуют (односторонние) производные φ по любому направлению s в любой точке $u \in E^m$, и эти производные задаются формулами*

$$D\varphi(u; s) = \min_{x \in X(u)} \sum_{i=1}^m s_i \frac{\partial F(x, u)}{\partial u_i}. \quad (4)$$

Доказательство. Пусть u является произвольной точкой в E^m и пусть $\{\alpha^k\}_{k=1}^{\infty}$ — последовательность положительных чисел, стремящаяся к нулю. Обозначим

$$u^k = u + \alpha^k s. \quad (5)$$

Пусть $x^k \in X(u^k)$, $x \in X(u)$. Рассмотрим следующие отношения:

$$\frac{\varphi(u^k) - \varphi(u)}{\alpha^k} = \frac{F(x^k, u^k) - F(x, u)}{\alpha^k}, \quad (6)$$

$$\frac{\varphi(u^k) - \varphi(u)}{\alpha^k} = \frac{F(x^k, u^k) - F(x, u^k)}{\alpha^k} + \frac{F(x, u^k) - F(x, u)}{\alpha^k}. \quad (7)$$

Так как $x^k \in X(u^k)$, то

$$F(x^k, u^k) \leq F(x, u^k), \quad (8)$$

и опуская первое слагаемое в правой части (7), получаем

$$\frac{\varphi(u^k) - \varphi(u)}{\alpha^k} \leq \frac{F(x, u^k) - F(x, u)}{\alpha^k}, \quad \forall x \in X(u). \quad (9)$$

В силу теоремы о среднем, имеем

$$\frac{F(x, u^k) - F(x, u)}{\alpha^k} = \sum_{i=1}^m \frac{u_i^k - u_i}{\alpha^k} \cdot \frac{\partial F(x, u + \theta^k(u^k - u))}{\partial u_i}, \quad (10)$$

$$0 < \theta^k < 1,$$

$$\frac{F(x, u^k) - F(x, u)}{\alpha^k} = \sum_{i=1}^m s_i \frac{\partial F(x, u + \theta^k(u^k - u))}{\partial u_i}. \quad (11)$$

Комбинируя (9) и (11), получаем

$$\frac{\varphi(u^k) - \varphi(u)}{\alpha^k} \leq \sum_{i=1}^m s_i \frac{\partial F(x, u + \theta^k(u^k - u))}{\partial u_i}, \quad \forall x \in X(u). \quad (12)$$

При $k \rightarrow \infty$ $u^k \rightarrow u$.

По непрерывности, частные производные в правой части (12) стремятся к $\frac{\partial F(x, u)}{\partial u_i}$. В силу (12), каждый элемент последовательности разностных отношений ограничен сверху и наибольшая из ее предельных точек ограничена:

$$\limsup_{u^k \rightarrow u} \frac{\varphi(u^k) - \varphi(u)}{\alpha^k} \leq \sum_{i=1}^m s_i \frac{\partial F(x, u)}{\partial u_i}, \quad \forall x \in X(u). \quad (13)$$

Дальнейшая цель состоит в том, чтобы получить неравенство, подобное (13), но обратного знака, определяющее нижний предел последовательности разностных отношений. Пусть $\{u^{k*}\}$ — последовательность такая, что

$$\frac{\varphi(u^{k*}) - \varphi(u)}{\alpha^{k*}} \rightarrow \liminf_{u^k \rightarrow u} \frac{\varphi(u^k) - \varphi(u)}{\alpha^k}. \quad (14)$$

Пусть $x^{k*} \in X(u^{k*})$. Так как каждая $x^{k*} \in S$, и S является компактным, то существует подпоследовательность $\{x^{k'}\}$ последовательности $\{x^{k*}\}$, которая сходится к пределу $x' \in S$. Соответствующая ей подпоследовательность $\{u^{k*}\}$ есть $\{u^{k'}\}$. Так как

$$F(x^{k'}, u^{k'}) \leq F(x, u^{k'}), \quad \forall x \in S, \quad \forall k', \quad (15)$$

то, полагая $k' \rightarrow \infty$ и учитывая непрерывность, получаем

$$F(x', u) \leq F(x, u), \quad \forall x \in S, \quad (16)$$

откуда

$$x' \in X(u). \quad (17)$$

Рассмотрим теперь свойства разностных отношений, соответствующих последовательностям $\{u^{k'}\}$, $\{x^{k'}\}$.

$$\frac{\varphi(u^{k'}) - \varphi(u)}{\alpha^{k'}} = \frac{F(x^{k'}, u^{k'}) - F(x^{k'}, u)}{\alpha^{k'}} + \frac{F(x^{k'}, u) - F(x, u)}{\alpha^{k'}}. \quad (18)$$

Так как $x \in X(u)$, то

$$F(x, u) \leq F(x^{k'}, u). \quad (19)$$

Таким образом, (18) можно переписать в виде неравенства

$$\frac{\varphi(u^{k'}) - \varphi(u)}{\alpha^{k'}} \geq \frac{F(x^{k'}, u^{k'}) - F(x^{k'}, u)}{\alpha^{k'}}. \quad (20)$$

Вновь используем теорему о среднем:

$$\frac{\varphi(u^{k'}) - \varphi(u)}{\alpha^{k'}} \geq \sum_{i=1}^m \left(\frac{u_i^{k'} - u_i}{\alpha^{k'}} \right) \cdot \frac{\partial F(x^{k'}, u + \omega^{k'}(u^{k'} - u))}{\partial u_i},$$

$$0 < \omega^{k'} < 1 \quad (21)$$

$$\frac{\varphi(u^{k'}) - \varphi(u)}{\alpha^{k'}} = \sum_{i=1}^m s_i \frac{\partial F(x^{k'}, u + \omega^{k'}(u^{k'} - u))}{\partial u_i}. \quad (22)$$

Пусть $k' \rightarrow \infty$. Вновь, в силу непрерывности, из (22) получаем

$$\liminf_{u^k \rightarrow u} \frac{\varphi(u^k) - \varphi(u)}{\alpha^k} \geq \sum_{i=1}^m s_i \frac{\partial F(x', u)}{\partial u_i}. \quad (23)$$

Рассмотрим теперь соотношения (13) и (23). Поскольку (13) справедливо для всех $x \in X(u)$, то

$$\limsup_{u^k \rightarrow u} \frac{\varphi(u^k) - \varphi(u)}{\alpha^k} \leq \min_{x \in X(u)} \sum_{i=1}^m s_i \frac{\partial F(x, u)}{\partial u_i}. \quad (24)$$

Из соотношений (23) — (24) следует, что последовательность $\{(\varphi(u^k) - \varphi(u))/\alpha^k\}$ имеет только одну предельную точку, даваемую соотношением

$$D\varphi(u; s) = \lim_{u^k \rightarrow u} \frac{\varphi(u^k) - \varphi(u)}{\alpha^k} = \min_{x \in X(u)} \sum_{i=1}^m s_i \frac{\partial F(x, u)}{\partial u_i}. \quad (25)$$

Теорема доказана.

Для того чтобы применить теорему 1 к функции $h(u)$, положим $F(x, u) = L(x, u)$. Поскольку $\partial L / \partial u_i = g_i(x)$, предположения 1 и 2 приобретают следующий вид:

Предположение 1'. S является замкнутым и ограниченным подмножеством E^n .

Предположение 2'. $f(x)$ и все $g_i(x)$ непрерывны на S .

При этих предположениях $D = (E^m)^+$. Однако двойственная функция h может быть определена для всех $u \in E^m$ просто путем распространения исходного определения на значения u с отрицательными компонентами. Мы используем это распространение следующим образом.

Теорема 2 (производные h по направлению). При предположениях 1 и 2 существуют (односторонние) производные h по любому направлению s в любой точке $u \in E^m$, которые даются соотношением

$$Dh(u; s) = \min_{x \in X(u)} \sum_{i=1}^m s_i g_i(x). \quad (26)$$

Так как при предположениях 1' и 2' функция h замкнута, правильна и выпукла, и область ее определения совпадает с E^m , существование и конечность производных по направлению может быть также выведена из теорем 9 и 15 Приложения 2. По аналогии с определением выпуклых функций мы определим субградиент вогнутой функции h в точке u как вектор коэффициентов опорной гиперплоскости к графу h в точке u . Множество всех субградиентов h в точке u и есть $\partial h(u)$. Таким образом, мы ввели следующее определение.

Определение 1. x^* есть субградиент h в точке u , если

$$h(z) \leq h(u) + x^*(z - u), \quad \forall z \in E^m.$$

Из следующего ясно, что оценка $h(u)$ немедленно дает элемент $\partial h(u)$.

Теорема 3.

$$g(x) \in \partial h(u), \quad \forall x \in X(u).$$

Доказательство. По определению,

$$h(z) \leq f(x) + zg(x), \quad \forall x \in S, \quad (27)$$

$$h(u) = f(x^0) + ug(x^0), \quad \forall x^0 \in X(u).$$

Полагая $x = x^0$ в (27) и вычитая эти соотношения, получаем

$$h(z) \leq h(u) + g(x^0)(z - u), \quad \forall z \in E^m.$$

Легко построить примеры, которые показывают, что могут существовать элементы $\partial h(u)$, которые не равны $g(x)$ при любых $x \in X(u)$. Определим

$$V(u) = \{g(x) / x \in X(u)\}.$$

В примере 2 § 7.4 $\partial h\left(-\frac{1}{2}\right) = [-1, 1]$, в то время как $V\left(-\frac{1}{2}\right) = \{-1, 1\}$. Таким образом, теорема 2 дает более сильный результат, чем тот, который может быть получен прямым применением теоремы 16 из Приложения 2 к $-h$, который дал бы следующее:

$$Dh(u; s) = \min \{u^*s / u^* \in \partial h(u)\}. \quad (28)$$

По теореме 2, указанный выше минимум всегда достигается в точках подмножества $\partial h(u)$, точнее, в точках множества $V(u)$. Отсюда легко показать, что $V(u)$ должно содержать все крайние точки $\partial h(u)$. (В предшествующем примере $V(u)$ совпадало с множеством крайних точек.)

Теорема 4. При предположениях 1' и 2', если v является крайней точкой $\partial h(u)$, то $v \in V(u)$.

Доказательство. Если v — крайняя точка, тогда, так как $h(u)$ компактно, существует гиперплоскость, которая строго отделяет эту точку от $\partial h(u)$, т. е. существует точка $s^0 \in E^n$ такая что

$$vs^0 < u^*s^0, \quad \forall u^* \in \partial h(u), \quad u^* \neq v.$$

Таким образом, минимум в (28) достигается в единственной точке v . Так как этот минимум должен обязательно достигаться в $V(u)$, то $v \in V(u)$.

Из условия $v \in V(u)$ не обязательно следует, что v крайняя точка $\partial h(u)$, поскольку легко привести пример обратного. Рассмотрим задачу

$$\min \{x/1 - 2x \leq 0, 0 \leq x \leq 1\},$$

где

$$h(u) = \min_{0 \leq x \leq 1} (x + u(1 - 2x)).$$

$$\text{Здесь } X\left(\frac{1}{2}\right) = [0, 1] \text{ и } V\left(\frac{1}{2}\right) = [-1, 1] = \partial h\left(\frac{1}{2}\right).$$

Другие свойства производной по направлению следуют из вогнутости $h(u)$ и других результатов Данскина.

1. (Приложение 2, теорема 9.) $Dh(u; s)$ вогнута и положительно однородна по s при фиксированном u .

2. (Данскин [12]). $Dh(u; s)$ полунепрерывна снизу по u при фиксированном s , т. е. если $\{u^k\} \rightarrow u$, то

$$\liminf_{u^k \rightarrow u} Dh(u^k; s) \geq Dh(u; s).$$

3. (Данскин [12]). Если $s = \alpha s_1 + \beta s_2$, где α, β — любые вещественные числа, то

$$Dh(u; s) \geq \alpha Dh(u; s_1) + \beta Dh(u; s_2). \quad (29)$$

Соотношение (29) в виде равенства есть обычная формула, которая сохраняет свою силу, когда производная по направлению может быть записана в виде $Dh(u; s) = \nabla h'(u)s$. Так как производная по направлению h не всегда может быть записана таким образом, то гарантируется лишь выполнение неравенства (29).

Следовательно, вообще говоря, знание производных по координатным осям недостаточно для вычисления производной по другим направлениям.

Для любой заданной точки u частная производная h справа получается, если положить $s=i$ -й единичный вектор e_i в (26):

$$\left(\frac{\partial h}{\partial u_i}\right)^+ = \min_{x \in X(u)} g_i(x). \quad (30)$$

Частная производная слева соответственно такова:

$$\left(\frac{\partial h}{\partial u_i}\right)^- = \lim_{\alpha \rightarrow 0^+} \frac{h(u) - h(u - \alpha e_i)}{\alpha} = -Dh(u; -e_i) = -\min_{x \in X(u)} (-g_i(x)),$$

или

$$\left(\frac{\partial h}{\partial u_i}\right)^- = \max_{x \in X(u)} g_i(x). \quad (31)$$

Формулы (30)–(31) согласованы с тем обстоятельством, что для вогнутой функции имеет место

$$\left(\frac{\partial h}{\partial u_i}\right)^+ \leq \left(\frac{\partial h}{\partial u_i}\right)^-.$$

Необходимое условие дифференцируемости h в точке u состоит в равенстве

$$\left(\frac{\partial h}{\partial u_i}\right)^+ = \left(\frac{\partial h}{\partial u_i}\right)^- = \frac{\partial h}{\partial u_i}, \quad \forall i. \quad (32)$$

В силу (30)–(31) это справедливо если и только если каждая g_i постоянна на $X(u)$ и равна, скажем, $c_i(u)$. В силу (26), тогда производные по направлению могут быть представлены в виде

$$Dh(u; s) = \sum_{i=1}^m s_i c_i(u), \quad (33)$$

т. е. производные по направлению s линейны по компонентам направления s . Так как h вогнута, то согласно следующей теореме Фенхеля из (33) вытекает, что h дифференцируема в точке u .

Лемма 1 (Фенхель [13]). Пусть h — вогнутая функция, определенная на выпуклом множестве D . Пусть $u^0 \in \text{int } D$, и предположим, что $Dh(u^0; s)$ линейна по s . Тогда h дифференцируема в u^0 .

Конечно, если не все g_i постоянны на $X(u)$, то правостороннее и левостороннее частные производные не равны, и h недифференцируема по u . Таким образом, мы приходим к следующей теореме.

Теорема 5. Пусть предположения 1' и 2' справедливы. Тогда двойственная целевая функция h дифференцируема в точке u^* , если и только если все g_i постоянны *) на $X(u^*)$. В этом случае частные производные h определяются формулами

$$\left. \frac{\partial h}{\partial u_i} \right|_{u=u^*} = g_i(x), \quad \forall x \in X(u^*). \quad (34)$$

Интересно отметить, что эти частные производные совпадают с результатом дифференцирования L , если при дифференцировании игнорировать зависимость x от u . Такие производные легко оценить, если минимизирующая точка $x \in X(u)$ известна. Таким образом, если h может быть оценена, и гипотезы теоремы 5 справедливы, то градиент h может быть получен с малыми дополнительными усилиями.

Теорема 5 подтверждается примером 2 § 7.4. В нем $X(u)$ состоит из единственной точки для всех u , за исключением $u^0 = -1/2$. Множество $X(-1/2) = \{0, 1\}$, и функция g непостоянна на этом множестве, следовательно, h дифференцируема всюду, за исключением u^0 . Однако правосторонние и левосторонние частные производные существуют в u^0 и даются формулами

$$\begin{aligned} \left(\frac{\partial h}{\partial u} \right)_{u=-1/2}^+ &= -1 = \min_{x \in X(u^0)} (1 - 2x) = \min \{1, -1\}, \\ \left(\frac{\partial h}{\partial u} \right)_{u=-1/2}^- &= 1 = \max_{x \in X(u^0)} (1 - 2x) = \max \{1, -1\}. \end{aligned}$$

Важный специальный случай, при котором гипотезы теоремы 5 выполняются, возникает когда $L(x, u^*)$ достигает минимума на S в единственной точке, так что $X(u^*)$ состоит только из одного элемента. Сформулируем это в виде следствия.

Следствие 1. Пусть справедливы предположения 1' и 2'. Если $L(x, u^*)$ достигает минимума на S в единственной точке $x(u^*)$, то h дифференцируема в u^* , причем

$$\left. \frac{\partial h}{\partial u_i} \right|_{u=u^*} = g_i(x(u^*)). \quad (35)$$

Может быть дано большое число достаточных условий, которые гарантируют единственность минимума $L(x, u)$. Простейшее из них состоит в строгой выпуклости.

Следствие 2. Пусть выполнены предположения 1' и 2', и, кроме того, f строго выпукла и все g_i выпуклы на S . Тогда h дифференцируема в любой точке $u \geq 0$.

) Поскольку $L(x, u^)$ постоянна на $X(u^*)$, то f также постоянна на этом множестве.

Если сделать некоторые добавочные предположения, то для доказательства соотношения (35) могут быть использованы простые рассуждения. Предположим, например, что f и все g_i дифференцируемы всюду и что $S = E^n$. Тогда лагранжева задача содержит минимизацию без ограничений:

$$\min_{x \in E^n} L(x, u). \quad (36)$$

Предположим, что $-L(x, u^*)$ имеет единственный минимум $x(u^*)$. Тогда $x(u^*)$ удовлетворяет условиям

$$\frac{\partial L(x, u^*)}{\partial x} = 0, \quad i = 1, \dots, m. \quad (37)$$

Предположим теперь, что существует дифференцируемая функция $x(u)$, определенная в окрестности u^* , которая минимизирует $L(x, u)$ для всех u в этой окрестности. Тогда

$$h(u) = L(x(u), u),$$

$$\left. \frac{\partial h}{\partial u_i} \right|_{u^*} = \left(\frac{\partial L}{\partial u_i} + \sum_j \frac{\partial L}{\partial x_j} \frac{\partial x_j}{\partial u_i} \right) \Big|_{u^*}, \quad (38)$$

$$\left. \frac{\partial h}{\partial u_i} \right|_{u^*} = \left. \frac{\partial L}{\partial u_i} \right|_{u^*}, \quad (39)$$

$$\left. \frac{\partial h}{\partial u_i} \right|_{u^*} = g_i(x(u^*)). \quad (40)$$

Теорема 5 прямо приводит к следующей теореме.

Теорема 6. Пусть выполнены предположения 1' и 2'. Пусть u^0 является решением двойственной задачи, и предположим, что h дифференцируема в u^0 . Тогда любой элемент $x^0 \in X(u^0)$ является решением прямой задачи.

Доказательство. При предположениях 1' и 2' $D = (E^m)^+$. Так как h дифференцируема в u^0 , выполняются следующие условия оптимальности:

$$(a) \quad u_i^0 > 0 \Rightarrow \left. \frac{\partial h}{\partial u_i} \right|_{u^0} = g_i(x^0) = 0, \quad (41)$$

$$(b) \quad u_i^0 = 0 \Rightarrow \left. \frac{\partial h}{\partial u_i} \right|_{u^0} = g_i(x^0) \leq 0. \quad (42)$$

Эти условия могут быть переписаны в виде $g(x^0) \leq 0$ и $u^0 g(x^0) = 0$. Из них вместе с утверждением $x^0 \in X(u^0)$ следует, в силу теоремы 7.3—1, что (x^0, u^0) является седловой точкой L , и по теореме 7.3—2, x^0 является решением прямой задачи.

Комбинируя следствие 1 и теорему 6, получаем, что при предположениях 1' и 2', если u^0 является решением двойствен-

ной и $L(x, u^0)$ имеет единственный минимум на S в точке x^0 , то x^0 решает прямую задачу. Таким образом, любой класс задач, для которых эта единственность может быть гарантирована, разрешим путем минимизации $L(x, u)$ на S при некотором u . Следствие 2 указывает такой класс задач. Однако могут существовать и другие типы задач.

§ 7.6. ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ РЕШЕНИЯ ДВОЙСТВЕННОЙ ЗАДАЧИ

Градиентные алгоритмы. Временно предположим, что условия 1' и 2' выполнены, и что h — дифференцируема *). Тогда для максимизации h может быть использован алгоритм наискорейшего спуска, модифицированный с учетом ограничений $u \geq 0$. Это приводит к вычислительной процедуре, которая является более полной версией метода, указанного в § 7.5:

1. Выбрать начальное значение $u^0 \geq 0$. Далее действовать по рекуррентии, причем шаг $i, i=0, 1, 2, \dots$ состоит в следующем:

2. Решить лагранжеву задачу при $u = u^i$, получая решение $x(u^i)$. В сепарабельном случае это может быть выполнено путем решения подзадач (7.3—24).

3. Образовать двойственную функцию $h(u^i) = L(x(u^i), u^i)$ и ее градиент $\nabla h(u^i) = g(x(u^i))$.

4. Определить направление спуска, s^i , следующим образом:

$$s_k^i = \begin{cases} \frac{\partial h}{\partial u_k} \Big|_{u^i}, & \text{если } u_k^i > 0, \\ \max \left\{ 0, \frac{\partial h}{\partial u_k} \Big|_{u^i} \right\}, & \text{если } u_k^i = 0, \end{cases} \quad k = 1, \dots, m. \quad (1)$$

(2)

Выбрать новый вектор u^{i+1} следующим образом:

$$u^{i+1} = u^i + \alpha_i s^i. \quad (3)$$

Размер шага α_i должен быть выбран так, что

$$h(u^{i+1}) > h(u^i). \quad (4)$$

Если h дифференцируема, то существует $\alpha_i > 0$, удовлетворяющее условию (4), если u^i не максимизирует h . Обычная процедура состоит в выборе α_i , максимизирующем

$$r(\alpha) = h(u^i + \alpha s^i) \quad (5)$$

при ограничениях $\alpha_i \geq 0$ и $u^{i+1} \geq 0$.

5. Вернуться к шагу 2, остановившись при $\alpha_i \approx 0$.

В процессе максимизации $r(\alpha)$ приходится оценивать ее значение многократно, что требует решения каждый раз

*) Это имеет место для задач со строго выпуклой целевой функцией (см. § 7.5 и § 7.7).

лагранжевой задачи. Подходящие процедуры одномерного поиска хорошо известны (см., например [14], [15]). Шаг 4 является просто реализацией метода Розена проектирования градиента для линейных ограничений [16], специализированного для случая ограничений простейшего вида $u \geq 0$. Сходимость этого алгоритма к глобальному решению доказана в части 2 работы [16] для случая вогнутых дифференцируемых целевых функций.

Эта градиентная процедура может рассматриваться как координирующий алгоритм для координатора верхнего уровня, задача которого состоит в решении двойственной проблемы при заданных значениях h и ∇h . На нижнем уровне решаются подзадачи и находятся эти значения. Алгоритм имеет интересную экономическую интерпретацию, если прямая задача трактуется как задача распределения ресурсов с минимальными затратами. Пусть прямая задача такова:

$$\min \sum_j f_j(x_j), \quad (6)$$

$$g_{ij}(x) = \sum_j g_{ij}(x_j) \leq b_i, \quad i = 1, \dots, m, \quad (7)$$

$$x_j \in S_j, \quad (8)$$

где $f_j(x_j)$ есть затраты на осуществление производственного способа j на уровне x_j ; $g_{ij}(x_j)$ есть количество ресурса i , используемое способом j на уровне x_j ; b_i есть располагаемое количество ресурса i ; S_j определяет технологические ограничения для каждого способа. Пусть $u^* \geq 0$ рассматривается как вектор цен ресурсов. Предположим, что лагранжева задача имеет единственное решение $u = u^*$, $x(u^*)$. Тогда подзадачи состоят в следующем:

$$\min \tilde{f}_i(x_i, u) = f_i(x_i) + \sum_k u_k^* g_{ki}(x_i), \quad (9)$$

$$x_i \in S_i. \quad (10)$$

Целевые функции подзадач представляют собой прямые затраты f_i плюс затраты на используемые ресурсы. Двойственная функция является суммой минимальных значений целевых функций подзадач:

$$h(u^*) = \sum_i \tilde{f}_i(x_i(u^*), u^*). \quad (11)$$

Градиент h в точке u^* имеет компоненты

$$\left. \frac{\partial h}{\partial u_i} \right|_{u^*} = \sum_j g_{ij}(x_{ij}(u^*)) - b_i = e_i(u^*). \quad (12)$$

Сумма может быть рассматриваема как общая потребность в ресурсе i при условии, что доступное количество равно b_i . Та-

ким образом, $e_i(u^*)$ есть избыточная потребность в i -м ресурсе. В силу (1)–(3), градиентный алгоритм требует, чтобы u_i^* возрастала, если избыточная потребность положительна, и в противном случае уменьшалась, если не достигнуто ограничение $u_i^* = 0$. Но в этом и состоит классическое экономическое правило настройки цен. Хотя метод наискорейшего спуска теоретически интересен, для вычислительных целей стоит рекомендовать более новые методы. Если все ограничения представляют собой равенства, то множители u неограничены по знаку, и эффективно применение метода с переменной метрикой, описанного Флетчером и Пауэллом [14], или метода сопряженного градиента Флетчера–Ривса [17]. Модификация метода Флетчера и Пауэлла для случая линейных ограничений описана в [18] и может быть рекомендована, когда имеет место ограничение $u_i \geq 0$. Все эти процедуры требуют знания только значений функции и градиента и, вообще говоря, сходятся за меньшее число итераций, чем метод наискорейшего спуска.

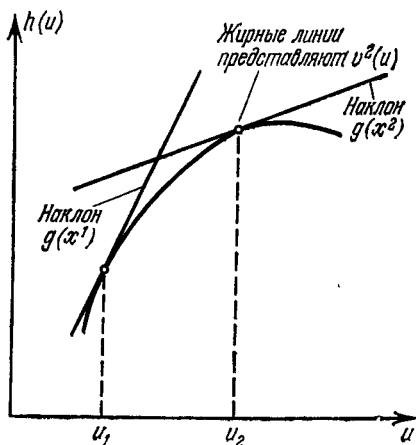


Рис. 7.6. Аппроксимация, касательными к h .

Аппроксимация касательными. Как показано ранее, h не обязательно является дифференцируемой всюду. Если это не имеет места, сходимость различных вариантов метода градиентного спуска не может быть гарантирована. На практике имеет место медленная сходимость или отсутствие сходимости вовсе, если число угловых точек в h велико или же угловой точкой является максимум. Алгоритм, который позволяет избежать этих трудностей, был предложен Джоффрионом [30]. Алгоритм основывается на том факте, что если лагранжева задача разрешима при $u = u^i$, то линейная опорная функция *) к графику h в точке u^i легко может быть построена. Это функция вида

$$h(u^i) + (u - u^i)g(x^i) = f(x^i) + ug(x^i), \quad (13)$$

где $x^i \in X(u^i)$, так что $g(x^i) \in \partial h(u^i)$. Пусть $g(x^1), \dots, g(x^r)$ — субградиенты h в $u^1, \dots, u^r$. Тогда, как видно из рис. 7.6,

*) Такая линейная функция, которая никогда не лежит ниже графика h и контактирует с ним в точке u^i .

естественно использовать вогнутую кусочно-линейную функцию

$$v^r(u) = \min_{1 \leq j \leq r} \{f(x^j) + ug(x^j)\} \quad (14)$$

в качестве аппроксимации h . Предположим, что S замкнуто и ограничено, и f и все g_i непрерывны на S , так что субградиенты существуют во всех точках $u \in E^m$ и $D = (E^m)^+$. Таким образом, двойственная задача может быть заменена следующей:

$$\max_{u \geq 0} v^r(u). \quad (15)$$

Эта задача в свою очередь эквивалентна линейной задаче относительно переменных (u, σ) :

$$\max \sigma, \quad (16)$$

$$f(x^j) + ug(x^j) \geq \sigma, \quad j = 1, \dots, r, \quad (17)$$

$$u \geq 0. \quad (18)$$

Итеративная процедура состоит из последовательности циклов, причем цикл r распадается на следующие шаги:

1. Решить задачу (16)–(18), получая оптимальное решение u^{r+1} .

2. Решить лагранжеву задачу

$$\min_{x \in S} (f(x) + u^{r+1}g(x)),$$

получая оптимальное решение x^{r+1} и субградиент $g(x^{r+1})$.

3. Добавить линейное ограничение вида (17), образованное в точке x^{r+1} , к задаче (16)–(18), заменить r на $r+1$ и вернуться к шагу 1.

Конечно, если существует более, чем одно оптимальное решение на шаге 2, то ограничения могут быть образованы для каждого из них и добавлены к задаче на шаге 3.

Описанная процедура является двойственной к обобщенному методу декомпозиции Данцига — Вулфа, описанному в § 3.3 и 3.4. Прямая задача имела вид

$$\min \{f(x)/g(x) \leq 0, x \in S\}, \quad (19)$$

и решалась итеративно с помощью линейной задачи

$$\min \sum_{j=1}^r \lambda_j f(x^j), \quad (20)$$

$$\sum_{j=1}^r \lambda_j g(x^j) \leq 0, \quad (21)$$

$$\sum_{j=1}^r \lambda_j = 1, \quad \lambda_j \geq 0. \quad (22)$$

Новый столбец для этой задачи генерировался на каждом цикле путем решения нелинейной подзадачи

$$\min_{x \in S} (f(x) + u^{r+1}g(x)),$$

где u^{r+1} — оптимальный вектор двойственных переменных, соответствующих ограничениям (21). Это и есть лагранжева задача, решаемая в нашем алгоритме на шаге 2. Далее, если σ — двойственная переменная, соответствующая ограничениям (22), то двойственная к (20) — (22) в точности совпадает с аппроксимационной задачей (16) — (18). При реальных вычислениях, вообще говоря, решалась бы (20) — (22), так как она содержит только $m+1$ ограничение. В этом случае схема аппроксимации касательными становится идентичной схеме генерации столбцов обобщенного линейного программирования.

Вследствие этих двойственных соотношений, если S и g являются выпуклыми, то, как показано в § 3.4, допустимая точка прямой задачи может быть получена на каждом цикле. Она дается формулой

$$\hat{x}^{r+1} = \sum_{j=1}^r \lambda_j^r x^j,$$

где λ_j^r решает (20) — (22). Для того чтобы обнаружить, когда $\hat{x}^{r+1}$ оптимально, заметим, что, так как u^{r+1} допустимо для двойственной, то $h(u^{r+1})$ является нижней границей максимального значения целевой функции. Легко видеть, что σ^{r+1} , максимальное значение целевой функции в (16) — (18), дает верхнюю границу, поскольку целевая функция этой задачи всегда является оценкой сверху функции h . Ясно, что если $h(u^{r+1}) = \sigma^{r+1}$, то u^{r+1} решает двойственную. Тогда по теореме 4.4—1 x^{r+1} является оптимальным. Практически вычисления были бы закончены, когда разность между σ^{r+1} и $h(u^{r+1})$ становится меньшей некоторой заранее назначенной погрешности. Заметим, что σ^r уменьшается монотонно, хотя для $h(u^r)$ монотонность не может быть гарантирована.

Как указывалось в § 3.4, Данциг доказал сходимость этой процедуры к оптимальному решению при $r \rightarrow +\infty$ при условии, что все столбцы в (20) — (22), генерируемые на предшествующих циклах, сохраняются. Двойственная интерпретация этого условия состоит в том, что недопустимо отбрасывание ни одной из линейных опорных функций, задающих функцию $v^r(u)$.

Как это уже имело место для многих алгоритмов предшествующих глав, линейная задача (16) — (18) может быть сформулирована многими различными способами, если исходная

нелинейная задача сепарабельна, т. е. имеет вид

$$\begin{aligned} \min \sum_{i=1}^p f_i(x_i), \\ \sum_{i=1}^p g^i(x_i) \leq 0, \\ x_i \in S_i, \quad i = 1, \dots, p. \end{aligned}$$

Тогда двойственная целевая функция записывается в форме

$$h(u) = \sum_{i=1}^p h_i(u),$$

где

$$h_i(u) = \min_{x_i \in S_i} (f_i(x_i) + u g^i(x_i)). \quad (23)$$

Аппроксимация касательными может быть проведена по отдельности к каждой из функций $h_i(u)$. Если $X_i(u)$ представляет собой множество x_i , в которых достигается минимум в (23), то $g^i(x_i)$ дают субградиенты h_i в u для любых $x_i \in X_i(u)$. Функция $f_i(x_i) + u g^i(x_i)$ является линейной опорной функцией к графику h_i в u , и если $x_i^j \in X_i(u_i^j)$, $j=1, \dots, r$, то аппроксимация касательными к h_i дается функцией

$$v_i^r(u) = \min_{1 \leq j \leq r} \{f_i(x_i^j) + u g^i(x_i^j)\}. \quad (24)$$

Двойственная проблема теперь заменяется на

$$\max \left\{ \sum_{i=1}^p v_i^r(u)/u \geq 0 \right\},$$

которая эквивалентна линейной задаче вида

$$\begin{aligned} \max \sum_{i=1}^p \sigma_i, \\ \sigma_i \leq f_i(x_i^j) + u g^i(x_i^j), \quad j = 1, \dots, r; \quad i = 1, \dots, p; \\ u \geq 0. \end{aligned}$$

В описанной задаче возникает возможность добавления p новых ограничений на каждом цикле, по одному от каждой подзадачи (23), в отличие от добавления единственного ограничения, которое имело место в задаче общего типа. Имеющийся опыт (см. § 3.2) работы с принципом декомпозиции Данцига — Вульфа позволяет предположить, что это может ускорить сходи-

мость. Если решается двойственная к указанной выше задаче, то это соответствует генерации только одного нового столбца для каждой подзадачи, как уже обсуждалось в § 3.4. Конечно, подмножество функций $h_i(u)$ может быть сгруппировано вместе, и аппроксимация касательными проведена применительно к сумме функций h_i , входящих в это подмножество. Это дает возможность добавлять любое число от 1 до p ограничений на каждом цикле.

В заключение этого параграфа отметим преимущество решения с помощью двойственного алгоритма, несвойственное прямым методам. Любой итеративный алгоритм будет производить последовательность значений u , $\{u^i\}$, сходящуюся к значению u^* , которое является решением двойственной. Доступной также является последовательность нижних границ $\{h(u^i)\}$ и точек $\{x^i\}$, таких, что $x^i \in X(u^i)$. Каждая такая точка в силу теоремы 7.3—3 решает модифицированную исходную задачу

$$\begin{aligned} \min f(x), \\ g_j(x) \leq y_j^i, \quad j = 1, \dots, m, \\ x \in S, \end{aligned}$$

где

$$y_j^i = g_j(x^i), \quad (u_j^i) > 0, \quad y_j^i \geq g_j(x^i), \quad (u_j^i) = 0.$$

Таким образом, мы получаем дополнительную информацию, касающуюся чувствительности решений к возмущениям правых частей. Поскольку входные данные для большинства математических моделей являются обычно только приближенными, то эта информация может быть столь же ценной, как и само по себе оптимальное решение.

Вычислительные соображения. В любом алгоритме, решающем двойственную задачу, большая часть времени затрачивается на решение лагранжевой задачи. Если исходная задача является невыпуклой, то эта задача, вообще говоря, будет невыпуклой также. Поскольку должен быть найден глобальный максимум, то это и может стать основным источником затруднений (см. [27] в качестве примера). С другой стороны, подзадачи могут быть иногда решены с помощью динамического программирования (см. § 7.8.3) или, если S конечно, с помощью некоторых эффективных процедур перебора. В этих случаях нахождение глобального решения гарантировано. В любом случае важно, что алгоритм, который решает лагранжевую задачу, делает эффективным использование информации, полученной из минимизации $L(x, u^i)$ при минимизации $L(x, u^{i+1})$. Так как u будет, вообще говоря, изменяться малыми шагами, то это должно значительно сократить время решения. В частности, можно использовать значение x^i (являющееся допустимым для лагранжевой задачи при $u = u^{i+1}$) в качестве исходной точки новой минимизации.

Относительно сравнительной эффективности метода аппроксимации касательными и метода градиентного спуска известно мало. Не следует преувеличивать простоту градиентных процедур с точки зрения построения вычислительных программ, поскольку даже простейшие методы требуют введения в качестве подпроцедуры одномерного поиска. Если это делается достаточно эффективно, то программа в целом может оказаться достаточно сложной. Метод аппроксимации касательными требует использования программ линейного программирования. Однако, даже если эти подпрограммы доступны, то при составлении общей программы требуется предусмотреть связи, позволяющие использовать результаты решения предшествующих подпрограмм для ускорения решения последующих.

§ 7.7. СПЕЦИАЛЬНЫЕ РЕЗУЛЬТАТЫ ДЛЯ ВЫПУКЛЫХ ЗАДАЧ

В этом параграфе будут представлены некоторые двойственные теоремы, справедливые при различных предположениях о выпуклости и регулярности. Один из главных результатов уже был доказан. Теорема 7.3—6 дает условия, гарантирующие, что седловая точка существует, в то время как теорема 7.4—3 устанавливает, что (x^0, u^0) является седловой точкой, если и только если x^0 дает решение прямой задачи, u^0 решение двойственной задачи и $f(x^0) = h(u^0)$. Комбинируя эти два результата, мы получаем следующую теорему.

Теорема 1 (теорема двойственности для выпуклых программ). Пусть S — выпуклое подмножество E^n , f — выпуклая функция, определенная на S , и $g(x)$ — m -вектор выпуклых функций, определенных на S . Предположим, что существует точка $x \in S$ такая, что $g(x) < 0$. Тогда, если x^0 — решение прямой задачи $\min \{f(x)/g(x) \leq 0, x \in S\}$, то существует вектор $u^0 \geq 0$, который решает двойственную, $\max \{h(u)/u \in D\}$, и $h(u^0) = f(x^0)$.

Фалк [2] показал, что можно получить некоторые дополнительные результаты, если принять следующие предположения:

Предположение 1. S замкнуто и выпукло, но не обязательно ограничено.

Предположение 2. f строго выпукла на S .

Предположение 3. Все компоненты g выпуклы на S .

Теорема 2 (Фалк [2]). При предположениях 1—3 D является открытым относительно $(E^m)^+$, т. е. D является пересечением $(E^m)^+$ с некоторым открытым множеством.

Доказательство. В последующем доказательстве будут использованы следующие обозначения: $N(x, \varepsilon)$ — открытая окрестность точки x радиуса ε , $\|x\|$ — евклидова норма x , $\bar{N}(x, \varepsilon)$ — замыкание N , $\partial N(x, \varepsilon)$ — граница N , $\emptyset$ — пустое множество.

Основная идея доказательства состоит в выборе точки $u^* \in D$ и нахождении некоторого $\delta > 0$ такого, что $u \geq 0$ и $u \in N(u^*, \delta) \Rightarrow u \in D$.

Пусть $u^* \in D$. Так как $L(x, u^*)$ строго выпукла на S , то она имеет минимум в единственной точке $x(u^*) \equiv x^*$. Пусть $N(x^*, \varepsilon)$ — окрестность x^* радиуса $\varepsilon > 0$, такая, что $S \cap \partial N(x^*, \varepsilon) \neq \emptyset$. Если не существует такого ε , то S представляет собой единственную точку x^* , и теорема тривиальна, так как $D = (E^m)^+$.

Пусть

$$\mu_1 = h(u^*) = L(x^*, u^*), \quad (1)$$

$$\mu_2 = \min \{L(x, u^*) / x \in S \cap \partial N(x^*, \varepsilon)\}. \quad (2)$$

Заметим, что допустимое множество $b(2)$ непусто при принятых ранее предположениях. Так как $L(x, u^*)$ строго выпукла, и $x^* \notin \partial N(x^*, \varepsilon)$, то

$$\mu_2 > \mu_1. \quad (3)$$

Теперь предположим, что может быть найдено некоторое $\delta > 0$ такое, что $u \geq 0$ и при $\|u^* - u\| < \delta$ выполняются следующие два неравенства:

$$|L(x^*, u) - \mu_1| < \frac{\mu_2 - \mu_1}{3}, \quad (4)$$

$$|L(x, u^*) - L(x, u)| < \frac{\mu_2 - \mu_1}{3}, \quad \forall x \in S \cap \partial N(x^*, \varepsilon). \quad (5)$$

Из неравенства (5) следует

$$L(x, u^*) - L(x, u) < \frac{\mu_2 - \mu_1}{3} \quad (6)$$

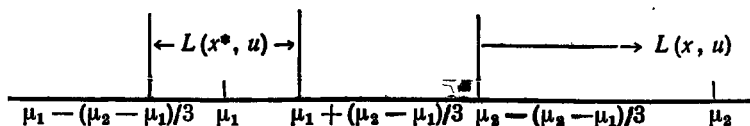
или

$$L(x, u) > L(x, u^*) - \frac{\mu_2 - \mu_1}{3}. \quad (7)$$

Поскольку $\mu_2 \leq L(x, u^*)$ для всех $x \in S \cap \partial N(x^*, \varepsilon)$, то

$$L(x, u) > \mu_2 - \frac{\mu_2 - \mu_1}{3}. \quad (8)$$

Неравенства (4)—(8) могут быть условно обрисованы следующим образом:



Очевидно, что

$$L(x, u) > L(x^*, u), \quad \forall x \in S \cap \partial N(x^*, \varepsilon). \quad (9)$$

Минимум функции $L(x, u)$ находится где-то в $S \cap \bar{N}(x^*, \varepsilon)$, поскольку это множество компактно. Минимум не может находиться в $S \cap \partial N(x^*, \varepsilon)$, так как в силу (9) существует точка x^* , не принадлежащая $S \cap \partial N(x^*, \varepsilon)$ и доставляющая меньшее значение $L(x, u)$, чем любая точка в $S \cap \partial N(x^*, \varepsilon)$. Таким образом, минимум находится внутри $S \cap \bar{N}(x^*, \varepsilon)$ и, следовательно, является локальным минимумом $L(x, u)$. Но $L(x, u)$ является строго выпуклой, так что этот минимум является и глобальным, т. е. $L(x, u)$ достигает глобального минимума в S для всех $u \geq 0$, таких, что $\|u^* - u\| < \delta$. Это и есть то, что мы желали показать первоначально, так что доказательство будет завершено, если мы покажем, что δ , указанное ранее, действительно может быть найдено. Сконструируем такое δ . Обратим внимание прежде всего на условие (4):

$$|L(x^*, u) - \mu_1| = |f(x^*) + ug(x^*) - f(x^*) - u^*g(x^*)|, \quad (10)$$

$$|L(x^*, u) - \mu_1| = |(u - u^*)g(x^*)|. \quad (11)$$

В силу неравенства Шварца имеем

$$|L(x^*, u) - \mu_1| \leq \|u - u^*\| \cdot \|g(x^*)\|. \quad (12)$$

Пусть

$$r_1 = \|g(x^*)\|. \quad (13)$$

Так как мы желаем получить $\|u - u^*\| \cdot r_1 < (\mu_2 - \mu_1)/3$, выберем

$$\|u - u^*\| < \frac{\mu_2 - \mu_1}{3r_1}, \quad r_1 > 0, \quad (14)$$

$$\|u - u^*\| < 1, \quad r_1 = 0 \quad (15)$$

(очевидно, что в (15) может быть подставлено любое положительное число). Тогда, в силу (12), условие (4) удовлетворяется.

Для того чтобы удовлетворить условию (5), заметим

$$|L(x, u^*) - L(x, u)| \leq \|(u^* - u)g(x)\|. \quad (16)$$

Вновь используя неравенство Шварца, получаем

$$|L(x, u^*) - L(x, u)| \leq \|u^* - u\| \cdot \|g(x)\|. \quad (17)$$

Обозначим

$$r_2 = \max \{\|g(x)\| / x \in S \cap \partial N(x^*, \varepsilon)\}. \quad (18)$$

Тогда

$$|L(x, u^*) - L(x, u)| \leq \|u^* - u\| \cdot r_2, \quad \forall x \in S \cap \partial N(x^*, \varepsilon). \quad (19)$$

Для того чтобы сделать правую часть меньшей, чем $(\mu_2 - \mu_1)/3$, выберем

$$\|u^* - u\| < \frac{\mu_2 - \mu_1}{3r_2}, \quad r_2 > 0, \quad (20)$$

$$\|u^* - u\| < 1, \quad r_2 = 0. \quad (21)$$

Вновь в (21) может быть подставлено любое положительное число. Комбинируя (14)—(15) и (20)—(21), приходим к следующим ограничениям для δ :

$$0 < \delta \leq \begin{cases} 1, & r_1 = 0, \quad r_2 = 0, \\ \frac{\mu_2 - \mu_1}{3r_1}, & r_1 > 0, \quad r_2 = 0, \\ \frac{\mu_2 - \mu_1}{3r_2}, & r_1 = 0, \quad r_2 > 0, \\ \min \left\{ \frac{\mu_2 - \mu_1}{3r_1}, \frac{\mu_2 - \mu_1}{3r_2} \right\}, & r_1 > 0, \quad r_2 > 0. \end{cases} \quad (22)$$

Если δ удовлетворяет (22), то из $u \geq 0$ и $\|u^* - u\| < \delta$ следует (4)—(5). Теорема доказана.

Следующая теорема, совместно с теоремой 7.4—2, показывает, что при предположениях 1—3 двойственная задача является задачей выпуклого программирования.

Теорема 3. При предположениях 1—3 D является выпуклым.

Доказательство. Выберем $u_1 \in D$, $u_2 \in D$. Пусть

$$u_3 = \alpha u_1 + (1 - \alpha) u_2, \quad 0 < \alpha < 1. \quad (23)$$

Так как

$$L(x, u_3) = \alpha L(x, u_1) + (1 - \alpha) L(x, u_2), \quad (24)$$

то

$$\inf_{x \in S} L(x, u_3) \geq \alpha \inf_{x \in S} L(x, u_1) + (1 - \alpha) \inf_{x \in S} L(x, u_2). \quad (25)$$

Однако оба инфимума в правой части (25) конечны, поскольку $L(x, u_3)$ ограничена снизу на S . Остается показать, что $\inf \{L(x, u_3) / x \in S\}$ достигается в некоторой точке S . Пусть

$$r_i = \inf \{L(x, u_i) / x \in S\}, \quad i = 1, 2, 3. \quad (26)$$

Положим

$$r > \max \{r_1, r_2, r_3\}. \quad (27)$$

Рассмотрим множества

$$R_1(r) = \{x \in S/L(x, u_1) \leq r\}. \quad (28)$$

Очевидно, что

$$\inf_{x \in S} L(x, u_3) = \inf_{x \in R_1(r)} L(x, u_3), \quad (29)$$

так что, если $L(x, u_3)$ достигает своего инфимума на S , то он должен достигаться и в $R_1(r)$. Так как $u_1, u_2 \in D$, множества $R_1(r_1)$ и $R_2(r_2)$ каждое состоит из единственной точки, а следовательно, ограничены по всем направлениям *).

Так как для любой выпуклой функции $L(x)$ все непустые множества вида $\{x/L(x) \leq r\}$ ограничены по тем же самым направлениям [13], то $R_1(r)$ и $R_2(r)$ являются ограниченными множествами. Однако, в силу (24),

$$L(x, u_3) \leq r \Rightarrow \text{либо } L(x, u_1) \leq r, \text{ либо } L(x, u_2) \leq r. \quad (30)$$

Таким образом,

$$R_3(r) \subseteq R_1(r) \cup R_2(r), \quad (31)$$

так что $R_3(r)$ является подмножеством ограниченного множества и, следовательно, ограничено. Так как $R_3(r)$ является также замкнутым, то оно компактно и $\inf\{L(x, u_3)/x \in S\}$ должен достигаться в $R_3(r)$. Это завершает доказательство.

Ряд других результатов также следует из предположений 1—3. В [2] доказано:

1. Минимизирующая точка $x(u)$ является непрерывной функцией u .

2. Двойственная целевая функция h имеет непрерывные частные производные всюду в $\text{int } D$.

Эти результаты сходны с результатами, полученными в § 7.5. Легко построить примеры, в которых вторые производные $h(u)$ не являются непрерывными всюду.

Согласно теоремам этого параграфа, при предположениях 1—3 двойственная задача не содержит ограничений, за исключением условий $u \geq 0$, и h дифференцируема всюду в $\text{int } D$. Таким образом, для решения двойственной задачи вновь подходит метод наискорейшего спуска. Такая вычислительная техника и была применена к решению задач со строго выпуклыми сепарабельными целевыми функциями и линейными ограничениями Фалком [21].

*) Говорят, что подмножество R в E^n ограничено по направлению u , если существует гиперплоскость с вектором внешней нормали u такая, что R содержится в полупространстве, задаваемом этой гиперплоскостью.

§ 7.8. ПРИЛОЖЕНИЯ

7.8.1. Задачи для связанных подсистем *)

Многие системы могут рассматриваться как совокупность взаимодействующих подсистем. Каждая подсистема описывается функциональной связью ее выхода с ее же входами. Взаимодействие подсистем проявляется в том, что выходы некоторых подсистем являются одновременно входами других. Пример такого рода показан на рис. 7.7. Каждый из связывающих потоков перенумерован и переменная x_i , связанная с потоком i ,

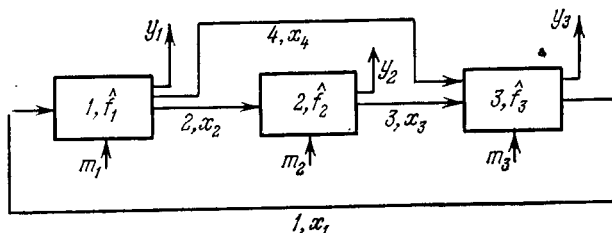


Рис. 7.7. Структура связей в системе

представляет количество или мощность потока. Стрелки, обозначенные y_i , представляют выходы подсистемы i , которые не идут к другим подсистемам, а направлены во «внешний» мир и предположительно имеют некоторое заданное значение. Переменные m_i являются управлениями или решающими переменными, связанными с подсистемой i .

Подобная структура может моделировать химическую систему, в которой подсистемами являются реакторы, сепараторы и т. п. В этом случае m_i представляют такие физические переменные, как давления, температуры и степени рецикла, в то время как x_i представляют собой мощности потоков и концентрации продуктов в них. Если рассматривается некоторая сложная экономическая система, то под подсистемами будут подразумеваться экономики различных стран, а под величинами x_i уровни экспорта и импорта. Для того чтобы дать математическое описание такой системы, введем два множества индексов для каждой подсистемы:

$$I(j) = \{i/\text{поток } i \text{ является входом подсистемы } j\}, \quad (1)$$

$$O(j) = \{i/\text{поток } i \text{ является выходом подсистемы } j\}. \quad (2)$$

*) См. [22]—[27] по поводу более ранних работ, касающихся этого вопроса.

Например, для рис. 7.7 имеем

$$\begin{aligned} I(1) &= \{1\}, & O(1) &= \{2, 4\}, \\ I(2) &= \{2\}, & O(2) &= \{3\}, \\ I(3) &= \{3, 4\}, & O(3) &= \{1\}. \end{aligned}$$

Пусть q_j — вектор, содержащий все входы подсистемы j :

$$q_j = (x_{k_1}, \dots, x_{k_r} | m_j), \quad k_i \in I(j). \quad (3)$$

Например, для системы на рис. 7.7 $q_3 = (x_3, x_4 | m_3)$. Каждая подсистема может быть теперь описана заданием ее выходов, как функций входов

$$x_i = t_i(q_j), \quad i \in O(j), \quad j = 1, \dots, r, \quad (4)$$

$$y_j = s_j(q_j), \quad j = 1, \dots, r, \quad (5)$$

где t_i — функция q_j , и s_j — вектор таких функций.

Свяжем с подсистемой j целевую функцию $\hat{f}_j(q_j, y_j)$, представляющую выход оперирующей подсистемы j . Например, $\hat{f}_j$ может быть стоимостью продуктов, составляющих вектор y_j , за вычетом затрат на действие подсистемы j , когда ее входы находятся на уровне q_j . Обозначение упрощается, если мы используем (5) для исключения y_j из $\hat{f}_j$, представляя их просто функциями q_j :

$$f_j(q_j) = \hat{f}_j(q_j, s_j(q_j)). \quad (6)$$

Общий выход f от действия системы в целом предполагается суммой выходов подсистем:

$$f = \sum_{j=1}^r f_j(q_j). \quad (7)$$

Рассматриваемая здесь проблема состоит в выборе векторов q_j таких, что они максимизируют (7) при ограничениях на связях (4) и каких-нибудь других ограничениях, характеризующих области изменения векторов q_j . Такие ограничения могут возникать, например, в силу наличия ограничений на управляющие переменные m_j и на межсистемные потоки x_i . Предполагается, что эти ограничения заданы независимо для каждой подсистемы, т. е. они могут быть представлены следующим образом:

$$q_j \in S_j, \quad j = 1, \dots, r. \quad (8)$$

Тогда задача оптимизации связанной системы может быть записана в виде

$$\max \sum_{j=1}^r f_j(q_j), \quad (9)$$

$$x_i = t_i(q_j), \quad i \in O(j), \quad j = 1, \dots, r, \} \text{ограничений}, \quad (10)$$

$$q_j \in S_j, \quad j = 1, \dots, r. \quad (11)$$

Предположим, что эта проблема имеет оптимальное решение.

Проблему (9)—(11) можно попытаться решать двумя путями. Первый состоит в том, чтобы рассматривать в качестве независимых только переменные m_j и использовать равенство (10) для исключения x_i . Так как, вообще говоря, каждое x_i зависит от всех векторов m_j , то результат исключения будет представлять собой задачу математического программирования относительно вектора $m = (m_1, \dots, m_r)$. Хотя этот «интегральный» подход сокращает число независимых переменных, он приводит к серьезным трудностям, особенно в случаях, когда имеются обратные связи, или рециклы в связанной системе. Например, для системы рис. 7.7 поток 1 является потоком обратной связи от подсистемы 3 к подсистеме 1. Вследствие этого, уравнения (10) являются неявными; x_i не могут быть найдены последовательно. Система уравнений поэтому должна решаться численно, поскольку получение решений в явной форме возможно только в простейших случаях. Таким образом, для того, чтобы оценить значение целевой функции для заданного вектора m , приходится использовать итеративный процесс. Несмотря на указанные трудности, первый подход широко используется и часто бывает успешным. Альтернативой к нему является рассмотрение вектора $q = (q_1, \dots, q_r)$, как вектора независимых переменных, и использование штрафных функций для учета ограничений (см. [28], [29]).

Другой альтернативой, рассматриваемой далее детально, является решение исходной задачи путем декомпозиции ее на подзадачи меньшего размера. Для того, чтобы показать, как это может быть сделано, проведем первоначально некоторые правдоподобные рассуждения, справедливость которых будет обоснована позже. Проблема не может быть разложена на подпроблемы, если не произвести разделение исходной системы на подсистемы некоторым способом. Представим себе, что такое разделение произведено путем разрезания связей между подсистемами, так что в (10) $x_i \neq t_i(q_j)$. Далее рассмотрим каждую подсистему, как находящуюся под независимым управлением менеджера, который продает выходные продукты подсистемы

другим подсистемам и покупает свои входные продукты у других подсистем. Свяжем с каждым i -м подблоком цену u_i , по которой производятся взаимные купли — продажи.

Менеджер подсистемы j рассматривает цены как фиксированные, и максимизирует чистый доход своей подсистемы. Этот чистый доход

$$\tilde{f}_j(q_j; u) = f_j(q_j) + \sum_{i \in O(j)} u_i t_i(q_j) - \sum_{i \in I(j)} u_i x_i. \quad (12)$$

Первое слагаемое, f_j , есть чистый доход, получаемый при производстве выходных продуктов y_j при использовании входов m_j . Второе слагаемое есть доход, получаемый от продажи выходных продуктов t_i другим подсистемам. Наконец, последний член представляет собой затраты на покупку входных продуктов x_i у других подсистем. j -я подпроблема теперь принимает вид

$$\max \tilde{f}_j(q_j, u), \quad (13)$$

$$q_j \in S_j. \quad (14)$$

Задача настройки цен будет рассматриваться как задача координатора верхнего уровня. Поскольку входы $x_i(u)$ и выходы $t_i(u)$ определяются менеджерами независимо, то необязательно выполнение условия равенства на разрезах, и очевидно, что координатор должен настроить цены так, чтобы свести различие в потоках к нулю. При рассмотрении любого из потоков значение переменной на выходной стороне разреза, $t_i(u)$, может считаться поставкой, а значение переменной на входной стороне $x_i(u)$ — спросом. Различие между ними есть просто разница между спросом и поставками, т. е. избыточный спрос $e_i(u)$:

$$e_i(u) = x_i(u) - t_i(q_j(u)), \quad i \in O(j). \quad (15)$$

В этом смысле алгоритм

$$\frac{du}{dt} = e_i(u), \quad i = 1, \dots, m \quad (16)$$

может быть трактован как требование роста цен при превышении спроса над поставками и снижение цен в противном случае. Это правило настройки цен может служить в качестве координирующего алгоритма, хотя, как мы вскоре покажем, существуют и более эффективные законы.

Математическое обоснование всего сказанного следует из результатов предшествующих параграфов. Лагранжева функция для задачи (9) — (11)

$$L = \sum_j f_j(q_j) + \sum_j \sum_{i \in O(j)} u_i (t_i(q_j) - x_i). \quad (17)$$

Слагаемое $\sum_j \sum_{i \in O(j)} u_i x_i$ является просто суммой всех возможных слагаемых вида $u_i x_i$, и может быть переписано в форме

$$\sum_j \sum_{i \in O(j)} u_i x_i = \sum_{i: i \in I(j)} u_i x_i. \quad (18)$$

Тогда L записывается в виде

$$L = \sum_j \left[f_j(q_j) + \sum_{i \in O(j)} u_i x_i(q_j) - \sum_{i \in I(j)} u_i x_i \right]. \quad (19)$$

Это выражение сепарабельно по отношению к векторам q_j при фиксированном u . Величины, заключенные в скобки (7) — (12), есть целевые функции или функции чистого дохода для j -й подпроблемы $f_j(q_j; u)$. Двойственная к исходной задаче (9) — (11) такова:

$$\min_{u \in D} h(u),$$

где

$$h(u) = \max_{q_1 \in S_1, \dots, q_r \in S_r} \left(\sum_{j=1}^r \tilde{f}_j(q_j; u) \right), \quad (20)$$

$$D = \{u/\text{для которых max в (20) существует}\}. \quad (21)$$

Задача сепарабельна и может быть решена путем решения ряда подзадач вида

$$\max_{q_j \in S_j} \tilde{f}_j(q_j; u), \quad (22)$$

которая, очевидно, совпадает с j -й подпроблемой (13) — (14), построенной ранее из эвристических соображений.

Двойственная функция h представляет собой сумму максимальных значений целевых функций подпроблем при заданном u :

$$h(u) = \sum_j (\max_{q_j \in S_j} \tilde{f}_j(q_j; u)). \quad (23)$$

По теореме 7.4—2 h является выпуклой на любом выпуклом подмножестве D (заметим, что роль максимума и минимума переставлена по сравнению с предшествующими параграфами). Все другие результаты предшествующих параграфов здесь вполне приложимы. Пусть

$$Q_j(u) = \{q_j/q_j \text{ — решение подпроблемы } j \text{ с множителями } u\}, \quad (24)$$

$$Q(u) = Q_1(u) \times Q_2(u) \times \dots \times Q_r(u). \quad (25)$$

Тогда, поскольку ограничения, введенные в функцию Лагранжа, являлись равенствами, теорема 7.3—2 может быть представлена в следующей форме:

Теорема 1. *Предположим, что D является непустым, и рассмотрим некоторый вектор множителей $u^0 \in D$. Тогда вектор $q \in Q(u^0)$ дает решение исходной задачи, если q является допустимым для нее, т. е. удовлетворяет равенствам (10).*

Удовлетворение (10) означает, конечно, что ошибки на связях, т. е. избыток спроса (15) $e_i(u)$, обращается в нуль для всех i . Это обосновывает интуитивное представление о том, что экономическая система с механизмом «совершенной конкуренции», построенная выше, приходит к глобальному оптимуму, если и только если спрос и предложение равны на всех рынках.

Теорема 7.4—3, связывающая седловые точки и двойственные задачи, преобразуется здесь к следующей форме.

Теорема 2. *Вектор множителей $u^0 \in D$ обладает тем свойством, что некоторое значение $q \in Q(u^0)$ дает решение прямой задачи, если и только если u^0 дает решение двойственной и $\min h = h(u^0) = \max f$.*

Если все множества S_i являются замкнутыми и ограниченными, и все функции f_i и t_i являются непрерывными, то $D = E^m$, и двойственная задача является задачей безусловной минимизации. Решение двойственной в этом случае может разыскаться с помощью алгоритмов наискорейшего спуска или аппроксимации касательными.

Все результаты § 7.5, относящиеся к производным h , здесь также применимы. В частности, полагая $D = E^m$, получаем, что h дифференцируема в точке u^* , если $Q(u^*)$ состоит из единственной точки. В этом случае ее частные производные равны

$$\left. \frac{\partial h}{\partial u_i} \right|_{u^*} = t_i(q_i(u^*)) - x_i(u^*), \quad i \in O(j). \quad (26)$$

Это выражение представляет (с обратным знаком) избыток спроса:

$$\left. \frac{\partial h}{\partial u_i} \right|_{u^*} = -e^i(u^*). \quad (27)$$

Таким образом, алгоритм $du_i/dt = e_i$ является просто непрерывным вариантом метода наискорейшего спуска, примененного к h . Практически полезнее использовать более эффективные методы минимизации при отсутствии ограничений, описанные, например, в [28]. Если h недифференцируема в точке u^* , то вектор

$$-e(u^*) = -(e_1(u^*), \dots, e_m(u^*))$$

является субградиентом h в u^* и может быть использован в методе аппроксимации касательными.

7.8.2. Пример — оптимальное управление дискретной по времени динамической системой

Множество взаимодействующих подсистем может быть связано не в пространстве, а во времени, так, как это имеет место в дискретной по времени динамической системе, показанной на рис. 7.8. Величина x_i есть n -вектор, представляющий состояние процесса, в то время как m_i есть m -вектор управлений, причем

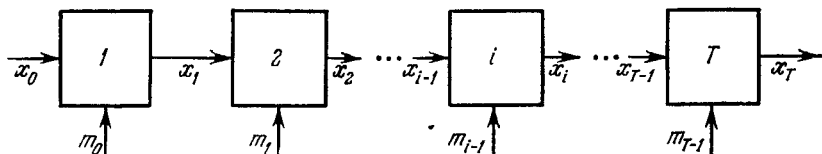


Рис. 7.8. Динамическая система, дискретная по времени.

и то и другое относится к моменту времени $t=i$. Состояние эволюционирует во времени согласно разностному уравнению:

$$x_i = g^i(x_{i-1}, m_{i-1}, i), \quad x_0 \text{ — дано, } i = 1, \dots, T. \quad (28)$$

Полный выход (доход) от действия системы за время T предполагается суммой доходов на каждом этапе:

$$\max \sum_{i=0}^{T-1} f_i(m_i, x_i, i). \quad (29)$$

Во многих случаях требуется, чтобы переменные m_i и x_i удовлетворяли некоторым ограничениям типа неравенств, которые часто имеют форму ограничений сверху и снизу. Представим эти ограничения в следующем виде:

$$(m_i, x_i) \in S_i, \quad i = 0, \dots, T-1. \quad (30)$$

В дополнение могут быть наложены требования на то, чтобы вектор конечного состояния x_T лежал в некотором многообразии:

$$\psi(x_T) = 0, \quad (31)$$

где ψ есть p -вектор функций, $p < n$. Математическая проблема состоит в максимизации (29) при ограничениях (28), (30) и (31).

В качестве специального примера рассмотрим следующую задачу:

$$\max J = - \sum_{i=1}^s x'_i x_i - \sum_{i=0}^s m'_i m_i, \quad (32)$$

где x_i и m_i — двухкомпонентные векторы, удовлетворяющие ограничениям *):

$$x_i = g(x_{i-1}, m_{i-1}), \quad i = 1, \dots, 9, \quad (33)$$

$$x_0 = (-1, 1)$$

$$|x_i| \leq 1, \quad i = 1, \dots, 8, \quad (34)$$

$$|m_i| \leq 1, \quad i = 0, \dots, 8, \quad (35)$$

$$x_{1,9} = \frac{1}{2}, \quad (36)$$

причем g — двумерный вектор функций

$$g_1(x_{i-1}, m_{i-1}) = -x_{1,i-1} + x_{2,i-1} + m_{1,i-1}, \quad (37)$$

$$g_2(x_{i-1}, m_{i-1}) = x_{1,i-1} - x_{2,i-1} + m_{2,i-1}. \quad (38)$$

Этот пример представляет линейную систему с квадратичной целевой функцией, ограниченными переменными состояния и управляющими переменными и с одним конечным ограничением. Сконструированная целевая функция обеспечивает малость отклонений переменных состояния от нуля и одновременно малость энергетических затрат на управление; иначе говоря, это есть задача проектирования регулятора с ограниченными управляющими воздействиями.

Лагранжиан для этой задачи таков:

$$L = f + \sum_{i=1}^{K_0} u_i (g(x_{i-1}, m_{i-1})_i - x_i) + p_{1,9} \left(x_{1,9} - \frac{1}{2} \right), \quad (39)$$

и задача Лагранжа состоит в максимизации L при ограничениях (34) — (35). Поскольку L линейна относительно x_9 , максимум не существует, если коэффициенты при x_9 не обращаются в нуль. Однако обращение в нуль этих коэффициентов является одним из необходимых условий оптимальности:

$$\frac{\partial L}{\partial x_{1,9}} = -u_{1,9} + p_{1,9} = 0, \quad (40)$$

$$\frac{\partial L}{\partial x_{2,9}} = -u_{2,9} = 0. \quad (41)$$

Эти условия служат для исключения $u_{2,9}$ и $p_{1,9}$ из L . Сепарабельность L тогда приводит к следующим подзадачам.

*) Уравнения (34) и (35) означают, что каждая компонента x_i и m_i ограничена.

Подзадача 0:

$$\left. \begin{aligned} \max \tilde{f}_0 &= -m'_0 m_0 + u_1 g(m_0, x_0), \\ x_0 &= (-1, 1), \quad |m_0| \leq 1. \end{aligned} \right\} \quad (42)$$

Подзадачи $i, i=1, \dots, 7$:

$$\left. \begin{aligned} \max \tilde{f}_i &= -m'_i m_i - x'_i x_i + u_{i+1} g(m_i, x_i) - u_i x_i, \\ |x_i| &\leq 1, \quad |m_i| \leq 1. \end{aligned} \right\} \quad (43)$$

Подзадача 8:

$$\left. \begin{aligned} \max \tilde{f}_8 &= -m'_8 m_8 - x'_8 x_8 + u_{1,0} g_1(m_8, x_8) - u_8 x_8, \\ |m_8| &\leq 1, \quad |x_8| \leq 1. \end{aligned} \right\} \quad (44)$$

Двойственная целевая функция h

$$h(u) = \sum_{i=1}^9 (\max \tilde{f}_i) - \frac{1}{2} u_{1,0}. \quad (45)$$

Поскольку подзадачи имеют единственные оптимальные решения для любого вектора $u = (u_1 | \dots | u_9)$, $h(u)$ определена для всех u и всюду дифференцируема. Таким образом, по теореме 7.5—6, если u^0

решает двойственную задачу, то решения подпроблем для $u = u^0$ решают прямую. Так как целевая функция каждой подзадачи сепарабельна, то эти подзадачи могут быть решены в замкнутой форме. Полученные таким образом решения являются линейными функциями множителей u , так что двойственная функция h квадратична с положительно полуопределенным гессианом. Известно, что ([28]) методы гра-

диентного спуска в модификации Флетчера и Пауэлла или Флетчера и Ривса позволяют найти минимум такой функции за число шагов, не превышающее число переменных. Таким образом, применение любой из этих процедур к задаче минимизации h дает конечный алгоритм для решения прямой задачи.

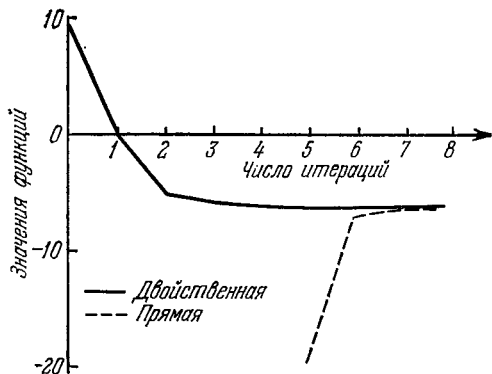


Рис. 7.9. Зависимость значений целевых функций прямой и двойственной задачи (для 9-этапной последовательной системы) от числа итераций.

В ходе реализации этих вычислений можно получать допустимые решения прямой задачи, а следовательно, получать границу для оптимума этой задачи. Для этого достаточно просто подставить решение подзадач в разностные уравнения (33) и вычислить соответствующие значения x_i . Если эти x_i удовлетворяют ограничениям (34), то пара $(\{x_i, m_i\})$ является допустимым решением исходной задачи. Из рис. 7.9 видно, как изменяются в ходе итеративного процесса значения целевых функций прямой и двойственной задачи для проблемы (32) — (36), если допустимые точки прямой вычисляются, как было сказано выше [20]. Для минимизации h был использован метод Флетчера и Пауэлла. Этот метод позволил свести ∇h к нулю за 8 итераций, что существенно меньше, чем теоретическая гарантия в [17] (число компонент u). Допустимые точки для прямой задачи были получены на итерации 5.

7.8.3. Задачи, в которых допустимое множество конечно: многопродуктовые задачи распределения плана

Декомпозиция особенно привлекательна для решения сепарабельных задач, включающих оптимизацию на конечном, но достаточно большом множестве, поскольку дискретная оптимизация при наличии только нескольких переменных легко выполняется с помощью динамического программирования или другой комбинаторной техники. Примером такой задачи является задача, рассмотренная в § 3.2. Там, предполагая, что число продуктов много больше, чем число периодов времени, мы нашли приближенное решение с использованием линейного программирования. Алгоритм включал решение множества однопродуктовых задач при использовании цен, генерируемых в координирующей линейной задаче. Здесь при использовании минимаксной двойственной задачи нелинейного программирования мы попытаемся решить эту задачу прямо. Будут решаться те же самые подзадачи, однако цены будут генерироваться путем максимизации двойственной целевой функции. Любые полученные решения окажутся целочисленными независимо от числа продуктов и числа периодов. Однако не всегда могут быть получены оптимальные или даже допустимые решения.

Сама задача была описана в § 3.2. Для удобства перепишем здесь обозначения и математическую постановку.

Пусть

m_{it} — число машин, используемых для производства продукта i в периоде t ;

$m_i = (m_{i1}, \dots, m_{iT})'$ — программа для продукта i , развернутая во времени;

$s_t(m_t)$ — затраты, связанные с программой m_t (замечим, что s_t могут быть вычислены при данных m_t только, если затраты независимы по всем продуктам);

d_{it} — спрос на продукт i в периоде времени t , предполагаемый известным;

y_{it} — запас продукта i в конце периода времени t ;

p_s — затраты на увеличение производства на единицу;

n_i — число единиц оборудования, доступных для производства продукта i ;

k_i — производительность машин;

b_t — число машин, доступных в периоде t .

Стоимость переналадки, связанной с использованием программы m_t ,

$$s_t(m_t) = p_s \sum_{i=1}^T (m_{it} - m_{i,t-1})_+, \quad m_{i0} \text{ — дано,} \quad (46)$$

где

$$(x)_+ = \begin{cases} x, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (47)$$

Затраты, связанные с запасами,

$$\gamma_t(m_t) = \sum_{i=1}^T \gamma_{it}(y_{it}), \quad (48)$$

где γ_{it} представляют издержки хранения при $y_{it} > 0$ и издержки неудовлетворения спроса при $y_{it} < 0$, и предполагаются линейными в каждом из указанных диапазонов. Полные затраты, отвечающие программе m_t ,

$$c_t(m_t) = s_t(m_t) + \gamma_t(m_t). \quad (49)$$

Задача состоит в выборе $m_1, \dots, m_T$, минимизирующих

$$C = \sum_{t=1}^T c_t(m_t) \quad (50)$$

при ограничениях

$$y_{it} = y_{i,t-1} + k_i m_{it} - d_{it}, \quad t = 1, \dots, T, \quad y_{i0} \text{ — даны,} \quad (51)$$

$$(y_{it})_{\max} \geq y_{it} \geq (y_{it})_{\min}, \quad (52)$$

$$m_{it} \leq n_i, \quad t = 1, \dots, T, \quad (53)$$

$$m_{it} \geq 0, \text{ — целое,} \quad (54)$$

$$\sum_{i=1}^I m_{it} \leq b_t, \quad t = 1, \dots, T. \quad (55)$$

Задача (50) — (55) является дискретной вследствие наличия ограничения (54). Соотношение (51) описывает изменения уровней запасов продукта i , ограничение (52) задает ограничения на уровень запаса, а (55) требует, чтобы число машин, используемых в каждом периоде времени t , не превышало общего доступного числа b_t .

Решение этой задачи весьма затруднительно при больших I и T . Ее можно было бы разложить на I отдельных подзадач, если бы отсутствовало ограничение (55). Учитывая это, образуем лагранжиан:

$$L(m, u) = \sum_{i=1}^I c_i + \sum_{t=1}^T u_t \left(\sum_{i=1}^I m_{it} - b_t \right), \quad u_t \geq 0, \quad (56)$$

$$L(m, u) = \sum_{i=1}^I \left(c_i(m_i) + \sum_{t=1}^T u_t m_{it} \right) - \sum_{t=1}^T u_t b_t, \quad (57)$$

и обозначим

$$S_i = \{m_i/m_i \text{ удовлетворяет (51) — (54)}\}. \quad (58)$$

Двойственная задача

$$\max_{u \geq 0} \min_{m_i \in S_i, \dots, m_I \in S_I} L(m, u). \quad (59)$$

Вновь, поскольку лагранжиан сепарабелен, внутренняя минимизация приводит к подзадачам вида

$$\min \{c_i(m_i) + u m_i\}, \quad (60)$$

$$m_i \in S_i. \quad (61)$$

Двойственная к ней задача:

$$\max_{u \geq 0} h(u),$$

где

$$h(u) = \sum_{i=1}^I h_i(u) - \sum_{t=1}^T u_t b_t. \quad (62)$$

Здесь $h_i(u)$ — минимальные значения целевой функции в задачах (60). Частные производные двойственной целевой функции, если они существуют, даются формулами

$$\frac{\partial h}{\partial u_t} = \sum_{i=1}^I m_{it}(u) - b_t. \quad (63)$$

Процедура решения состоит в выборе некоторого начального значения $u_0 \geq 0$ ($u_0 = 0$ является хорошим исходным вариантом), в решении подзадач для этого u , оценке двойственной функции

и ее градиента и, например, изменении u_0 путем движения в направлении градиента, увеличивая тем самым h . Заметим, что множители u_i появляются в (60), как цены для m_i . Градиентный алгоритм увеличивает цены u_i , если (63) положительно, т. е. если число машин, используемых для выпуска всех продуктов в период времени t , больше, чем допустимое b_i . Таким образом, штрафуются каждая из подсистем, если они коллективно используют больше машин, чем это допустимо.

Отметим, что, поскольку множества S_i являются конечными, минимумы (60) существуют, и, таким образом, двойственные функции существуют и являются вогнутыми для всех $u \geq 0$. Дополнительные свойства h следуют из конечности множеств S_i . Задавая произвольные $u_0 \geq 0$, мы можем представить себе, что элементы m_i в S_i перемещаются вдоль линии в соответствии с целевыми функциями подзадач (60), как это показано на рис. 7.10. Самая нижняя точка представляет собой минимизирующее решение для $u = u_0$ и мы можем предположить, что если каждое оптимальное решение $m_i^*(u_0)$ единственно, то h дифференцируема в u_0 . Рассмотрим изменение u_0 при движении по направлению s , т. е.

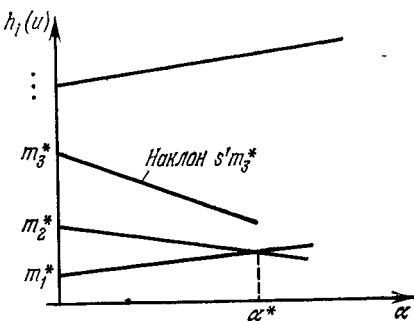


Рис. 7.10.

$$u = u_0 + \alpha s, \quad \alpha \geq 0. \quad (64)$$

При увеличении α функция (60) при каждом фиксированном m_i^* принимает вид

$$h_i^*(\alpha) = (c_i(m_i^*) + u_0^* m_i^*) + \alpha (s' m_i^*), \quad (65)$$

т. е. линейна по α при фиксированных s, m_i^* . Так как каждая из h_i^* изменяется по схеме, показанной на рис. 7.10, оптимальное решение остается неизменным вплоть до некоторой точки $\alpha^* > 0$. Таким образом, целевая функция подзадачи $h_i(u(\alpha))$ линейна по α при $0 \leq \alpha \leq \alpha^*$ для любого направления s . Иначе говоря, двойственная функция линейна, и можно сдвинуться на положительное расстояние по любому направлению в пространстве u вплоть до того момента, когда оптимальное решение некоторой подзадачи изменится. Это означает, что двойственная функция образуется из кусков пересекающихся гиперплоскостей. Одна из возможных стратегий максимизации такой функции состоит в

том, чтобы следовать по градиенту вдоль данной гиперплоскости вплоть до того, как достигается другая гиперплоскость, после чего вычислить градиент и т. д. (ясно, что градиент существует, за исключением точек пересечения двух гиперплоскостей, где он теряет непрерывность). Метод аппроксимации касательными здесь, возможно, более эффективен, так как он не зависит от дифференцируемости h , и так как здесь h является кусочно-линейной функцией. Можно подозревать, что метод

Таблица 7. 8.1

Продукт	y_{i0}	m_{i0}	$(y_i)_{\min}$	$(y_i)_{\max}$	n_i
1	500	0	50	6000	4
2	4000	3	500	6000	4
3	300	0	100	2000	1
4	500	0	100	2000	1
5	5000	4	2500	8000	7
6	2500	3	1000	5000	3
7	1500	0	100	4000	1
8	0	0	0	5000	3
9	5500	1	500	9000	1
10	6000	3	1000	9000	5
11	200	1	0	4000	2
12	100	0	100	6000	2
13	0	0	0	4000	1
14	5000	6	2500	8000	7
15	0	2	600	4000	4

аппроксимации касательными будет находить оптимум двойственной задачи за конечное число шагов, хотя мы и не имеем доказательства этого.

Примеры. Не существует теоретической гарантии, что дискретные проблемы рассмотренного здесь типа могут быть решены при использовании двойственности. Поэтому были предприняты попытки решения отдельных задач для того, чтобы удостовериться в полезности этого подхода. Данные для таких задач приведены ниже в таблицах 7.8.1 и 7.8.2,

$$T = 6, I = 15, \quad \gamma_{ii}(y_{ii}) = 0,0015y_{ii}, \quad y_{ii} \geq 0,$$

$$p_s = 15, 20, \quad k_i = k = 900.$$

Если каждая $c_i(m_i)$ минимизируется отдельно при ограничениях $m_i \in S_i$, и если результирующее решение удовлетворяет (55), то совокупность этих решений оптимальна для задачи (50) — (55). Общее использование машин $\sum_i m_{ii}$ для этих отдельных

решений оказывается следующим:

Период	1	2	3	4	5	6
Полное использование	27	28	31	33	36	28

Число машин, доступных в каждом периоде t , b_t , выбрано одинаковым для всех t , b , и изменялось при решении четырех различных задач:

Задача	1	2	3	4
Доступное число машин	35	34	33	32

Таким образом, задача 1 была сконструирована так, чтобы общее число машин, используемое при решении без ограничений, было больше 35 только в периоде 5. С уменьшением b область решения становится меньше, и, следовательно, решение, полученное без учета ограничений, становится в меньшей степени недопустимым.

Таблица 7.8.2

Спрос (в сотнях ед./неделя)

продукт	1-я неделя	2-я неделя	3-я неделя	4-я неделя	5-я неделя	6-я неделя
1	25	28	24	22	20	18
2	15	20	20	18	16	15
3	0,5	1	0,5	0,5	1	1
4	1	1	1	1	1	1
5	40	40	50	60	70	80
6	30	30	30	25	25	25
7	3	3	4	4	4	4
8	25	28	26	25	26	28
9	5	5	5	5	5	5
10	40	42	43	45	40	35
11	10	12	14	16	18	20
12	0	0	0	40	40	0
13	0	0	0	0	3	14
14	45	45	46	48	50	52
15	26	27	24	20	18	12

Таблица 7.8.3

Оптимальные решения

Проблема	Оптимум двойственной	Оптимум прямой (если он достигается)
1	2877,40	2877,40
2	2885,14	—
3	2910,88	2910,88
4	2947,60	—

При решении всех задач использовался простейший метод наискорейшего спуска при минимизации h , модифицированный для учета ограничений неотрицательности $u_i \geq 0$. Результаты решений представлены в таблицах 7.8.3 — 7.8.6. Оптимальное решение удалось получить в задачах 1 и 3; это следует из того, что удовлетворялись ограничения по доступному парку машин и условия дополняющей нежесткости. В задачах 2 и 4 ограни-

Таблица 7.8.4

Оптимальные множители

Проблема	u_1	u_2	u_3	u_4	u_5	u_6
1	0	0	0	0	6	0
2	0	0	0	1,2495	7,87495	0
3	0	0	0	3	9	0
4	0	0	0,045	6,78	1,3275	0

Таблица 7.8.5

Недоиспользование машин в оптимуме двойственной: $\sum_i m_{ii} - b$

Проблема	Время }					
	1	2	3	4	5	6
1	-8	-6	-4	-2	0	-7
2	-8	-6	-2	+2*)	-2	-5
3	-6	-4	-2	0	0	-3
4	-4	-3	-1	0	+1*)	-2

*) Плюс указывает на нарушение ограничений.

Таблица 7.8.6

Сводка результатов последовательных итераций для $b = 33$

Итерация	Множители						Недоиспользование машин						Двойственные функции
	u_1	u_2	u_3	u_4	u_5	u_6	1	2	3	4	5	6	
1	0	0	0	0	0	0	-6	-5	-2	0	3	-5	2867,14
2	0	0	0	0	3	0	-6	-5	-2	0	3	-5	2885,14
3	0	0	0	0	6	0	-6	-4	-2	0	2	-5	2901,40
4	0	0	0	0	8	0	-6	-4	-2	3	-1	-5	2901,90
5	0	0	0	3	7	0	-6	-4	-2	0	2	-5	2905,40
6	0	0	0	3	9	0	-6	-4	-2	0	0	-3	2910,88

чения выполняются не во всех периодах времени, однако, недопустимость проявляется только в одном периоде, и при том величина невязки мала. Таким образом, полученное решение остается полезным.

Задачи

1. Доказать теорему 7.3-4.
2. Доказать теорему 7.3-5.
3. Доказать: если x^i — решение лагранжевой задачи при $u = u^i$, $i=1,2$, и если

$$\begin{aligned} g_j(x^2) &= g_j(x^1), \quad j \neq k, \\ g_k(x^2) &> g_k(x^1), \end{aligned}$$

то

$$u_k^1 \geq \frac{f(x^1) - f(x^2)}{g_k(x^2) - g_k(x^1)} \geq u_k^2.$$

4. Для задачи

$$\begin{aligned} \min f(x), \\ g_i(x) \leq b_i, \quad i = 1, \dots, m, \end{aligned}$$

где f и все g_i дифференцируемы, предположить, что оптимальное решение, $x(b)$, существует при всех $b = (b_1, \dots, b_m)$ в некоторой окрестности точки b^* , и что $x(b)$ дифференцируем для всех b в этой окрестности. Далее предположить, что существует множество множителей таких, что условия Куна — Таккера удовлетворяются при $b = b^*$.

Показать, что

$$\left. \frac{\partial(\min f)}{\partial b_i} \right|_{b^*} = u_i^*, \quad i = 1, \dots, m.$$

Дать интерпретацию этого результата на языке цен и показать его полезность в анализе чувствительности. Также показать, что результат задачи 3 стремится к указанному выше соотношению в пределе. (В главе VIII дан усиленный вариант этого результата, справедливый для выпуклых задач.)

5. Предполагая: что f , g и S выпуклы, и что S компактно, показать, что $w(y)$ в (7.4-47) имеет выпуклую область определения и выпукла на этой области.

6. Доказать, что для прямой задачи (7.4-5) имеет место

$$\min f = \min_{x \in S} \sup_{u \geq 0} L(x, u).$$

7. Изобразить график функции $w(y)$ и вид множеств R для примеров 1 и 2 в § 7.4. Обсудить связь между этими множествами и седловой точкой в этих примерах.

8. Рассмотреть задачу квадратичного программирования

$$\min \left(\frac{1}{2} x' D x + d' x \right)$$

при ограничениях

$$A x \geq b,$$

где D симметричная и положительно определенная матрица. Показать, что

двойственная к этой задаче имеет вид

$$\max \left(-\frac{1}{2} u' A D^{-1} A' u + u' (A D^{-1} d + b) - \frac{1}{2} d' D^{-1} d \right),$$

$$u \geq 0.$$

Эта двойственная задача используется в методе Лемке квадратичного программирования [19].

9. Доказать: если $s = \alpha s_1 + \beta s_2$, то для любой целевой функции $\varphi(u)$, удовлетворяющей условиям теоремы 7.5-1, имеет место

$$D\varphi(u; s) \geq \alpha D\varphi(u; s_1) + \beta D\varphi(u; s_2).$$

Построить пример, когда это условие выполняется как жесткое неравенство.

10. Рассмотреть задачу, для которой выполняются все условия теоремы 7.7-1, за исключением существования $x \in S$, такого, что $g(x) < 0$. Дать пример такой задачи, для которой теорема 7.7-1 несправедлива.

11. Рассмотреть прямую задачу вида

$$\min (2x_1^2 - 3x_1 + 4x_2^2 - 3x_2 + 3x_3^2 + 2x_3),$$

$$x_1 + x_2 + x_3 = 1,$$

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

Построить аналитическое выражение двойственной функции $h(u)$, минимизирующих функций $x_i(u)$ и первых производных $h(u)$. Построить эти функции графически и доказать, что утверждения § 7.7 сохраняют свою силу. Являются ли вторые производные $h(u)$ непрерывными всюду?

12. Установить условия для функций f_i , t_i и множеств S_i , гарантирующие, что декомпозиционная процедура § 7.8.1 может решить прямую задачу.

13. (Фалк [21]). Рассмотреть прямую задачу вида

$$\min \left\{ \sum_{i=1}^n f_i(x_i) \mid Ax = b, x \geq 0 \right\},$$

где A есть $m \times n$ -матрица, $n \geq m$, и x_i — скалярная переменная. Предположив, что каждая из f_i строго выпукла, т. е. $\partial^2 f_i / \partial x_i^2 > 0$, $\forall x_i$, (а) построить двойственную задачу, используя множители u ; (б) показать, что значения x_i , которые минимизируют $L(x, u)$ при $x \geq 0$, $x_i(u)$, единственны для данного u . Вывести явные формулы для $x_i(u)$; (в) показать, что область определения двойственной задачи, D , есть множество $\{u \mid A'u < L\}$, где $L = (L_1, \dots, L_n)$ и $L_i = \lim_{x_i \rightarrow \infty} \frac{df_i}{dx_i}$; (г) показать, что D разделимо на многогранные подобласти,

внутри которых двойственная функция $h(u)$ непрерывно дифференцируема, установить неравенства, определяющие эти подобласти; (д) показать, что $h(u)$ непрерывна на границах этих подобластей, но что вторые производные, вообще говоря, разрывны; (е) привести численный пример и иллюстрировать выводы (б) — (д).

ЛИТЕРАТУРА

1. H. W. Kuhn and A. W. Tucker, Nonlinear Programming, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1950, pp. 481—492.
2. J. E. Falk, Lagrange Multipliers and Nonlinear Programming, J. Math. Anal. Appl., 19, No. 1, 1967.

3. H. Everett, Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Res.*, **11**, 1963, pp. 399—417.
4. S. Karlin, *Mathematical Methods and Theory in Games, Programming and Economics*, vol. 1, Addison-Wesley Publishing Company, Inc., Reading, Mass., 1959. Русский перевод: С. Карлин, Математические методы в теории игр, программировании и экономике, «Мир», М., 1964.]
5. G. Hadley, *Nonlinear and Dynamic Programming*, Addison-Wesley Publishing Company, Inc., Reading, Mass., 1964. [Русский перевод: Дж. Хедли, Нелинейное и динамическое программирование, «Мир», М., 1967.]
6. O. L. Mangasarian and J. P. Ponnstein, Minimax and Duality in Nonlinear Programming, *J. Math. Anal. Appl.*, **11**, 1965, pp. 504—518.
7. J. Stoer, Duality in Nonlinear Programming and the Minimax Theorem. *Numerische Mathematik*, **5**, 1963, pp. 371—379.
8. R. T. Rockafellar, Duality and Stability in Extremum Problems Involving Convex Functions, *Pacific J. Math.*, **21**, 1967, pp. 167—187.
9. P. Wolfe, A Duality Theorem for Nonlinear Programming, *Quart. Appl. Math.*, **19**, 1961, pp. 239—244.
10. R. T. Rockafellar, *Nonlinear Programming*. American Mathematical Society Summer Seminar on the Mathematics of the Decision Sciences Stanford University, July—Aug. 1967.
11. D. G. Luenberger, *Convex Programming and Duality in Normed Space*. Proceedings of the IEES Systems Science and Cybernetics Conference, Boston, Oct. 11—13, 1967.
12. J. M. Danskin, The Theory of Max-Min with Applications, *J. SIAM Appl. Math.*, **14**, No. 4, 1966, pp. 641—665.
13. W. Fenchel, *Convex Cones, Sets, and Functions*. Mimeographed notes, Princeton University, 1963.
14. R. Fletcher and M. J. D. Powell, A Rapidly Convergent Descent Method for Minimization, *Computer. J.*, July 1963.
15. L. S. Lasdon and A. D. Waren, Mathematical Programming for Optimal Design, *Electro-Technology*, Nov. 1967, pp. 53—71.
16. J. B. Rosen, The Gradient Projection Method for Nonlinear Programming, Parts I and II. *J. Soc. Ind. Appl. Math.*, **8**, 1960, pp. 181—217; **9**, 1961, pp. 514—532.
17. R. Fletcher and C. M. Reeves, Function Minimization by Conjugate Gradients, *Computer. J.*, July 1964.
18. D. Goldfarb, Extension of Davidson's Variable Metric Method to Maximization under Linear Equality and Inequality Constraints, *J. SIAM Appl. Math.*, **17**, No. 4, 1969, pp. 739—764.
19. C. E. Lemke, A Method of Solution for Quadratic Programming, *Management Sci.*, **8**, No. 4, 1962.
20. L. Munini, Optimization of Coupled Systems by Decomposition. M. S. Thesis, Systems Research Center, Case Institute of Technology, 1966.
21. J. E. Falk, An Algorithm for Separable Convex Programming under Linear Equality Constraints, Research Analysis Corp. Tech. Paper RAC-TP-148, AD 623-092, 1965.
22. L. S. Lasdon, A Multi-Level Technique for Optimization. Ph. D. Thesis, Case Institute of Technology, 1964, System Research Center Rept. SRC 50-C-64-19.
23. L. S. Lasdon and J. D. Schoeffler, A Multi-Level Technique for Optimization. Proceedings of the 1965 Joint Automatic Control Conference, Troy, New York.
24. L. S. Lasdon and J. D. Schoeffler, Decentralized Plant Control. *ISA Trans.*, **5**, April 1966, pp. 175—183.
25. C. B. Brosilow and L. S. Lasdon, A Two level Optimization Technique for Recycle Proceedings of the A. I. Ch. E.—Ind. Chem. Eng. Joint Meeting,

- London, June 13—17, 1965, Symposium on the Application of Mathematical Modes in Chemical Engineering Research, Design; and Production.
26. L. S. Lasdon, Duality and Decomposition in Mathematical Programming. *IEEE Trans. Systems Sci. Cybernetics*, 4, No. 2, 1968, pp. 86—100.
 27. C. B. Brosilow and E. Nunez, Multi-Level Optimization Applied to a Catalytic Cracking Plant. *Can. J. Chem. Eng.*, 46, June 1968, pp. 205—212.
 28. A. V. Fiacco and G. P. McCormick, Sequential Unconstrained Minimization Techniques for Nonlinear Programming, John Wiley & Sons, Inc., New York, 1968.
 29. R. Fox and L. Schmit, Advances in the Integrated Approach to Structural Synthesis, *J. Spacecraft Rockets*, 3, No. 6, 1966, p. 858.
 30. A. M. Geoffrion, Primal Resource — Directive Approaches for Optimizing Nonlinear Decomposable Systems. Working Paper 141, The RAND Corporation, 1968.
 31. S. Kaplan, Solution of the Lorie-Sevage and Similar Integer Programming Problems by the Generalized Lagrange Multiplier Method. *Operations Res.*, 14, 1960, pp. 1130—1136.
 32. G. L. Nemhauser and Z. Ullmann, A Note on the Generalized Lagrange Multiplier Solution to an Integer Programming Problem. *Operations Res.*, 16, No. 2, 1968, pp. 450—453.

ГЛАВА VIII

ДЕКОМПОЗИЦИЯ ПУТЕМ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ (РАЗДЕЛЕНИЕ ОГРАНИЧЕНИЙ)

§ 8.1. ВВЕДЕНИЕ

В главе VII мы использовали двойственную программу для декомпозиции исходных проблем вида

$$\min f(x), \quad (1)$$

$$g(x) \leq 0, \quad g = (g_1, \dots, g_m), \quad (2)$$

$$x \in S \subseteq E^n. \quad (3)$$

Было показано (§ 7.4), что двойственная задача тесно связана с функцией

$$w(y) = \min \{f(x)/g(x) \leq y, x \in S\}. \quad (4)$$

В этой главе *) непосредственно решается прямая задача с использованием функции $w(y)$. Как и в двойственном подходе, декомпозиция дает результаты, когда f и все компоненты g являются аддитивно-сепарабельными, а S представляет собой прямое произведение множеств меньшей размерности. Однако здесь на каждом итеративном цикле находится допустимая точка исходной задачи, и обеспечивается монотонное убывание целевой функции. Это отличает данный подход от решения с помощью двойственной процедуры, где допустимость обеспечивается только в конце итеративной процедуры. В то время как двойственный метод имеет дело с ценами на ограниченные ресурсы, позволяя рынку определять размещение этих ресурсов, прямой подход реализует размещение результатов непосредственно.

§ 8.2. ФОРМУЛИРОВКА ПРОБЛЕМЫ

Удобно переписать ограничение (2) с вектором в правой части, равным b , который, конечно, может быть в частности равен нулю:

$$g(x) \leq b. \quad (5)$$

*) Эта глава основывается непосредственно на материалах [1], [2].

Как и в предшествующих главах, мы будем интерпретировать b как вектор доступных ресурсов, g — как количество ресурсов, потребляемых при уровне производства x , и f — как затраты, связанные с производством. Ограничения $x \in S$ выражают технологические ограничения для различных производственных способов. Далее мы вновь предположим, что задача (1) — (3) сепарабельна:

$$\min \sum_{i=1}^p f_i(x_i), \quad (6)$$

$$\sum_{i=1}^p g_i(x_i) \leq b, \quad (7)$$

$$x_i \in S_i, \quad i = 1, \dots, p. \quad (8)$$

Здесь x_i имеет n_i компонент, в то время как каждое g_i имеет m компонент. Функции f_i , g_i и множества S_i будут интерпретироваться, как связанные с i -й подсистемой большой системы. Эта прямая задача может рассматриваться и как задача оптимального распределения вектора ресурсов b по подсистемам. Мы попытаемся решать ее путем выбора допустимого распределения, проверки его на оптимальность и улучшения распределения, если таковое окажется не оптимальным.

Пусть количество ресурса, выделенного подсистеме i , равно y_i ; тогда для допустимости необходимо

$$\sum_{i=1}^p y_i \leq b. \quad (9)$$

Предполагая, что эти количества y_i подсистемы должны использовать наилучшим образом, мы приходим к следующим подзадам.

Задача $P_i(y_i)$:

$$\left. \begin{array}{l} \min f_i(x_i), \\ g_i(x_i) \leq y_i, \\ x_i \in S_i. \end{array} \right\} \text{Подпроблема } i. \quad (10)$$

Конечно, y_i должны быть выбраны так, чтобы эти подзадачи обязательно имели допустимые решения, так что необходимо учесть ограничения:

$$y_i \in V_i = \{y_i / \exists x_i \in S_i, \text{ такое, что } g_i(x_i) \leq y_i\}. \quad (11)$$

Минимальное значение целевой функции в подзадаче

$$w_i(y_i) = \min \{f_i(x_i) / g_i(x_i) \leq y_i, x_i \in S_i\}. \quad (12)$$

Это функция, того же самого вида, как и в (4), в дальнейшем будет фигурировать в качестве характеристики i -й подзадачи. Ясно, что для того, чтобы решить прямую задачу в целом, мы должны выбрать y_i , минимизирующее сумму таких функций при ограничениях (11) и (9). Иными словами, исходная задача (6) — (8) эквивалентна следующей координирующей задаче:

$$\min w(y) = \sum_{i=1}^p w_i(y_i), \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \quad (13)$$

$$\sum_{i=1}^p y_i \leq b, \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \quad (14)$$

$$y_i \in V_i, \quad i = 1, \dots, p. \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \quad (15)$$

Т. е., если $y^0 = (y_1^0, \dots, y_p^0)$ — решение координирующей задачи и x_i^0 — решение задачи $P_i(y_i^0)$, то $x^0 = (x_1^0, \dots, x_p^0)$ — решение исходной задачи. Обратно, если x^0 — решение исходной задачи, то $(g_1(x_1^0), \dots, g_p(x_p^0))$ — решение координирующей. Далее, координирующая задача недопустима, если и только если прямая задача не имеет допустимых решений, и не имеет ни одного оптимального решения, если и только если прямая не имеет ни одного оптимального решения.

Теперь сфокусируем внимание на координирующей задаче. В ней имеются простые линейные ограничения (14), нелинейные ограничения (15) и (вообще говоря, нелинейная) целевая функция. Конечно, за исключением, возможно, линейных и квадратичных задач, найти явное выражение целевой функции весьма трудно, если не невозможно, в случаях, когда y_i — вектор, содержащий несколько компонент (см. [2]). Таким образом, оценка w в некоторой точке y требует решения подзадачи $P_i(y_i)$ вида (10). Однако эти задачи имеют меньшую размерность, чем исходная, и могут быть решены независимо, что может оказаться выгодным, особенно, когда число связывающих ограничений m много меньше, чем общее число ограничений, определяющих все множества S_i . Далее, как будет показано, решение $P_i(y_i)$ дает также информацию, касающуюся производных w_i по направлению. Это облегчает решение координирующей задачи.

Эквивалентность между координирующей и исходной задачами никоим образом не зависит от выпуклости этих задач. Таким образом, если доступны средства для оценки w_i (и, возможно, ее производных), решение может быть получено любыми методами нелинейного программирования. Однако для дальнейшего продвижения в теории мы наложим следующие условия, которые и будут предполагаться выполняемыми в ходе дальнейшего изложения в этой главе.

Предположения о выпуклости и допустимости. Для $i = 1, 2, \dots, p$ имеют место условия:

1. S_i непусты, компактны и выпуклы.
2. $f_i(x_i)$ выпуклы и дифференцируемы на S_i .
3. Каждая компонента g_i выпукла и дифференцируема на S_i .
4. Прямая задача (6) — (8) имеет допустимое решение.

Предположения о выпуклости f_i , g_i и S_i обеспечивают, что V_i выпукло, и что w_i выпукла на V_i (см. задачу 5, гл. VII), так что координирующая программа является задачей выпуклого программирования. Допустимость исходной задачи приводит к тому, что координирующая задача также имеет допустимое решение. Из компактности S_i вместе с непрерывностью f_i и g_i следует, что подзадача i имеет оптимальное решение, поскольку она имеет допустимое. Дифференцируемость всех функций позволит нам сконструировать выражение для производных w_i по направлениям с использованием линейного программирования.

Для того чтобы обеспечить возможность прямого применения теории, излагаемой в Приложениях 1 и 2, мы распространим каждую функцию w_i на все E^m , считая, что

$$w_i(y_i) = +\infty, \quad y_i \notin V_i. \quad (16)$$

Таким образом, каждая w_i становится правильной выпуклой функцией с эффективной областью $\text{dom } w_i = V_i$. Мы также определим w условием

$$w(y) = +\infty, \quad \text{если } \sum_{i=1}^p y_i \not\leq b, \quad (17)$$

так что w также окажется выпуклой функцией с эффективной областью, совпадающей с допустимым множеством координирующей программы

$$\text{dom } w = F,$$

где

$$F = \left\{ y \mid \sum_{i=1}^p y_i \leq b, y_i \in V_i, i = 1, \dots, p \right\}.$$

Несмотря на все эти идеализирующие предположения, w_i может не быть дифференцируемой всюду на V_i , и разрывы производной могут оказаться в критических точках, например, в минимуме. Даже строгая выпуклость f_i не обеспечивает того, что w_i будут всюду дифференцируемы, и это — главный источник трудностей в решении координирующей задачи. Как показано в § 7.5, предположения 1—3 плюс условие строгой выпуклости целевой функции обеспечивают, что двойственная функция $h(u)$ дифференцируема для $u \geq 0$. Таким образом, в этом смысле пря-

мые функции w_i обладают не столь хорошим поведением, как двойственные h_i . С другой стороны, w_i строго выпуклы, если строго выпуклы f_i , и если выполнены предположения 1—3 в то время, как это не всегда справедливо для h_i .

Пример 1. Для иллюстрации рассмотрим задачу (Фалк [3])

$$w(y) = \min_{x_1, x_2 \geq 0} \left\{ \frac{1}{2} x_1^2 + x_1 + \frac{1}{2} x_2^2 - x_1 + x_2 = y \right\}.$$

Для $y \geq 0$, $x_1 = 0$, так что $x_2 = y$, в то время как для $y \leq 0$ $x_2 = 0$, что дает $x_1 = -y$. Таким образом,

$$w(y) = \begin{cases} \frac{y^2}{2}, & y \geq 0, \\ \frac{y^2}{2} - y, & y \leq 0. \end{cases}$$

Эта функция строго выпукла, но имеет разрыв непрерывности производной в точке минимума $y=0$, как показано на рис. 8.1.

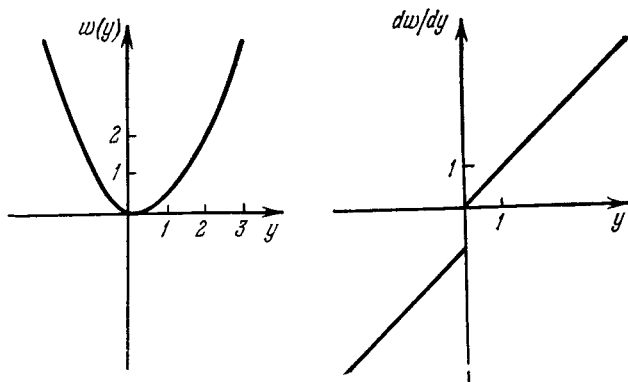


Рис. 8.1. Прямая функция и ее производная

Несмотря на эти трудности, так как каждая w_i выпукла, она имеет конечные производные по направлению во всех точках в $\text{int } V_i$ (теорема 16 Приложения 2). Это обосновывает алгоритм, предлагаемый в следующем параграфе, который использует производные по направлениям для получения улучшенного допустимого решения.

§ 8.3. АЛГОРИТМ ВОЗМОЖНЫХ НАПРАВЛЕНИЙ ДЛЯ КООРДИНИРУЮЩЕЙ ЗАДАЧИ

Рассматриваемый здесь алгоритм *) начинает решение координирующей программы с некоторого допустимого вектора y . Строится задача для определения направления, которая позволяет найти это направление s , обеспечивающее улучшение распределения y , либо покажет, что y является оптимальным. Если y не оптимальное, то может быть произведен шаг в направлении s , определяющий новое распределение, и процесс повторяется.

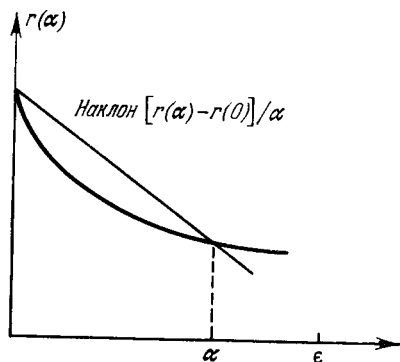


Рис. 8.2.

Предположим, что известна допустимая точка $y = (y_1, \dots, y_p) \in F$. Для того чтобы мы могли произвести малый шаг в некотором направлении s_i из y_i , и при этом $y_i + \alpha s_i \in V_i$, предположим следующее.

Предположение 1. Каждый вектор y_i принадлежит внутренности V_i .

Пусть $s = (s_1, \dots, s_p)$ есть предполагаемое направление поиска. Для того чтобы быть полезным, s должно обладать следующими свойствами [6]:

Определение 1. Направление s называется *возможным* в точке $y \in F$, если существует $\varepsilon > 0$ такое, что

$$y + \alpha s \in F, \quad 0 < \alpha \leq \varepsilon. \quad (1)$$

Определение 2. Возможное направление s называется *подходящим* для w в точке $y \in F$, если

$$w(y + \alpha s) < w(y) - \alpha \varepsilon, \quad 0 < \alpha \leq \varepsilon. \quad (2)$$

Заметим, что в обоих определениях использован один и тот же параметр ε . Грубо говоря, подходящее направление есть такое, что малое смещение по этому направлению улучшает целевую функцию без нарушения какого-нибудь ограничения. Соотношение (2) может быть интерпретировано геометрически при фиксации y и s , если ввести в рассмотрение

$$r(\alpha) = w(y + \alpha s). \quad (3)$$

*) Общая схема возможных направлений предложена Зойтендейком. По поводу алгоритма см. также [20] в списке дополнительной литературы (Прим. ред.).

Перепишем (2) в виде

$$\frac{r(\alpha) - r(0)}{\alpha} < -\varepsilon, \quad 0 < \alpha \leq \varepsilon. \quad (4)$$

Из рис. 8.2 видно, что это требование эквивалентно условию: наклон хорды, проходящей через точки $(\alpha, r(\alpha))$ и $(0, r(0))$, должен быть меньше $-\varepsilon$ при всех α в указанной области.

Задача выбора направления. Для того чтобы s было возможным направлением, необходимо, чтобы в некоторой области положительных значений α были выполнены условия

$$y_i + \alpha s_i \in V_i, \quad i = 1, \dots, p \quad (5)$$

и

$$\sum_i (y_i + \alpha s_i) \leq b \quad (6)$$

или

$$\alpha \sum_i s_i \leq b - \sum_i y_i. \quad (7)$$

Поскольку $y_i \in \text{int } V_i$, имеется область положительных значений α , таких, что (5) удовлетворяется для любого направления s_i . Если k -я компонента вектора $b - \sum_i y_i$ положительна, то су-

ществует область положительных α , таких, что k -е неравенство удовлетворяется при любом выборе векторов $\{s_i\}$, так что соответствующие ограничения могут не приниматься во внимание. Если же k -я компонента равна нулю, то сдвиг по некоторым направлениям немедленно нарушает это ограничение. Пусть

$$B = \left\{ j / b_j - \sum_i y_{ij} = 0 \right\} \quad (8)$$

есть индексное множество ограничений, связывающих в y . Тогда для возможности направления необходимо выполнение условий

$$\sum_i s_{ij} \leq 0, \quad j \in B. \quad (9)$$

Интуитивно ясно, что для того, чтобы найти локально лучшее подходящее направление, целесообразно разыскивать такое возможное направление, которое минимизирует линейное приращение целевой функции, иначе говоря, целесообразно решить следующую проблему выбора направления:

Задача (DF).

$$\min D\varphi(y; s) = \sum_{i=1}^p D\varphi_i(y_i, s_i) \quad (10)$$

при ограничениях

$$\sum_{i=1}^p s_{ij} \leq 0, \quad j \in B \quad (11)$$

и ограничениях нормализации

$$\|s\| \leq 1, \quad (12)$$

где $\|s\|$ есть норма s .

Ограничения (11) однородны и, как показано в теореме 9 Приложения 2, $Dw_i(y_i; s_i)$ положительно однородно по s_i , т. е.

$$Dw_i(y_i; \lambda s_i) = \lambda Dw_i(y_i; s_i), \quad \lambda \geq 0. \quad (13)$$

Таким образом, если опущено требование нормализации и существует s , для которого целевая функция (10) отрицательна, то всегда можно сделать (10) стремящимся к $-\infty$ путем умножения s на произвольно большую положительную константу. Добавление ограничений (12) обеспечивает конечность оптимального решения проблемы (DF). Может показаться, что получение явного выражения для $Dw_i(y_i; s_i)$ является столь же трудным, как и нахождение такого выражения для самого w_i . Однако теория, изложенная в Приложениях 1 и 2, дает нам возможность решить эту задачу и записать (DF) в виде линейной программы.

Прежде всего, однако, мы установим критерий оптимальности y и покажем, что (DF) всегда дает подходящее возможное направление, если таковое существует.

Теорема 1. Пусть y является допустимым для координирующей программы. Если $s=0$ — оптимальное решение задачи выбора направления (DF), то y является решением координирующей программы.

Доказательство. Поскольку $Dw(y; 0) = 0$, то должно иметь место

$$Dw(y; s) \geq 0 \quad (14)$$

для всех s , удовлетворяющих условиям возможности (11). Для всех других векторов s в силу (8.2—17) имеет место $w(y + \alpha s) = +\infty$ для всех $\alpha > 0$, так что $Dw(y; s) = +\infty$. Таким образом, (14) справедливо для всех $s \in E^n$. Тогда в силу теоремы 11 Приложения 2 y является решением координирующей программы.

Если критерий оптимальности, выраженный предшествующей теоремой, не проходит, то найдется такое направление s , для которого $Dw(y; s)$ отрицательно. Покажем теперь, что это направление и является подходящим в смысле определения 2.

Теорема 2. Пусть s — решение задачи (DF) для некоторого $y \in F$. Если $Dw(y; s)$ — отрицательно, то s является подходящим направлением для w в y .

Доказательство. Вектор s является возможным по построению. Следовательно, существует α такое, что

$$y + \alpha s \in F, \quad 0 < \alpha \leq \bar{\alpha}. \quad (15)$$

Для того чтобы доказать, что s является подходящим, используем тот факт, что

$$Dw(y, s) = \lim_{\lambda \rightarrow 0+} (w(y + \lambda s) - w(y))/\lambda < 0. \quad (16)$$

По теореме 8 Приложения 2 разностное отношение

$$d(\lambda) = (w(y + \lambda s) - w(y))/\lambda$$

является неубывающим. Совместно с (16) это утверждение приводит к существованию некоторого $\varepsilon > 0$ и $\bar{\lambda} > 0$, таких, что

$$d(\lambda) < -\varepsilon, \quad 0 < \lambda \leq \bar{\lambda}. \quad (17)$$

Обозначим $\delta = \min(\varepsilon, \bar{\lambda})$. Тогда можно показать, что

$$d(\lambda) < -\delta, \quad 0 < \lambda \leq \delta. \quad (18)$$

Случай 1. $\varepsilon > \bar{\lambda}$. Здесь в силу (17) имеет место

$$d(\lambda) < -\varepsilon < -\bar{\lambda} = -\delta, \quad 0 < \lambda \leq \delta.$$

Случай 2. $\varepsilon \leq \bar{\lambda}$. Тогда в силу (17)

$$d(\lambda) < -\varepsilon = -\delta.$$

Так как последнее справедливо для всех $0 < \lambda \leq \bar{\lambda}$, оно остается справедливым для всех $0 < \lambda \leq \delta$. Положим $\lambda^* = \min(\bar{\alpha}, \delta)$. Тогда в силу (15)

$$y + \alpha s \in F, \quad 0 < \alpha \leq \lambda^*, \quad (19)$$

и в силу (18)

$$d(\lambda) < -\lambda^*, \quad 0 < \lambda \leq \lambda^*.$$

Последнее может быть переписано в виде

$$w(y + \lambda s) < w(y) - \lambda \lambda^*, \quad 0 < \lambda \leq \lambda^*; \quad (20)$$

Следовательно, теорема доказана, поскольку (19) и (20) представляют собой определяющие неравенства (1) и (2) для подходящего направления s в точке y .

Возвращаясь теперь к вопросу о получении явного выражения $Dw(y; s)$, мы прежде всего охарактеризуем субградиенты функций w_i . Геометрические идеи главы VII подсказывают этот результат немедленно. Субградиент w_i в точке y_i есть вектор коэффициентов опорной гиперплоскости к графику w_i в y_i . Из § 7.4 ясно, что если u_i^0 является решением задачи, двойственной $P_i(y_i)$, то $-u_i^0$ и есть вектор коэффициентов такой гипер-

плоскости. Тогда, если x_i^0 — решение $P_i(y_i)$, то пара (x_i^0, u_i^0) дает седловую точку функции Лагранжа задачи $P_i(y_i)$. Таким образом, субградиенты w_i в точке y_i совпадают с точностью до знака с множителями Лагранжа задачи $P_i(y_i)$. Далее, любое оптимальное решение x_i^0 достаточно для того, чтобы генерировать все субградиенты, как мы теперь и покажем. В доказательстве опустим индекс i , поскольку он не играет существенной роли.

Теорема 3. Пусть $L(x, u) = f(x) + u(g(x) - y)$ есть лагранжева функция для задачи $P(y)$, и пусть x^0 решение этой задачи. Тогда $-u^0$ есть субградиент w в точке y (формально это записывается в виде: $-u^0 \in \partial w(y)$), если и только если (x^0, u^0) является седловой точкой функции Лагранжа, т. е.

$$L(x^0, u) \leq L(x^0, u^0) \leq L(x, u^0).$$

Доказательство. Необходимость. В силу свойства седловой точки, имеем

$$f(x^0) + u^0(g(x^0) - y) \leq (f(x) + u^0(g(x) - y)), \quad \forall x \in S,$$

или, поскольку

$$f(x^0) = w(y), \quad u^0(g(x^0) - y) = 0,$$

то

$$w(y) \leq f(x) + u^0(g(x) - y). \quad (21)$$

Выберем $z \in E^m$. Если существует $x \in S$, такое, что $g(x) \leq z$, то для такого x мы можем заменить в (21) $g(x)$ на z , что дает

$$w(y) \leq f(x) + u^0(z - y).$$

Вычисляем инфимум правой части этого неравенства по всем $x \in S$, таким, что $g(x) \leq z$. Тогда

$$w(y) \leq w(z) + u^0(z - y). \quad (22)$$

Если не существует $x \in S$, таких, что $g(x) \leq z$, то $w(z) = +\infty$, так что записанное неравенство справедливо для всех $z \in E^m$. Это и есть соотношение, утверждающее, что $-u^0$ является элементом $\partial w(y)$.

Достаточность. Предположим, что (22) справедливо для всех $z \in E^m$. Выберем

$$z = g(x^0), \quad (23)$$

что приводит к

$$w(z) = f(x^0) = w(y). \quad (24)$$

Подставляя (23) и (24) в (22), получаем

$$u^0(g(x^0) - y) \geq 0,$$

но $u^0 \geq 0$ и $g(x^0) - y \leq 0$, так что

$$u^0(g(x^0) - y) = 0. \quad (25)$$

Это условие вместе с соотношением $g(x^0) \leq y$ и дает два из трех условий для седловой точки, приведенных в теореме 7.3—1. Для того чтобы показать справедливость третьего, используем равенство $f(x^0) = w(y)$. Подставив его и (25) в (22), получим

$$f(x^0) + u^0(g(x^0) - y) \leq w(z) + u^0(z - y), \quad \forall z \in E^m. \quad (26)$$

Выберем x произвольно из S и положим

$$z = g(x). \quad (27)$$

Поскольку x является допустимой точкой для $P(g(x))$, то

$$w(z) \leq f(x). \quad (28)$$

Используя (27) и (28) в (26), получим

$$f(x^0) + u^0(g(x^0) - y) \leq f(x) + u^0(g(x) - y). \quad (29)$$

Так как x является произвольным элементом S , то (29) и служит третьим недостающим условием. Теорема доказана.

В силу теоремы 17 Приложения 2, выпуклая функция дифференцируема в точке, если и только если она имеет в этой точке единственный субградиент. Так как при x^0 , оптимальном для $P(y)$, пара (x^0, u^0) есть седловая точка L , если и только если u^0 дает решение двойственной к задаче $P(y)$, то из теоремы 3 вытекает следующее утверждение.

Теорема 4. *Функция w , дифференцируемая в y , если и только если задача, двойственная к задаче $P(y)$, имеет единственное оптимальное решение.*

В примере 1 § 8.2 показано, что w недифференцируема при $y=0$. Задача, двойственная к $P(0)$, состоит в максимизации

$$h(u) = \min_{x_1, x_2 \geq 0} \left(\frac{x_1^2}{2} + x_1 + \frac{x_2^2}{2} + u(-x_1 + x_2) \right)$$

для всех u . Легко показать, что

$$h(u) = \begin{cases} -\frac{u^2}{2}, & u \leq 0, \\ 0, & 0 \leq u \leq 1, \\ -\frac{(u-1)^2}{2}, & u \geq 1. \end{cases}$$

Как ясно из рис. 8.3, все значения u в интервале $[0, 1]$ дают решение этой двойственной задачи. Из рис. 9.1 мы видим, что w недифференцируема в $y=0$, и что субградиенты в точке 0

совпадают с отрицательными значениями оптимальных множителей.

Как было показано Джоффрионом [2], теорема 3 может быть использована для получения явного выражения $D\omega_i(y_i; s_i)$ в точках $y_i \in \text{Int dom } \omega_i$, т. е. $y_i \in \text{int } V_i$. Тогда, как показано в Приложении 2, $\partial\omega_i(y_i)$ непусто, компактно и выпукло, и, в силу теоремы 16 Приложения 2 имеет место

$$D\omega_i(y_i; s_i) = \max_{x^* \in \partial\omega_i(y_i)} (x^* s_i) = \max_{u_i \in U_i} (-u_i s_i), \quad (30)$$

где U_i есть множество всех оптимальных двойственных переменных для задачи $P_i(y_i)$. Отметим, что, по определению V_i , условие

$y_i \in \text{int } V_i$ эквивалентно существованию $x_i \in S_i$, такого, что $g_i(x_i) < y_i$. Это и есть данное Карлином условие в теореме 7.3—6, гарантирующее существование оптимального вектора двойственных переменных для $P_i(y_i)$.

Пусть каждое множество S_i определено с помощью вектор-функции c_i :

$$S_i = \{x_i / c_i(x_i) \leq 0\}. \quad (31)$$

Предположим, что каждая компонента c_i дифференцируема и выпукла. Тогда, если x_i^0 есть любое оптимальное решение $P_i(y_i)$, то $u_i \in v_i$, если и только если x_i^0 , u_i и некоторое λ_i удовлетворяют условиям Куна — Таккера для $P_i(y_i)$.

Образует лагранжиан

$$\tilde{L}_i = f_i + u_i(g_i - y_i) + \lambda_i c_i.$$

В его седловой точке выполняются условия:

$$I'_{g_i} u_i + I'_{c_i} \lambda_i = -\nabla f_i, \quad (32)$$

$$u_i(g_i - y_i) = 0, \quad (33)$$

$$\lambda_i c_i = 0, \quad (34)$$

$$u_i \geq 0, \quad \lambda_i \geq 0. \quad (35)$$

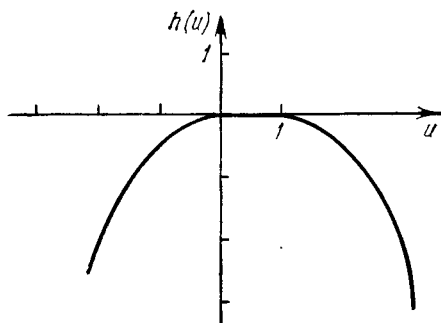


Рис. 8.3. Двойственная целевая функция в примере 1.

Здесь I_{g_i} есть якобиан для g_i , т. е. матрица, строками которой являются градиенты функций, составляющих g_i . Сходным образом определено I_{c_i} . Все функции x_i вычислены при x_i^0 . В силу (30) $Dw_i(y_i; s_i)$ есть оптимальное значение целевой функции линейной задачи

$$\max \{-u_i s_i / (u_i, \lambda_i) \text{ удовлетворяет (32) — (35)}\} \quad (36)$$

Некоторые переменные в этой задаче могут быть опущены, если использовать условия дополняющей нежесткости (33) — (34). Поскольку $g_i - y_i$ и c_i оба являются неположительными векторами, эти условия эквивалентны следующему:

$$u_{ij} = 0, \text{ если } g_{ij} - y_{ij} < 0, \quad (37)$$

$$\lambda_{ij} = 0, \text{ если } c_{ij} < 0. \quad (38)$$

Таким образом, (36) можно переписать в виде

$$\max \left\{ - \sum_j u_{ij} s_{ij} \right\} \quad (39)$$

при ограничениях

$$\sum_j \nabla g_{ij} u_{ij} + \sum_j \nabla c_{ij} \lambda_{ij} = - \nabla f_i, \quad (40)$$

$$u_{ij} \geq 0, \lambda_{ij} \geq 0, \quad (41)$$

где суммирование ведется по таким j , что $g_{ij} - y_{ij} = 0$ и $c_{ij} = 0$.

Так как мы хотим найти наилучшее s_i , то записанная выше программа имеет не идеальную форму: целевая функция включает произведение переменных. Переход к двойственной задаче позволяет избежать этой трудности. Двойственная задача (при s_i временно зафиксированных) такова:

$$\min \{- \nabla f'_i z_i\}, \quad (42)$$

$$\nabla g'_{ij} z_i \geq - s_{ij}, \quad (43)$$

$$\nabla c'_{ij} z_i \geq 0. \quad (44)$$

Поскольку $Dw_i(y_i; s_i)$ — конечно, прямая задача (39) — (41) имеет конечное оптимальное решение. В силу теорем двойственности двойственная к ней задача также имеет конечное оптимальное решение, и оптимальное значение ее целевой функции, равное $Dw_i(y_i; s_i)$. Таким образом, для фиксированного вектора $s = (s_1, \dots, s_p)$ производная $w(y)$ по направлению дается формулой

$$Dw(y; s) = \sum_{i=1}^p \min \{- \nabla f'_i z_i / (z_i, s_i) \text{ удовлетворяет (43) — (44)}\}. \quad (45)$$

В задаче выбора направления нам требовалось найти вектор s , который минимизирует выражение, представленное выше,

при ограничениях на возможность и требованиях нормализации. Минимизация суммы минимумов (45) на подмножестве векторов s_i эквивалентна минимизации одновременно по s_i и z_i , т. е. мы должны решать задачу

$$\min \left\{ - \sum_{i=1}^p \nabla f'_i z_i \right\}, \quad (46)$$

$$\sum_{i=1}^p s_{ij} \leq 0, \quad j \text{ такое, что } \sum_{i=1}^p y_{ij} = b_j, \quad (47)$$

и для $i = 1, \dots, p$,

$$\nabla g'_{ij} z_i \geq -s_{ij}, \quad j \text{ такое, что } g_i(x_i^0) - y_{ij} = 0, \quad (49)$$

$$\nabla c'_{ij} z_i \geq 0, \quad j \text{ такое, что } c_{ij}(x_i^0) = 0. \quad (50)$$

Построенная задача будет линейной, если мы используем норму вида

$$\|s\| = \|s\|_\infty = \max_j |(s)_j|. \quad (51)$$

Тогда требование нормализации (48) записывается так:

$$-1 \leq s_{ij} \leq 1. \quad (52)$$

Сделаем два замечания по поводу этой задачи.

1. Во-первых, существенны только s_{ij} с индексами, указанными в условиях (49). Только они определяют изменение распределения ресурсов y_i на связующих ограничениях. Другие s_{ij} ассоциированы с неактивными ограничениями, и изменения в правых частях этих ограничений локально произвольны. Таким образом, s_{ij} , появляющиеся в (47), но не появляющиеся в (49), должны устанавливаться на их нижнем пределе, для того, чтобы сделать (47) как можно свободнее, в то время как s_{ij} , не входящие ни в (47), ни в (49), являются полностью произвольными. Во-вторых, задача имеет угловую структуру, причем (47) дает связующие ограничения, а (49) — (50) — угловые блоки. Поэтому такая задача может быть эффективно решена применением методов, изложенных в главах II, IV или V. Ограничения (52) могут быть преобразованы к ограничениям неотрицательности или к ограничениям сверху с помощью подстановки

$$\bar{s}_{ij} = s_{ij} + 1. \quad (53)$$

Процедура из § 5.3 позволяет учесть ограничения сверху без увеличения размера базиса.

Дальнейшие упрощения возникают в точках, где одна или более функций w_i являются дифференцируемыми. Это имеет место в точках y_i , если и только если $\partial w_i(y_i)$ содержит один-

единственный элемент, т. е. если и только если компонента u_i допустимого решения линейной системы (40) — (41) является единственной. Если эта компонента есть u_i^0 , то тогда производные по направлению даются формулами — $u_i^0 s_i$, и ограничения (49) — (50) могут быть опущены.

Когда все функции w_i дифференцируемы, задача выбора направления сводится к следующей:

$$\min \left\{ - \sum_{i=1}^p u_i^0 s_i \right\}, \quad (54)$$

$$\sum_{i=1}^p s_{ij} \leq 0, \quad j \in B, \quad (55)$$

$$-1 \leq s_{ij} \leq 1. \quad (56)$$

Здесь имеется не более m ограничений плюс ограничения на переменные сверху. Практически можно ожидать, что такая ситуация будет возникать часто, поскольку каждая из w дифференцируема почти всюду в $\text{int } V_i$. В любом случае маловероятно, что все функции w_i в данной точке y окажутся недифференцируемыми. С функциями, которые являются дифференцируемыми, можно обойтись так, как это было указано выше; другие же потребуют введения в множество ограничений типа (49) — (50) и слагаемого — $\nabla f_i' z_i$ в целевую функцию.

Выбор длины шага. Если найдено подходящее направление s , то новая точка y находится путем выбора длины шага $\alpha > 0$ в выражении

$$y^n = y + \alpha s. \quad (57)$$

Существует много схем выбора α [6]. Многие из них сводятся к решению выпуклой одномерной задачи

$$\min \{w(y + \alpha s) / y + \alpha s \in F\}. \quad (58)$$

Вообще говоря, можно принять схему движения малыми шагами по s от точки y до тех пор, пока либо будет достигнут минимум $w(y + \alpha s)$, или точка $y + \alpha s$ выйдет из F . Любые процедуры одномерной минимизации, разработанные для случая отсутствия ограничений, могут быть приспособлены и здесь, если учесть, что ограничения $y + \alpha s \in F$ в явной форме имеют вид

$$\sum_{i=1}^p (y_i + \alpha s_i) \leq b, \quad (59)$$

$$y_i + \alpha s_i \in V_i. \quad (60)$$

Ограничения (49) эквивалентны легко вычислимой верхней границе для α . Нарушение же ограничений (60) может быть

обнаружено по отсутствию допустимых решений в некоторой подзадаче.

Конечно, наша формулировка задачи выбора направления основывалась на условии $y_i \in \text{int } V_i$. Это условие может быть гарантировано путем выбора шага несколько меньшим, чем оптимальный, в смысле (58), если обнаружится, что y_i^n окажется на границе V_i . Если обеспечено, что такие возмущения решения будут стремиться к нулю по мере развития итерационной процедуры, то граничные точки могут достигаться как предельные значения.

Если получено удовлетворительное решение (58), то в точке $y = y^n$ вновь решается задача выбора направления, и итеративный процесс повторяется. Критерий остановки может быть основан на теореме 1, т. е. производится остановка, когда оптимальное значение целевой функции задачи (DF) становится больше, чем $-\varepsilon$ для некоторого числа циклов, или же можно основываться на поведении последовательности значений $w(y)$, которая является монотонно убывающей.

§ 8.4. АЛЬТЕРНАТИВНЫЙ ПОДХОД К РЕШЕНИЮ ЗАДАЧИ ВЫБОРА НАПРАВЛЕНИЯ

В работе [1] Сильвермен предложил другую формулировку задачи выбора направления, которая развивает идеи, высказанные в [4]; хотя она представляется более сложной, чем формулировка в предшествующем параграфе, но приводит к задаче выбора направления меньшей размерности. Интересным представляется также способ, в котором используются субградиенты и производные по направлению.

Построение начинается с характеристики множеств оптимальных двойственных решений в задачах $P_i(y_i)$, обозначаемых в дальнейшем U_i . Здесь и в дальнейшем мы будем интересоваться только одной из подпроблем, поэтому индекс i может быть опущен. Таким образом, $P_i(y_i) \equiv P(y)$, $U_i \equiv U$ и т. д.

Теорема 1. Пусть x^0 — решение $P(y)$. Тогда U совпадает с множеством u -компонент оптимального решения следующей линейной задачи относительно переменных (u, λ) :

$$\max \{u(g - y) + \lambda c\}, \quad (1)$$

$$I'_g u + J'_c \lambda = -\nabla f, \quad (2)$$

$$u \geq 0, \quad \lambda \geq 0, \quad (3)$$

где J_g и J_c те же, что и в (8.3—32), а все функции x вычислены при x^0 .

Доказательство. По определению, $u \in U$, если и только если x^0 , u и некоторое λ удовлетворяют условиям Куна — Так-

кера (8.3—32)—(8.3—35). Ограничения задачи (1)—(3) и представляют собой эти условия, но без условий дополняющей нежесткости. Однако, поскольку $g - y$ и c неположительны, требование выполнения условий дополняющей нежесткости эквивалентно нахождению переменных, обращающих целевую функцию (1) в нуль и, тем самым, доставляющих ей максимум.

Как было показано в § 8.3, все оптимальные решения задачи (1)—(3) будут такими, что $u_j = 0$, если $g_j - y_j < 0$, и $\lambda_j = 0$, если $c_j < 0$. Таким образом, мы можем работать в пространстве уменьшенной размерности, опуская все нулевые компоненты u и λ . Если обозначить

$$B_i = \{j/g_{ij}(x_i^0) - y_{ij} = 0\} \quad (4)$$

и мы имеем дело с подзадачей i , то u имеет одну компоненту для каждого индекса в B_i .

Согласно определению, U есть множество всех выпуклых комбинаций u -компонент оптимальных крайних точек в (1)—(3)—компактный выпуклый многогранник. Пусть $E(U)$ есть множество его крайних точек с элементами u_t , $t=1, 2, \dots$. Для любого вектора s максимум в $Dw(y; s) = \max_{u \in U} (-us)$ достигается

на одной или большем числе крайних точек. Обозначим

$$C_i = \{s/\max_{u \in U} (-us) = -u_i s\} \quad (5)$$

множество *) векторов s , для которых максимум достигается в u_i . Для того чтобы s принадлежало C_i , необходимо и достаточно, чтобы

$$-u_i s \geq -u_j s, \quad \forall u_j \in E(U) \quad (6)$$

или

$$(u_i - u_j)s \leq 0, \quad j \neq i. \quad (7)$$

Введем в рассмотрение матрицу W_i вида

$$W_i = \begin{bmatrix} u_i - u_1 \\ u_i - u_2 \\ \vdots \\ u_i - u_T \end{bmatrix}. \quad (8)$$

Тогда (7) может быть записано в виде

$$W_i s \leq 0, \quad (9)$$

и C_i определяется следующим образом:

$$C_i = \{s/W_i s \leq 0\}. \quad (10)$$

*) Отметим, что если мы рассматриваем подзадачу i , то u_i и s имеют такое число компонент, как и u , которое совпадает с числом элементов в B_i .

Иначе говоря, C_i представляет собой пересечение конечно-го числа полупространств, содержащих начало координат, т. е. C_i является выпуклым многогранным конусом, а следовательно, имеет конечное число направляющих векторов. Пусть v_{ri} есть r -й направляющий вектор C_i . Тогда любое $s \in C_i$ может быть представлено в виде

$$s = \sum_r \beta_{ri} v_{ri}, \beta_{ri} \geq 0. \quad (11)$$

Далее, поскольку для любого $s \in C_i$ имеет место

$$D\omega(y; s) = -u_i s, \quad (12)$$

то, используя (11), получаем

$$D\omega(y; s) = \sum_r \beta_{ri} (-u_i v_{ri}), \quad s \in C_i, \quad (13)$$

$$D\omega(y; s) = \sum_r \beta_{ri} D\omega(y; v_{ri}). \quad (14)$$

Таким образом, производные ω по направлению в точке y линейны на каждом конусе C_i . Так как для любого $s \in E^m$ максимум (5) достигается в некоторой крайней точке u_i , то объединение всех конусов C_i есть E^m . Таким образом, для любого $s \in E^m$ существуют $\beta_{ri} \geq 0$, такие, что

$$s = \sum_{r,i} \beta_{ri} v_{ri}. \quad (15)$$

$$D\omega(y; s) = \sum_{r,i} \beta_{ri} D\omega(y; v_{ri}). \quad (16)$$

Векторы v_{ri} , обладающие этими свойствами, названы в [1], [4] *базисными планами*. Конечно, ситуация сильно упрощается, когда U состоит только из единственного элемента u , т. е. когда ω дифференцируема в y . Тогда

$$D\omega(y; s) = -us, \quad (17)$$

и базисные планы являются столбцами матрицы $[I; -I]$, где I — единичная матрица.

Для последующего изложения необходимо восстановить индекс подзадачи i . Для простоты мы обозначим через r_{ij} j -й базисный план и через C_i^i i -й конус вида (10), ассоциированный с подзадачей i . Вектор $*r_{ij}$, конечно, представляет собой направляющий вектор одного из конусов C_i^i . При таких обозначениях

*) Заметим, что вектор r_{ij} имеет столько компонент, сколько индексов в B_i .

соотношения (15) и (16) приобретают вид

$$s_i = \sum_j \beta_{ij} r_{ij}, \quad (18)$$

$$Dw_i(y_i; s_i) = \sum_j c_{ij} \beta_{ij}, \quad (19)$$

где

$$c_{ij} = Dw_i(y_i; r_{ij}). \quad (20)$$

Суммируем теперь процедуру получения векторов r_{ij} .

1. Начать с вектора $y = (y_1, \dots, y_p)$ такого, что $y_i \in \text{int } V_i$ и $\sum_i y_i \leq b$. Для $i = 1, \dots, p$ делать следующее:

2. Решить $P_i(y_i)$, получая оптимальное решение x_i^0 .

3. Решить линейную задачу (1) — (3), оценивая все функции x_i в точке x_i^0 и получая все оптимальные крайние точки $\{u_j^i, \lambda_j^i\}$. Пусть $E(U_i)$ есть множество всех элементов u_j^i .

а) если для некоторого i $E(U_i)$ содержит только один элемент u_i^0 , то производная по направлению дается формулой $Dw_i(y_i; s_i) = -u_i^0 s_i$.

б) для i такого, что $E(U_i)$ содержит более, чем один элемент, образовать матрицы W_i^j в (8) и найти направляющие вектора конусов C_i^j . (Алгоритмы для этой цели описаны в [4].) Базисные планы $\{r_{ij}\}$ представляют собой объединение всех этих множеств направляющих векторов.

К сожалению, сумма всех линейных выражений вида (19) не может быть использована в качестве целевой функции задачи выбора направления без введения дополнительных ограничений. Трудность заключается в том, что в то время, как существуют $\beta_{ij} \geq 0$, такие, что (19) справедливо для любых s_i , правая часть (19) не обязательно равна $Dw_i(y_i; s_i)$ (при s_i , задаваемых (18)) для произвольных неотрицательных β_{ij} . Для того чтобы было справедливо (18) и (19), все r_{ij} , которые имеют положительные коэффициенты, должны быть в одном и том же конусе, т. е. β_{ij} должны удовлетворять следующему условию.

Ограничение на допустимую интерполяцию. Множество неотрицательных чисел $\{\beta_{ij}\}$ удовлетворяет условиям допустимости интерполяции (сокращенно УДИ), если из неравенств $\beta_{it} > 0$ и $\beta_{is} > 0$ следует, что r_{it} и r_{is} являются элементами одного и того же конуса, скажем, C_q^i .

Если УДИ выдержано для каждого i , то мы можем использовать (18) и (19) в проблеме выбора направления, сформулированной в п. 8.3-10 — п. 8.3-12, для получения следующей

формулировки:

$$\min \sum_{i,j} c_{ij} \beta_{ij}, \quad (21)$$

$$\left(\sum_{i,j} \beta_{ij} r_{ij} \right)_k \leq 0, \quad k \in B, \quad (22)$$

$$\beta_{ij} \geq 0, \quad \forall i, j,$$

$$\|s\| \leq 1, \quad (23)$$

$$\text{УДИ для каждого } i. \quad (24)$$

Ограничения (24), выражающие УДИ, нелинейны. К счастью, теперь можно показать, что, если УДИ опущено, а задача (21)–(23) разрешима, то любое оптимальное решение удовлетворяет УДИ.

Теорема 2. Любое оптимальное решение задачи выбора направления (21)–(23) удовлетворяет УДИ, т. е. (24)

Доказательство. Предположим, что задача (21)–(23) имеет оптимальное решение β^0 , которое не удовлетворяет УДИ, т. е. существует индекс i и числа β_{is}^0 и β_{it}^0 , такие, что r_{is} и r_{it} лежат не в одном и том же конусе. В силу (20) имеем

$$c_{ij} \beta_{ij}^0 = D\omega_i(y_i; \beta_{ij}^0 r_{ij}) = \max_{u \in U_i} (-u \beta_{ij}^0 r_{ij}), \quad \forall j. \quad (25)$$

Пусть

$$s_i^0 = \sum_j \beta_{ij}^0 r_{ij} \in C_q^i. \quad (26)$$

Тогда

$$D\omega_i(y_i; s_i^0) = \max_{u \in U_i} (-u s_i^0) = -u_q s_i^0 = -\sum_j \beta_{ij}^0 (u_q, r_{ij}). \quad (27)$$

По определению c_{ij} ,

$$c_{ij} \beta_{ij}^0 = \beta_{ij}^0 \max_{u \in U_i} (-u r_{ij}) \geq -\beta_{ij}^0 u_q r_{ij}, \quad \forall j. \quad (28)$$

В силу гипотезы, r_{is} и r_{it} не лежат в C_q^i . Предположим, что $r_{is} \notin C_q^i$, откуда следует, что $\beta_{is}^0 r_{is} \notin C_q^i$. Тогда при $j = s$ максимум (28) не достигается на u_q , так что

$$c_{is} \beta_{is}^0 > -\beta_{is}^0 (u_q, r_{is}). \quad (29)$$

В силу (28) и (29) имеем

$$\sum_j c_{ij} \beta_{ij}^0 > -\sum_j \beta_{ij}^0 (u_q, r_{ij}) = D\omega_i(y_i; s_i^0). \quad (30)$$

Запишем теперь s_i^0 в виде линейной комбинации направляющих векторов C_q^i :

$$s_i^0 = \sum_{r_{ij} \in C_q^i} \hat{\beta}_{ij} r_{ij}, \quad \hat{\beta}_{ij} \geq 0. \quad (31)$$

Тогда

$$Dx_i(y_i; s_i^0) = -u_q s_i^0 = \sum_{r_{ij} \in C_q^i} c_{ij} \hat{\beta}_{ij}. \quad (32)$$

Комбинируя (30) и (32), получаем

$$\sum_i c_{ij} \beta_{ij}^0 > \sum_{r_{ij} \in C_q^i} c_{ij} \hat{\beta}_{ij}. \quad (33)$$

Пусть $\hat{\beta}_i$ — вектор, компонентами которого являются $\hat{\beta}_{ij}$ для $r_{ij} \in C_q^i$ и нули — в противном случае. Пусть

$$\hat{\beta} = (\beta_1^0, \beta_2^0, \dots, \hat{\beta}_i, \dots, \beta_p^0), \quad (34)$$

$$\beta^0 = (\beta_1^0, \beta_2^0, \dots, \beta_i^0, \dots, \beta_p^0).$$

Поскольку s_i^0 задано как (26), так и (31), то $\hat{\beta}$ удовлетворяет условиям (22) — (23). В силу (33) $\hat{\beta}$ доставляет меньшее значение целевой функции, чем β^0 , что противоречит оптимальности β^0 . Таким образом, теорема доказана.

Заметим, что хотя в доказательстве явно и не использовались никакие предположения о выпуклости, однако доказательство связано с ними, поскольку использовались условия (25) и (27) — (28) для производной по направлению, справедливость которых сама вытекает из предположений о выпуклости.

Как и в § 8.3, задача выбора направления (21) — (23) становится задачей линейного программирования, если используется соответствующая норма. Тогда (23) принимает вид

$$-e_m \leq \sum_i \beta_{ij} r_{ij} \leq e_m, \quad i = 1, \dots, p, \quad (35)$$

где e_m есть m -вектор со всеми компонентами, равными единице. Это эквивалентно неравенствам

$$\sum_i \beta_{ij} r_{ij} + v_i = e_m, \quad i = 1, \dots, p, \quad (36)$$

$$0 \leq v_i \leq 2e_m. \quad (37)$$

Полученная задача (21) — (22) и (36) — (37) имеет столько линейных ограничений, сколько индексов имеется в объедине-

нии B и всех множеств B_i , а также ограничений на переменные сверху. Это меньше, чем в задаче (46)—(50), сформулированной в § 8.3. Она также имеет угловую структуру, причем (22) представляют связующие ограничения, а (36)—(37)—диагональные блоки. Следовательно, для решения таких задач могут быть использованы любые специальные приемы (см. гл. II, IV, V). Как и в § 8.3, задача упрощается, когда одна или более из функций w_i является дифференцируемой в текущей точке y . Тогда, если $u_i^0 = \nabla w_i(y_i)$, то мы можем опустить переменные β_{ij} для всех j , положив $s_i = \sum_j \beta_{ij} r_{ij}$ в (22), положив $\sum_j c_{ij} \beta_{ij} = -u_i^0 s_i$ в (21) и заменив i -е множество ограничений в (36) и (37) на двухсторонние ограничения $-e_m \leq s_i \leq e_m$. Если все w_i дифференцируемы в y , то формулировки задачи выбора направления в этом и предшествующем параграфах становятся идентичными.

§ 8.5. АППРОКСИМАЦИЯ КАСАТЕЛЬНЫМИ

Джоффрион в [2] предложил альтернативный подход к решению координирующей программы. Он основывается на идее аппроксимации касательными,

обсуждавшейся выше в § 7.6. Далее предполагается знакомство с этим материалом. Так же как для двойственной функции $h(u)$, оценка $w_i(y_i)$ позволяет найти (возможно, с некоторыми дополнительными усилиями) и субградиент $u_i \in \partial w_i(y_i)$. Как было показано в § 8.3, $-u_i$ есть просто вектор оптимальных двойственных переменных для $P_i(y_i)$. Таким образом, если $\bar{u}_i \in \partial w_i(\bar{y}_i)$, то $w_i(\bar{y}_i) + \bar{u}_i(y_i - \bar{y}_i)$ является линейной опорной функцией к графику w_i в точке y_i . Пусть $u_i^1, \dots, u_i^r$ есть субградиенты w_i в точках $y_i^1, \dots, y_i^r$. Тогда аппроксимация касательными к w_i , показанная на рис. 8.4, есть выпуклая кусочно-линейная функция

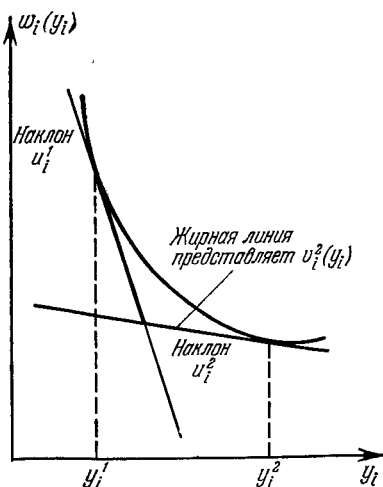


Рис. 8.4. Аппроксимация касательными к w_i .

званная на рис. 8.4, есть выпуклая кусочно-линейная функция

$$v_i^r(y_i) = \max_{1 \leq l \leq r} \{w_i(y_l^i) + u_i^l(y_i - y_l^i)\}. \quad (1)$$

Используя это, можно заменить координирующую программу на задачу вида

$$\min \sum_{i=1}^p v_i^r(y_i), \quad \sum_{i=1}^p y_i \leq b, \quad y_i \in V_i, \quad i=1, \dots, p. \quad (2)$$

Это в свою очередь эквивалентно задаче

$$\min \sum_{i=1}^p \sigma_i, \quad (3)$$

$$\sum_{i=1}^p y_i \leq b, \quad (4)$$

$$\sigma_i \geq w_i(y_i^j) + u_i^j(y_i - y_i^j), \quad j=1, \dots, r, \quad i=1, \dots, p, \quad (5)$$

$$y_i \in V_i, \quad i=1, \dots, p. \quad (6)$$

Эта задача линейна, за исключением ограничений $y_i \in V_i$. Можно описать r -й цикл итеративной процедуры следующим образом:

1. Решить задачу (3)–(6), получив оптимальное решение $y^{r+1} = (y_1^{r+1}, \dots, y_p^{r+1})$.

2. Для $i=1, \dots, p$ решить $P_i(y_i^{r+1})$, получая оптимальное значение целевой функции $w_i(y_i^{r+1})$ и субградиент u_i^{r+1} .

3. Добавить к решенной задаче p линейных ограничений типа (5), образуемых с помощью точек (y_i^{r+1}, u_i^{r+1}) , заменить r на $r+1$ и перейти к шагу 1.

Как и в § 7.6, здесь немедленно оцениваются верхние и нижние границы минимального значения целевой функции координирующей задачи. Верхняя граница дается выражением $\sum_{i=1}^p w_i(y_i^{r+1})$, в то время как минимальное значение (3) дает нижнюю границу. Когда же эти границы совпадут, то y^{r+1} дает решение координирующей задачи. При изложении метода мы молчаливо предполагали, что подзадача i имеет оптимальное решение, и что существует вектор оптимальных двойственных переменных. Для того чтобы обеспечить это, мы предположим для $i=1, \dots, p$, что справедливы:

Предположение 1. S_i компактно и выпукло.

Предположение 2. f_i и все компоненты g_i выпуклы и непрерывны в S_i .

Эти предположения представляют собой ослабленный вариант предположений, сделанных в § 8.2. Как было показано ранее, существование двойственных переменных обеспечено, если $y_i \in \text{int } V_i$, так что нам придется сделать дополнительные оговорки только по поводу граничных точек. Таким образом, мы можем поставить следующее условие:

Предположение 3. Для $i=1, \dots, p$ подзадача i имеет оптимальные двойственные переменные для всех y_i на границе V_i .

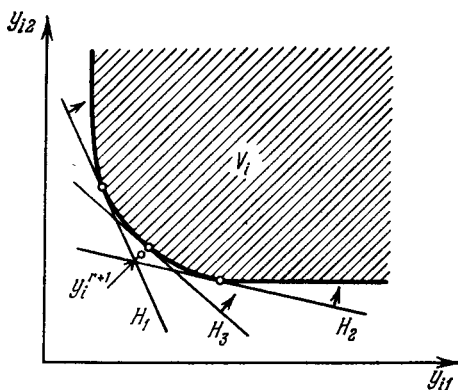
К сожалению, не существует никаких простых условий на f_i и g_i , которые гарантировали бы выполнение этого предположения. Альтернатива состоит в том, чтобы вводить возмущения любого вектора на границе так, чтобы он сдвигался хотя бы слегка в нужной области. Например, $y_i + \delta$ лежит в $\text{int } V_i$, если $y_i \in V_i$ и δ — вектор со всеми положительными компонентами. Если обеспечить, чтобы δ стремилось к нулю, то оптимальное решение, находящееся на границе, будет достигаться как предельное значение.

Поскольку оптимальные решения координирующей программы стремятся не попадать в точки y_i , такие, что $Dw_i(y_i) = \emptyset$ для некоторого i , проблема существования двойственных переменных, вероятно, не столь серьезна. Чтобы показать это, используем теорему 13 Приложения 2, из которой следует, что если $y_i \in V_i$, то $Dw_i(y_i) = \emptyset$, если и только если $Dw_i(y_i; s_i)$ не является правильной по s_i . Поскольку $Dw_i(y_i; 0) = 0$, это может случиться, если и только если $Dw_i(y_i; s_i) = -\infty$ для некоторого s_i , т. е. w_i уменьшается с бесконечным темпом, когда мы сдвигаемся от y_i внутрь V_i). Таким образом, алгоритм минимизации будет стремиться уводить решение от точек y_i , таких, что $Dw_i(y_i)$ является пустым.

Проблема, которой мы еще не касались состоит в том, как удовлетворять ограничениям вида $y_i \in V_i$ в (3) — (6). В этом и состоит главная трудность применения метода аппроксимации касательными и координирующей программе. Сходные трудности не возникают в двойственном методе § 7.6, так как там имеются только ограничения вида $u \geq 0$. Более того, проблема здесь оказывается более сложной, чем в процедуре выбора подходящего направления, рассмотренной ранее, вследствие того, что и теперь не имеется никакого способа для аппроксимации границы V_i постепенно, путем малых шагов вдоль заданного направления.

*) В работе [7, стр. 79], показано, что если $Dw_i(y_i; s_i) = -\infty$ для некоторого s_i , то оно равно $-\infty$ для всех s_i , таких, что $s_i \sim y_i$ лежат внутри V_i .

Поскольку явный вид ограничений, определяющих V_i , получить трудно, за исключением случая размерности один или два (см. [2]), мы рассмотрим две схемы построения многогранной аппроксимации V_i . В одной V_i аппроксимируется пересечением опорных полупространств, а в другой описывается как выпуклый многогранник, определяемый точками в V_i . В последнем случае обеспечивается допустимость на каждом шаге, в то время как в первом этого нельзя гарантировать.

Рис. 8.5. Внешняя линеаризация v_i .

Внешняя линеаризация V_i . Способ генерации опорных полупространств для V_i (см. рис. 8.5) базируется на следующей теореме.

Теорема 1. $y_i \in V_i$, если и только если y_i удовлетворяет (несчетно бесконечной) системе линейных неравенств

$$\lambda y_i \geq \min_{x_i \in S_i} \lambda g_i(x_i), \text{ для всех } \lambda \in \Lambda, \quad (7)$$

где

$$\Lambda = \left\{ \lambda / \sum_{j=1}^m \lambda_j = 1, \lambda_j \geq 0, j = 1, \dots, m \right\}. \quad (8)$$

Более того, каждое линейное неравенство (7) определяет опорное полупространство для V_i .

Доказательство требует использования следующей леммы.

Лемма 1 [8, стр. 84]. Пусть S есть выпуклое компактное множество в E^n и $f_1(x), \dots, f_m(x)$ — выпуклые непрерывные функции, определенные на S . Если не существует $x \in S$, такого, что $f_i(x) \leq 0, i = 1, \dots, m$, то существует m -вектор $\lambda \in \Lambda$, такой, что $\min_{x \in S} \sum_{i=1}^m \lambda_i f_i(x) > 0$.

Доказательство теоремы 1. Необходимость. Если $y_i \in V_i$, то существует $x_i \in S_i$, такое, что

$$\lambda y_i > \lambda g_i(x_i), \quad \forall \lambda \in E^m, \quad \lambda \geq 0.$$

Осуществляя минимизацию правой части на S_i и рассматривая только $\lambda \in \Lambda$, приходим к (7).

Достаточность. Предположим, что $y_i \notin V_i$, т. е. не существует $x_i \in S_i$, такого, что $g_i(x_i) \leq y_i$. При предположениях 1 и 2 этого параграфа применима лемма 1. Тогда имеется вектор $\lambda \in \Lambda$, такой, что $\min_{x_i \in S_i} \lambda(g_i(x_i) - y_i) > 0$, т. е. $\min_{x_i \in S_i} \lambda g_i(x_i) > \lambda y_i$, так что одно из неравенств (7) нарушается.

Рассмотрим теперь построение опорных полупространств. Ясно, что для любого $\lambda \in \Lambda$ полупространство

$$R_i(\lambda) = \{y_i / \lambda y_i \geq \min_{x_i \in S_i} \lambda g_i(x_i)\} \quad (9)$$

содержит V_i , поскольку из $y_i \in V_i$ следует $y_i \in R_i(\lambda)$. Это полупространство является опорным, так как его граница контактирует с V_i в точке $y_i = g_i(x_i(\lambda)) \in V_i$, где $x_i(\lambda)$ минимизирует $\lambda g_i(x)$ на S_i . Так как V_i определена бесконечным множеством линейных ограничений, естественно применить стратегию релаксации, описанную в § 4.2. Если на шаге 1 процедуры аппроксимации касательными y^{r+1} таково, что $y_i^{r+1} \notin V_i$ для некоторого i (что сигнализирует о недопустимости $P_i(y_i^{r+1})$ на шаге 2), то, по теореме 1, y_i^{r+1} нарушает одно из ограничений (7). Одно или более нарушенных ограничений должны быть тогда генерированы и добавлены к (3) — (6)*. Пример показан на рис. 8.5, где V_i уже аппроксимировано двумя полупространствами, определенными гиперплоскостями H_1 и H_2 . Решение y_i^{r+1} лежит вне V_i и должно быть добавлено полупространство, определенное гиперплоскостью H_3 , которая исключает y_i^{r+1} . Этот подход был использован Келли в его методе секущих плоскостей [5].

Для того чтобы иметь возможность реализовать эту стратегию, алгоритм, который решает подзадачу i , должен быть способен начинаться с недопустимой точки, и, если подпроблема i является недопустимой, должен выделять нарушенное ограничение (7). Как показано в [2], большинство двойственных методов делает это автоматически, равно как и фаза 1 процедур, используемых для нахождения начальной допустимой точки в прямых алгоритмах. Отметим, что добавление линейных ограничений в (3) — (6) приводит к блочно-диагональной линейной задаче со связующими ограничениями, которая допускает решение с помощью множества алгоритмов, обсуждавшихся выше в главах II, IV, V.

Линейный случай. Процедура релаксации упрощается, когда g_i является линейной, а S_i — многогранным. Для определенности положим

$$g_i(x_i) = Ax_i, \quad S_i = \{x_i / B_i x_i = b_i, x_i \geq 0\}, \quad (10)$$

*) Строго удовлетворяющиеся ограничения также могут быть опущены.

так что

$$V_i = \{y_i / \exists x_i \geq 0 \text{ такое, что } B_i x_i = b_i, A_i x_i \leq y_i\}. \quad (11)$$

Поскольку V_i многогранно, то оно определяется конечным множеством ограничений, которые могут быть найдены с помощью леммы Фаркаша, как это уже делалось в алгоритме Бендерса (§ 6.3). Результат состоит в том, что вектор y_i принадлежит V_i , если и только если

$$b'_i u'_{1j} + y'_i u'_{2j} \geq 0, \quad \forall j, \quad (12)$$

где (u'_{1j}, u'_{2j}) j -й образующий луч выпуклого многогранного конуса

$$C = \{(u_1, u_2) / B'_i u_1 + A'_i u_2 \geq 0, u_2' \geq 0\}. \quad (13)$$

Число ограничений вида (12), вообще говоря, очень велико. Для того чтобы проверить, удовлетворяются ли все они некоторым определенным вектором $\bar{y}_i$, рассмотрим линейную задачу

$$\min b'_i u_1 + \bar{y}_i u_2, \quad (u_1, u_2) \in C. \quad (14)$$

Все ограничения удовлетворяются, если и только если оптимальное значение целевой функции выпуклой задачи равно нулю. В противном случае минимальное значение является неограниченным, и должен быть найден крайний луч, с помощью которого строится нарушенное ограничение. Вследствие однородности (14) такой крайний луч, возможно, легче найти применением фазы I операции к двойственной к (14), в которой коэффициенты целевой функции равны нулю, а ограничениями являются (11). Подходящий двойственный крайний луч может быть легко получен из конечной симплексной таблицы, если эта задача не имеет допустимого решения. Полученное таким способом ограничение добавляется к (3)–(6) вместе с подобными же ограничениями, находимыми при других значениях индекса i , и ограничениями, связанными с аппроксимацией. После этого задача (3)–(6) решается заново.

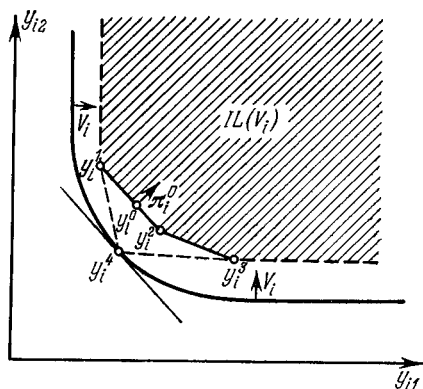


Рис. 8.6. Внутренняя линеаризация V_i .

Внутренняя линеаризация V_i . Другой подход, при котором сохраняется допустимость координирующей задачи, состоит в

аппроксимации V_i изнутри, как показано на рис. 8.6. Здесь точки y_i^j , $j=1, 2, 3$ лежат в V_i , а аппроксимирующее множество, внутренняя линеаризация ВЛ (V_i), есть множество всех точек, лежащих не выше и справа от выпуклой комбинации точек y_i^j :

$$\text{ВЛ}(V_i) = \left\{ y_i / y_i \geq \sum_{j=1}^{r_i} \alpha_{ij} y_i^j, \alpha_{ij} \geq 0, \sum_{j=1}^{r_i} \alpha_{ij} = 1 \right\}. \quad (15)$$

Это та же самая тактика, которая была использована при построении обобщенного линейного программирования (см. § 3.3. и 3.4). Поскольку V_i выпукло и неограниченно по любому направлению с неотрицательными координатами, то ВЛ (V_i) содержится в V_i . Когда каждое V_i обработано таким способом, задача (2) принимает вид

$$\min \sum_{i=1}^p v_i^r(y_i) \quad (16)$$

при ограничениях $\sum_{i=1}^p y_i \leq b$, и, для $i=1, \dots, p$, $y_i \geq \sum_{j=1}^{r_i} \alpha_{ij} y_i^j$,

$$\sum_{j=1}^{r_i} \alpha_{ij} = 1, \alpha_{ij} \geq 0, \forall j.$$

Если v_i^r построены так же, как в (3) — (6), то (16) вновь является блочно-диагональной линейной программой со связующими ограничениями.

Для улучшения аппроксимации V_i можно использовать следующее утверждение.

Теорема 2. Пусть $\{(y_1^0, \dots, y_p^0), (\alpha_1^0, \dots, \alpha_p^0), (\sigma_1^0, \dots, \sigma_p^0)\}$ — решение (16), и пусть π_i^0 , $i=1, \dots, p$ — векторы оптимальных двойственных переменных, соответствующих ограничениям $y_i \geq \sum_j \alpha_{ij} y_i^j$. Тогда для $i=1, \dots, p$ y_i^0 лежит на границе текущей

внутренней линеаризации, ВЛ (V_i), π_i^0 есть вектор нормали к опорной гиперплоскости к ВЛ (V_i), и эта гиперплоскость содержит y_i^0 .

Доказательство. Вектор π_i^0 удовлетворяет следующим двойственным ограничениям, образуемым с помощью столбцов прямой задачи, ассоциированных с α_{ij} :

$$\pi_i^0 y_i^j \geq \gamma_i^0, \quad j=1, 2, \dots, r_i, \quad (17)$$

где γ_i^0 есть оптимальная двойственная переменная, связанная с ограничением $\sum_j \alpha_{ij} = 1$. Условия дополняющей нежесткости и условия

неотрицательности также удовлетворяются:

$$\pi_i^0 \geq 0, \quad (18)$$

$$\pi_i^0 \left(y_i^0 - \sum_{j=1}^{r_i} \alpha_{ij}^0 y_i^j \right) = 0. \quad (19)$$

Пусть $\alpha_i = (\alpha_{i1}, \dots, \alpha_{ir})$ — неотрицательный вектор, компоненты которого в сумме равны единице. Если ограничения j из (17) умножить на α_{ij} , и результат просуммировать по j , то

$$\pi_i^0 \left(\sum_{j=1}^{r_i} \alpha_{ij} y_i^j \right) \geq \gamma_i^0. \quad (20)$$

Если $y_i \geq \sum_{j=1}^{r_i} \alpha_{ij} y_i^j$, то, поскольку $\pi_i^0 \geq 0$, имеем $\pi_i^0 y_i \geq \gamma_i^0$.

Таким образом, все y_i в ВЛ (V_i) лежат по одну сторону гиперплоскости

$$H_i^0 = \{y_i / \pi_i^0 y_i = \gamma_i^0\}. \quad (21)$$

В силу условий дополняющей нежесткости $\alpha_{ij}^0 (\pi_i^0 y_i^j - \gamma_i^0) = 0$, $\forall j$. После суммирования

$$\pi_i^0 \left(\sum_{j=1}^{r_i} \alpha_{ij}^0 y_i^j \right) = \gamma_i^0. \quad (22)$$

Используя (19), получаем

$$\pi_i^0 y_i^0 = \pi_i^0 \left(\sum_{j=1}^{r_i} \alpha_{ij}^0 y_i^j \right), \quad (23)$$

так что, комбинируя (22) и (23), находим $\pi_i^0 y_i^0 = \gamma_i^0$. Таким образом, H_i^0 должна быть опорной гиперплоскостью к ВЛ (V_i). Она содержит y_i^0 , а ее вектор нормали равен π_i^0 . Отсюда следует, что y_i^0 лежит на границе ВЛ (V_i).

Используем этот результат для улучшения аппроксимации V_i . Примем π_i^0 (соответственно нормированное) в качестве вектора нормали опорной гиперплоскости к V_i , найдем точку контакта этой плоскости с V_i и добавим эту точку к множеству $\{y_i^j\}$. Эта процедура иллюстрируется рис. 8.6, где y_i^4 — новая точка. Новая многогранная аппроксимация ограничена пунктирными линиями. Эта процедура имеет то преимущество, что добавляемые точки всегда оказываются граничными точками V_i .

и в некотором смысле близки к предшествующему решению y_i^0 . Предполагая, что истинное оптимальное решение y_i лежит вблизи y_i^0 , мы видим, что аппроксимация V_i уточняется именно там, где это наиболее важно, в окрестности оптимума. Вычислим новое значение y_i^1 . Заметим, что опорная гиперплоскость к V_i с нормалью π_i^0 в силу (7) определена линейным уравнением

$$\pi_i^0 y_i = \min_{x_i \in S_i} \pi_i^0 g_i(x). \quad (24)$$

Если мы решим задачу выпуклого программирования

$$\min \{\pi_i^0 g_i(x) / x_i \in S_i\},$$

то получим решение x_i^0 и можем положить $y_i^{r+1} = g_i(x_i^0)$. Ясно, что y_i^{r+1} удовлетворяет (24) и является желаемой точкой.

Задачи

1. Доказать, что если f_i — строго выпукла, и если предположения 1 и 3 § 8.2 выполнены, то w_i — строго выпукла.
2. Обосновать вычисление $w(y)$ в примере 1 § 8.2 и соответствующие вычисления $h(u)$ в § 8.3.
3. Показать, что если для $i=1, \dots, p$ имеет место $-u_i \in U_i$, т. е. $u_i \in \partial w_i(y_i)$, то $(u_1, \dots, u_p)$ есть субградиент функции $w = \sum_{i=1}^p w_i$ в точке $y = (y_1, \dots, y_p)$.
4. Вывести неравенства, определяющие V_i в п. 8.5-12.

ЛИТЕРАТУРА

1. G. J. Silverman, Primal Decomposition of Mathematical Programs by Resource Allocation. Tech. Memorandum 116, Operations Research Department, Case Western Reserve University, 1968.
2. A. M. Geoffrion, Primal Resource — Directive Approaches for Optimizing Nonlinear Decomposable Systems. Memorandum RM-5829-PR, The RAND Corporation, Santa Monica, Calif., 1968.
3. J. M. Falk, A Constrained Lagrangian Approach to Nonlinear Programming. Technical Report, Department of Mathematics, University of Michigan, 1965, pp. 76—83.
4. E. V. W. Zschau, A Primal Decomposition Algorithm for Linear Programming. Graduate School of Business, Stanford University, Working Paper 91, 1967.
5. J. E. Kelley, Jr., The Cutting Plane Method for Solving Convex Programs. J. Soc. Ind. Appl. Math., No. 8, pp. 703—712, 1960.
6. D. M. Topkis and A. Veinott, On the Convergence of Some Feasible Direction Algorithms for Nonlinear Programming. J. SIAM Control, 5, No. 2, 1967.
7. W. Fenchel, Convex Cones, Sets, and Functions, lecture notes, Princeton University, 1951.
8. C. Berge and A. Ghouila-Houri, Programming, Games, and Transportation Networks, John Wiley & Sons, New York, 1965.

ПРИЛОЖЕНИЕ 1

ВЫПУКЛЫЕ ФУНКЦИИ И ИМ СОПРЯЖЕННЫЕ *)

Здесь мы будем рассматривать функции, которые могут принимать значения $\pm\infty$. Это потребует следующего обобщенного определения выпуклости.

Определение 1. Пусть f — функция, преобразующая выпуклое множество C в $[-\infty, +\infty]$. Тогда f *выпукла*, если

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha \mu_1 + (1 - \alpha)\mu_2 \quad (1)$$

для всех $x_1, x_2 \in C$, всех $f(x_1) \leq \mu_1 \in E^1, f(x_2) \leq \mu_2 \in E^1, 0 < \alpha < 1$.

Если f не принимает значений $+\infty$ и $-\infty$, то в (1) не может возникнуть комбинация типа $(\infty, -\infty)$, и (1) упрощается до обычного определения

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2). \quad (2)$$

Задавая функцию, выпуклую на C , мы можем распространить ее на E^n , определяя $f(x) = +\infty$ для $x \notin C$. Инфимум этой функции на E^n равен ее инфимуму на C , так что эта конструкция заменяет минимизацию с ограничениями формально неограниченной. В остающейся части этого приложения все выпуклые функции будут именно такого расширенного типа из E^n в $[-\infty, +\infty]$.

Определение 2. *Эпиграфом* **) выпуклой функции f , обозначаемым $\text{epi } f$, назовем множество всех (конечных) точек на графике и выше графика f :

$$\text{epi } f = \{(x, \mu) / x \in E^n, \mu \in E^1, \mu \geq f(x)\}. \quad (3)$$

Легко показать что f выпукла, если и только если $\text{epi } f$ является выпуклым множеством.

Определение 3. *Эффективной областью выпуклой функции* называется множество

$$\text{dom } f = \{x / f(x) < +\infty\}. \quad (4)$$

*) Приложения 1 и 2 основываются главным образом на работах [1] и [3]. Полное изложение всей проблематики содержится в [8].

**) Употребляется также термин *надграфик*. (Прим. перев.)

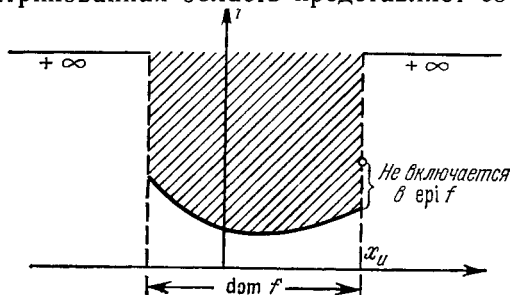
Множество $\text{dom } f$ есть проекция $\text{epi } f$ на E^n , т. е.

$$\text{dom } f = \{x / \exists \mu \in E^1, \text{ такое, что } (x, \mu) \in \text{epi } f\}. \quad (5)$$

Очевидно, что если f выпукла, то и $\text{dom } f$ выпукло.

Определение 4. Говорят, что выпуклая функция f является *собственной*, если $\text{dom } f$ непусто и $f(x) > -\infty$, $\forall x \in \text{dom } f$.

Собственная выпуклая функция показана на рис. П. 1.1. Заштрихованная область представляет собой $\text{epi } f$. Отметим, что f



имеет разрыв при $x = x_0$, и, следовательно, $\text{epi } f$ не является замкнутым множеством.

Определение 5. Относительной *внутренностью* выпуклого множества C , обозначаемой $r \text{ int } C$, называется внутренность относительно наименьшего аффинного многообразия, содержащего его.

Рис. П. 1.1. Собственная выпуклая функция.

Аффинное многообразие есть просто перенос от начала координат линейного подпространства, т. е. множество вида $\{x_0\} + V$, где x_0 — фиксированная точка, а V — линейное подпространство. Гиперплоскость, которая не содержит начала, является аффинным многообразием. Наименьшее такое многообразие, содержащее C , есть пересечение всех аффинных многообразий, содержащих C . Если линейное подпространство, ассоциированное с этим многообразием, имеет размерность r , то относительно внутренние точки определяются с использованием окрестности размерности r . Например, на рис. П. 1.2 сплошная линия означает $C \subseteq E^2$, является выпуклой и не имеет никакой внутренней. Наименьшее аффинное многообразие, содержащее C , есть вся прямая, отрезком которой является C . Окрестности, которые должны быть использованы, просто являются интервалами. Любая точка в C , такая, что некоторый интервал, окружающий ее, также лежит в C , принадлежит $r \text{ int } C$.

В последующем изложении $\text{cl } C$ означает замыкание множества C .

Определение 6. *Относительной границей* выпуклого множества C называется множество

$$\text{rb } C = \{x / x \in \text{cl } C \text{ и } x \notin r \text{ int } C\}. \quad (6)$$

Установим две основные теоремы, которые широко используются далее.

Теорема 1. *Непустое выпуклое множество имеет непустую относительную внутренность.*

Теорема 2. *Пусть f — собственная выпуклая функция, определенная на E^n , и пусть $x_1 \in r \text{ int dom } f$. Тогда существует опорная гиперплоскость к графику f в точке x_1 , т. е. существует некоторое $x^0 \in E^n$, такое, что*

$$f(x) \geq f(x_1) + x^0(x - x_1), \quad \forall x \in E^n. \quad (7)$$

Доказательство теоремы 1 можно найти в [4, стр. 9]. Теорема 2 следует из классического результата для функций вещественного переменного,

определенных на выпуклом множестве C ([5, стр. 398] и [6, стр. 28]). Множество $\text{dom } f$ играет роль C , и соотношение выполняется также и для x , не лежащих в $\text{dom } f$, поскольку там $f = +\infty$.

Теорема 3. *Если f — собственная и выпуклая функция, то существует точка (x^*, μ^*) , такая, что*

$$f(x) \geq x^*x - \mu^*, \quad \forall x \in E^n. \quad (8)$$

Доказательство. Так как f — собственная функция, то $\text{dom } f$ выпукло и непусто, а следовательно, содержит относительно внутреннюю точку x_1 . Тогда, по теореме 2,

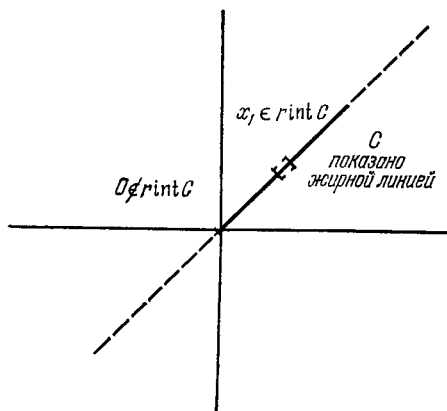


Рис. П. 1.2. Относительная внутренность выпуклого множества.

$$f(x) \geq x^*x - (x^*x_1 - f(x_1)) = x^*x - \mu^*, \quad \forall x \in E^n. \quad (9)$$

Теорема 4. *Если f является несобственной выпуклой функцией, то либо $f(x) = +\infty$ для всех $x \in E^n$, либо $f(x) = -\infty$ для всех $x \in r \text{ int dom } f$.*

Доказательство. Если $\text{dom } f = \emptyset$, то $f(x) = +\infty$ для всех $x \in E^n$. Если $\text{dom } f \neq \emptyset$, то выберем $x_1 \in r \text{ int dom } f$ и предположим, что $f(x_1)$ конечно. Тогда в силу (5) $f(x) > -\infty$ для всех $x \in E^n$, что противоречит предположению, что f является несобственной. Таким образом, $f(x_1) = -\infty$.

Определение 7. Замыкание выпуклой функции f , обозначаемое $\text{cl } f$, есть функция, чье значение в x таково:

$$(\text{cl } f)(x) = \sup_{(x^*, \mu^*) \in D} [x^*x - \mu^*], \quad (10)$$

где

$$D = \{(x^*, \mu^*)/x^* \in E^n, \mu^* \in E^1, x^*x - \mu^* \leq f(x) \forall x \in E^n\}. \quad (11)$$

Геометрически $(\text{cl } f)(x)$ есть супремум по x всех линейных аффинных функций, которые лежат на или ниже графика f .

Определение 8. Говорят, что выпуклая функция f *замкнута*, если $f(x) = (\text{cl } f)(x)$ для всех $x \in E^n$.

Функция на рис. П. 1.1 не замкнута, так как $(\text{cl } f)(x_u) \neq f(x_u)$. Здесь число $(\text{cl } f)(x_u)$ есть предел $\{f(x_i)\}$, когда $\{x_i\}$ приближается к x_u слева.

Может быть показано (см. [2]), что если f — собственная функция, эпиграф $\text{cl } f$ есть замыкание эпиграфа f . Таким образом, $\text{cl}(\text{cl } f) = \text{cl } f$ и $\text{cl } f$ есть замкнутая правильная выпуклая функция на E^n , и f замкнута, если и только если $\text{epi } f$ замкнуто.

Теорема 5. а) $(\text{cl } f)(x) \leq f(x)$ для всех $x \in E^n$.

б) $(\text{cl } f)(x) = f(x)$ для всех $x \notin \text{rb dom } f$.

Утверждение а) прямо следует из определения, в то время как б) есть прямое следствие того факта, что выпуклая функция на открытом множестве непрерывна на этом множестве. Очевидно из изложенного, что замыкание выпуклой функции есть просто операция регуляризации, которая доопределяет f в некоторых граничных точках. Рокафеллар устанавливает в [2], что если f — собственная и выпуклая функция, то она замкнута, если и только если она является полунепрерывной снизу и

$$(\text{cl } f)(x) = \liminf_{y \rightarrow x} f(y), \quad \forall x \in E^n. \quad (12)$$

Понятие сопряженности требует полного изучения множества D , определенного в силу (11).

Теорема 6. а) если f является несобственной и $\text{dom } f \neq \emptyset$, то $D = \emptyset$.

б) если f является несобственной и $\text{dom } f \neq \emptyset$, то $D = E^{n+1}$.

в) если f является собственной, то D выпукло и $\emptyset \neq D \neq R^{n+1}$.

Доказательство. В силу теоремы 4, в случае а) $f(x) = -\infty$ для всех $x \in \text{int dom } f$. Тогда $D = \emptyset$. В случае б) $f(x) = +\infty$ для всех $x \in E^n$ и $D = R^{n+1}$. В случае в), поскольку существует x , для которого f конечна и $f(x) > -\infty$ для всех $x \in E^n$, то D должно быть непусто, но не должно совпадать с E^{n+1} . Для того чтобы доказать выпуклость D , выберем $(x_1^*, \mu_1^*) \in D$, $(x_2^*, \mu_2^*) \in D$. Тогда

$$x_1^*x - \mu_1^* \leq f(x), \quad \forall x \in E^n, \quad (13)$$

$$x_2^*x - \mu_2^* \leq f(x), \quad \forall x \in E^n. \quad (14)$$

Умножая (13) на α и (14) на $(1-\alpha)$, где $0 < \alpha < 1$, и суммируя эти неравенства, получаем

$$(\alpha x_1^* + (1-\alpha)x_2^*)x - (\alpha\mu_1^* + (1-\alpha)\mu_2^*) \leq f(x), \quad \forall x \in E^n,$$

так что $(\alpha(x_1^*, \mu_1^*) + (1-\alpha)(x_2^*, \mu_2^*)) \in D$.

Определение 9. Пусть f выпукла на E^n . Сопряженной к f называется функция f^* , определенная на E^n и задаваемая условием

$$f^*(x^*) = \sup_x (x^*x - f(x)). \quad (15)$$

Функция f^* имеет интересную интерпретацию как в геометрических, так и в экономических терминах. Из определяющего соотношения (15) следует

$$f(x) \geq x^*x - f^*(x^*), \quad \forall x \in E^n,$$

т. е. аффинная функция $h(x) = x^*x - f^*(x^*)$ лежит всюду ниже графика f . Если супремум (15) достигается, скажем, при $x = x_1$, то $h(x)$ — опорная функция для f при $x = x_1$.

Как видно из рис. П. 1.3, — $f^*(x^*)$ есть пересечение этой опорной функции с осью f .

Если супремум не достигается, то не существует никакой опорной функции с наклоном x^* , и — $f^*(x^*)$ есть супремум пересечений f со всеми линейными функциями, имеющими наклоны x^* , и лежащими целиком ниже графика f .

С экономической точки зрения представим, что затраты на производство некоторого продукта в количестве x есть $f(x)$, а цена этого продукта составляет x^* . Тогда чистый доход, получаемый от продажи x единиц, есть $x^*x - f(x)$, и максимальный доход, получаемый при цене x^* , есть $f^*(x^*)$. Имеют место следующие свойства сопряженных функций.

Теорема 7. Если f выпукла, то f^* выпукла и $\text{epi } f^* = D$.

Доказательство. Множество D в (11) может быть записано в виде

$$D = \{(x^*, \mu^*) / x^* \in E^n, \mu^* \in E^1, \mu^* \geq \sup_x (x^*x - f(x)) = f^*(x^*)\}. \quad (16)$$

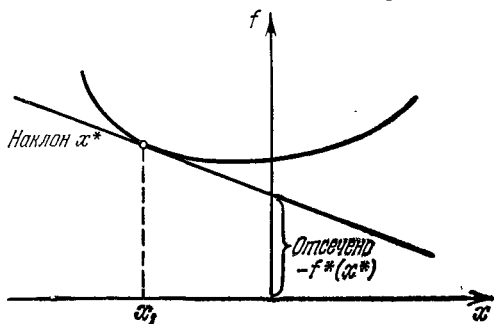


Рис. П. 1.3. Геометрическая интерпретация $f^*(x^*)$.

Таким образом, $\text{epi } f^* = D$. В силу теоремы 6 D выпукло, так что f^* — выпуклая функция.

Определение 10. Функция, дважды сопряженная к f , f^{**} , есть просто функция, сопряженная к f^* и даваемая соотношением

$$f^{**}(x) = \sup_{x^*} (x^*x - f^*(x^*)). \quad (17)$$

Теорема 8. Пусть

$$D^* = \{(x, \mu) / x \in E^n, \mu \in E^1, x^*x - \mu \leq f^*(x^*), \forall x^* \in E^n\}. \quad (18)$$

Тогда

- a) $\text{cl } f = f^{**},$
- b) $D^* = \text{epi } f^{**} = \text{epi } (\text{cl } f),$
- c) $f^{**}(x) = \sup_{x^*} \inf_y (x^*(x - y) + f(y)).$

Доказательство.

$$\begin{aligned} \text{a) } (\text{cl } f)(x) &= \sup_{(x^*, \mu^*) \in D} (x^*x - \mu^*) = \sup_{\mu^* \geq f^*(x^*)} (x^*x - \mu^*) = \\ &= \sup_{x^*} (x^*x - f^*(x^*)) = f^{**}(x). \end{aligned}$$

b) Доказательство такое же, как и в теореме 7.

$$\begin{aligned} \text{c) } f^{**}(x) &= \sup_{x^*} (x^*x - \sup_y (x^*y - f(y))) = \\ &= \sup_{x^*} (x^*x + \inf_y (f(y) - x^*y)) = \sup_{x^*} \inf_y (x^*(x - y) + f(y)). \end{aligned}$$

Теорема 9. Пусть f выпукла. Тогда а) f^* замкнута, б) f^* собственная, если и только если f собственная.

Доказательство.

Для а), по определению, имеем

$$\begin{aligned} (\text{cl } f^*)(x^*) &= \sup_{(x, \mu) \in D^*} (x^*x - \mu) = \sup_{\mu \geq f^{**}(x)} (x^*x - \mu) = \\ &= \sup_{\mu \geq (\text{cl } f)(x)} (x^*x - \mu) = \sup_x (x^*x - (\text{cl } f)(x)) = (\text{cl } f)^*(x^*). \end{aligned}$$

Так как, по теореме 5, $(\text{cl } f)(x) \leq f(x)$,

$$x^*x - (\text{cl } f)(x) \geq x^*x - f(x),$$

так что $\text{cl } f^* = (\text{cl } f)^* \geq f^*$. Но, по теореме 5, $\text{cl } f^* \leq f^*$. Таким образом, $\text{cl } f^* = f^*$ и f^* замкнута.

Для того, чтобы доказать б), отметим, что в силу рассуждений, использованных при доказательстве теоремы 6, f^* — собственная, если и только если $\emptyset \neq \text{epi } f^* \neq E^{n+1}$. (Замечание: f^* замкнута, и замыкание функции, которая равна где-либо $-\infty$,

равно $-\infty$ всюду.) По теореме 7 $\text{epi } f^* = D$ и в силу теоремы 6 $\emptyset \neq D \neq E^{n+1}$, если и только если f собственная.

Теорема 10. Если f выпукла, замкнута и собственная, то

$$f(x) + f^*(x^*) \geq x^*x, \quad \forall x \in E^n, \quad \forall x^* \in E^n. \quad (19)$$

Более того, если $x \in r \text{ int dom } f$, то существует x^* такое, что (19) выполняется как равенство. Двойственно, если $x^* \in r \text{ int dom } f^*$, то существует такое x , что (19) выполняется как равенство.

Доказательство. По определению,

$$f^*(x^*) \geq x^*x - f(x), \quad \forall x \in E^n, \quad \forall x^* \in E^n,$$

так что (19) выполняется. Для того чтобы доказать остающиеся утверждения, поскольку f собственная, то $\text{dom } f$ непусто, и в силу теоремы 1 имеет непустую относительную внутренность. Выберем $x_1 \in r \text{ int dom } f$. Тогда в силу теоремы 2 существует x^* , такое, что

$$f(x) \geq f(x_1) + x^*(x - x_1), \quad \forall x \in E^n$$

или

$$x^*x_1 - f(x_1) \geq x^*x - f(x).$$

Поскольку равенство сохраняется для $x = x_1$, то

$$x^*x_1 - f(x_1) = \sup_x \{x^*x - f(x)\} = f^*(x^*).$$

Для того чтобы доказать двойственный результат для f^* , убеждаемся, что из $x^* \in r \text{ int dom } f^*$ следует, что существует x , такое, что

$$f^*(z^*) \geq f^*(x^*) + x(z^* - x^*), \quad \forall z^*.$$

Повторение этих манипуляций приводит к результату

$$(\text{cl } f)(x) = f^{**}(x) = x^*x - f^*(x^*).$$

Так как f замкнута, $(\text{cl } f)(x) = f(x)$ и теорема доказана.

ПРИЛОЖЕНИЕ 2

СУБГРАДИЕНТЫ И ПРОИЗВОДНЫЕ ПО НАПРАВЛЕНИЮ ВЫПУКЛЫХ ФУНКЦИЙ *)

В этом приложении f предполагается выпуклой функцией, определенной на E^n со значениями в $[-\infty, +\infty]$.

Субградиенты.

Определение 1. Вектор-строка x^* называется *субградиентом* f в точке x , если

$$f(z) \geq f(x) + x^*(z - x), \quad \forall z \in E^n. \quad (1)$$

Определение 2. Множество всех субградиентов f в точке x обозначается $\partial f(x)$.

Как видно из рис. П. 2.1, x^* просто представляет собой наклон опорной гиперплоскости к графику f в точке x . Если f дифференцируема в x , то такая плоскость — единственная и является касательной плоскостью, так что $x^* = \nabla f(x)$. Тогда производная f в x по направлению s есть

$$Df(x; s) = \nabla f'(x) s.$$

Однако выпуклые функции не обязательно являются дифференцируемыми всюду. Тем не менее, когда f конечна, производные по направлению существуют

для всех направлений. Нашей главной целью в этом приложении является построение выражения для производной по направлению через субградиенты. Это потребует изучения некоторых свойств $Df(x; s)$ и $\partial f(x)$.

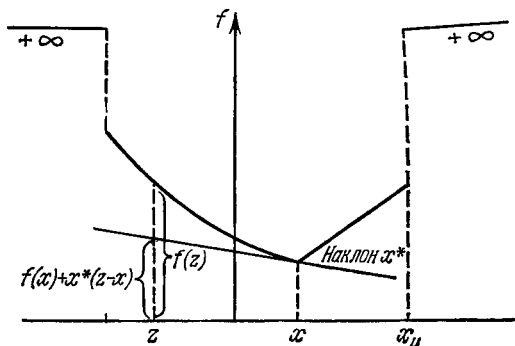


Рис. П. 2.1. Субградиент f в точке x .

*) Этот материал тесно связан с содержанием Приложения 1.

Теорема 1. Множество $\partial f(x)$ замкнуто и выпукло (возможно, пусто).

Доказательство. Для доказательства выпуклости рассмотрим x_1^* и $x_2^* \in \partial f(x)$. Тогда

$$f(z) \geq f(x) + x_1^*(z - x), \quad \forall z \in E^n, \quad (2)$$

$$f(z) \geq f(x) + x_2^*(z - x), \quad \forall z \in E^n. \quad (3)$$

Умножение (2) на α , (3) на $(1-\alpha)$, $0 < \alpha < 1$, и сложение неравенств дают

$$f(z) \geq f(x) + (\alpha x_1^* + (1-\alpha)x_2^*)(z - x), \quad \forall z \in E^n,$$

так что

$$(\alpha x_1^* + (1-\alpha)x_2^*) \in \partial f(x).$$

Для того чтобы показать замкнутость $\partial f(x)$, рассмотрим сходящуюся последовательность $\{x_i^*\}$ элементов из $\partial f(x)$ с пределом x^* и предположим, что $x^* \notin \partial f(x)$. Тогда существует $\hat{z} \in E^n$ и $\varepsilon > 0$, такие, что

$$f(\hat{z}) + \varepsilon = f(x) + x^*(\hat{z} - x). \quad (4)$$

Для любого элемента x_n^* последовательности $\{x_i^*\}$ должно выполняться

$$f(\hat{z}) \geq f(x) + x_n^*(\hat{z} - x). \quad (5)$$

Вычитая (4) из (5), получаем

$$\varepsilon \leq (x^* - x_n^*)(\hat{z} - x) \leq \|x^* - x_n^*\| \|\hat{z} - x\|.$$

Последнее соотношение следует из неравенства Шварца, причем через $\|y\|$ обозначена норма y . Таким образом,

$$\|x^* - x_n\| \geq \frac{\varepsilon}{\|\hat{z} - x\|} > 0, \quad \forall n,$$

что противоречит сходимости последовательности $\{x_i^*\}$ к x^* . Таким образом, $x^* \in \partial f(x)$, и $\partial f(x)$ замкнуто.

Теорема 2. Если f — собственная и $x \in \text{int dom } f$, то $\partial f(x) \neq \emptyset$.

Доказательство. Это немедленно следует из теоремы 2 Приложения 1.

Теорема 3. Если $\text{dom } f$ имеет непустую внутренность, и $x \in \text{int dom } f$, то $\partial f(x)$ компактно.

Доказательство. Так как x принадлежит внутренности $\text{dom } f$, то существует $\varepsilon > 0$, такое, что $N(x, \varepsilon) \subseteq \text{dom } f$, где N — открытая окрестность x радиуса ε . Если любая компонента x^* ,

скажем, $(x^*)_i$, может расти без ограничений, то

$$f(z) \geq f(x) + (x^*)_i(z-x)_i + \sum_{l \neq i} (x^*)_l(z-x)_l, \quad \text{для } \forall z \in E^n.$$

Выберем $z \in N(x, \epsilon)$ такое, что $(z-x)_i > 0$. Тогда при $(x^*)_i \rightarrow +\infty$ записанное выше неравенство нарушается, поскольку $f(z)$ конечна. Сходные аргументы сохраняют силу, если $(x^*)_i \rightarrow -\infty$, так что $\partial f(x)$ ограничено.

На рис. П. 2.1 $x_u \notin \text{int dom } f$ и $\partial f(x_u)$ неограничено, так как имеются опорные плоскости в x_u с наклонами, стремящимися к $+\infty$. Заметим, что условия $x \in \text{int dom } f$ недостаточно для обеспечения ограниченности $\partial f(x)$ (что видно из рассмотрения f на рис. П. 2.1), как функции, определенной на линии в E^2 , параллельной одной из осей. Тогда касательная линия в x становится плоскостью и может быть продолжена с использованием прямой на рис. П. 2.1 в качестве оси, сколь угодно далеко в любом направлении. При этом один из наклонов стремится к $+\infty$ или $-\infty$.

Теорема 4. $0 \in \partial f(x^0)$, если и только если x^0 минимизирует f .

Доказательство. $0 \in \partial f(x^0)$, если и только если

$$f(z) \geq f(x^0) + 0(z-x^0), \quad \text{для } \forall z \in E^n,$$

что сохраняет силу, если и только если x^0 минимизирует f .

Теорема 5. Если $\partial f(x) \neq \emptyset$, то $(\text{cl } f)(x) = f(x)$.

Доказательство. Если $x^* \in \partial f(x)$, то

$$f(z) \geq f(x) + x^*(z-x), \quad \text{для } \forall z \in E^n$$

и

$$f(z) \geq x^*z - [x^*x - f(x)] = x^*z - \mu^*.$$

Когда $z=x$, записанное неравенство превращается в равенство, откуда следует

$$f(x) = x^*x - \mu^* = \sup_{(y, \mu) \in D} (yx - \mu) = (\text{cl } f)(x),$$

где D есть множество, определенное формулой (11) в Приложении 1.

Существует тесная связь между субградиентами выпуклой функции и ее сопряженной, как показывают следующие два результата.

Теорема 6. $x^* \in \partial f(x)$, если и только если

$$f^*(x^*) + f(x) = x^*x. \quad (6)$$

Доказательство. $x^* \in \partial f(x)$, если и только если

$$f(z) \geq f(x) + x^*(z-x), \quad \text{для } \forall z \in E^n,$$

т. е. если и только если

$$x^*x - f(x) \geq x^*z - f(z), \quad \text{для } \forall z \in E^n,$$

переходящее в равенство при $z = x$. Отсюда следует

$$x^*x - f(x) = \sup_z \{x^*z - f(z)\} = f^*(x^*),$$

следовательно, теорема доказана.

Отметим, что последнее утверждение просто требует, чтобы неравенство в теореме 10 Приложения 1, которое выполняется в любой точке, выполнялось как равенство, когда $x^* \in \partial f(x)$.

Теорема 7. а) если $x^* \in \partial f(x)$, то $x \in \partial f^*(x^*)$; б) если $f(x) = (\text{cl } f)(x)$, то из $x \in \partial f^*(x^*)$ следует $x^* \in \partial f(x)$.

Доказательство. По теореме 6 имеем

$$x^* \in \partial f(x), \text{ если и только если } f^*(x^*) + f(x) = x^*x, \quad (7)$$

$$x \in \partial f^*(x^*), \text{ если и только если } f^{**}(x) + f^*(x^*) = x^*x. \quad (8)$$

Докажем пункт а). Поскольку $\partial f(x)$ непусто, то из теоремы 5 следует $(\text{cl } f)(x) = f(x)$. Тогда по теореме 8 Приложения 1,

$$f^{**}(x) = (\text{cl } f)(x) = f(x), \quad (9)$$

так что (7) переходит в (8) и $x \in \partial f^*(x^*)$. Для доказательства б) отметим, что из $f(x) = (\text{cl } f)(x)$ следует, что выполняется (9), так что (8) переходит в (7) и $x^* \in \partial f(x)$.

Часто используется следствие из теоремы 7.

Следствие. Если $(\text{cl } f)(x) = f(x)$, то $x^* \in \partial f(x)$, если и только если $x \in \partial f^*(x^*)$.

Производные по направлению.

Определение 3. Односторонней производной по направлению s функции f в точке x называется

$$Df(x; s) = \lim_{\lambda \rightarrow 0^+} \frac{f(x + \lambda s) - f(x)}{\lambda}. \quad (10)$$

Определение 4. Функция h , определенная на E^n , называется положительно однородной, если

$$h(\lambda x) = \lambda h(x) \quad \text{для } \forall \lambda > 0 \text{ и } \forall x \in E^n.$$

Теорема 8. Предположим, что $f(x)$ конечна. Тогда разностное отношение

$$d(\lambda) = \frac{f(x + \lambda s) - f(x)}{\lambda} \quad (11)$$

является невозрастающим для всех $\lambda > 0$.

Доказательство. Из факта, что $f(x)$ конечна, следует, что комбинация $(\infty, -\infty)$ не может возникнуть, так что $d(\lambda)$

хорошо определена. Поскольку f выпукла, то одномерная функция

$$g(\lambda) = f(x + \lambda s) \quad (12)$$

выпукла по λ для всех x и s . Пусть $\lambda_2 > \lambda_1 > 0$. Тогда, как видно из рис. П. 2.2,

$$g(0) + \lambda_1 \left[\frac{g(\lambda_2) - g(0)}{\lambda_2} \right] \geq g(\lambda_1), \quad (13)$$

или, используя (12), получаем

$$f(x) + \frac{\lambda_1}{\lambda_2} (f(x + \lambda_2 s) - f(x)) \geq f(x + \lambda_1 s),$$

которое, поскольку $\lambda_1 > 0$, $\lambda_2 > 0$, переходит в

$$\frac{f(x + \lambda_2 s) - f(x)}{\lambda_2} \geq \frac{f(x + \lambda_1 s) - f(x)}{\lambda_1}.$$

Дальнейшее доказывает утверждение, касающееся существования производных по направлению, сделанное ранее.

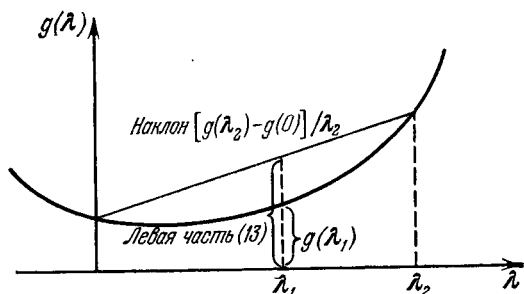


Рис. П. 2.2. Иллюстрация к неравенству (13).

Теорема 9. Пусть x точка, такая, что $f(x)$ конечна. Тогда $Df(x; s)$ существует для всех $s \in E^n$ (возможно, равное $\pm \infty$) и является выпуклой положительно однородной функцией s .

Доказательство. По теореме 8, $d(\lambda)$ является неубывающей для $\lambda > 0$. Таким образом, для любой последовательности $\{\lambda_i\}$ положительных чисел, сходящихся к нулю, последовательность $\{d(\lambda_i)\}$ является невозрастающей. Следовательно, имеется предел (который может быть $\pm \infty$), и этот предел есть $Df(x; s)$.

Для доказательства выпуклости возьмем $\alpha > 0$, $\beta > 0$, $\alpha + \beta = 1$. Тогда

$$\begin{aligned} Df(x; \alpha s_1 + \beta s_2) &= \lim_{\lambda \rightarrow 0+} \frac{f(x + \lambda(\alpha s_1 + \beta s_2)) - f(x)}{\lambda} = \\ &= \lim_{\lambda \rightarrow 0+} \frac{f(\alpha(x + \lambda s_1) + \beta(x + \lambda s_2)) - f(x)}{\lambda}. \end{aligned}$$

Используя выпуклость f , получаем

$$Df(x; \alpha s_1 + \beta s_2) \leq \lim_{\lambda \rightarrow 0^+} \frac{\alpha f(x + \lambda s_1) + \beta f(x + \lambda s_2) - f(x)}{\lambda}.$$

Представляя $f(x)$ в виде $\alpha f(x) + \beta f(x)$, преобразуем далее:

$$Df(x; \alpha s_1 + \beta s_2) = \lim_{\lambda \rightarrow 0^+} \left\{ \frac{\alpha [f(x + \lambda s_1) - f(x)]}{\lambda} + \right. \\ \left. + \frac{\beta [f(x + \lambda s_2) - f(x)]}{\lambda} \right\} = \alpha Df(x; s_1) + \beta Df(x; s_2)$$

Положительная однородность следует из определения

$$Df(x; \alpha s) = \lim_{\lambda \rightarrow 0^+} \frac{f(x + \lambda \alpha s) - f(x)}{\lambda}, \quad (14)$$

где $\alpha > 0$. Пусть $\lambda \alpha = \beta$. Так как $\alpha > 0$, то из $\lambda \rightarrow 0^+$ следует $\beta \rightarrow 0^+$, так что (14) преобразуется к виду

$$Df(x; \alpha s) = \alpha \lim_{\beta \rightarrow 0^+} \frac{f(x + \beta s) - f(x)}{\beta} = \alpha Df(x; s).$$

Теорема 10. Пусть f — собственная. Выберем $x \in \text{dom } f$. Тогда $x^* \in \partial f(x)$, если и только если

$$Df(x; s) \geq x^* s, \text{ для } \forall s \in E^n. \quad (15)$$

Доказательство. По определению, $x^* \in \partial f(x)$, если и только если

$$f(z) \geq f(x) + x^*(z - x), \text{ для } \forall z \in E^n.$$

Выбор $z = x + \lambda s$, $\lambda > 0$ дает

$$\frac{f(x + \lambda s) - f(x)}{\lambda} \geq x^* s,$$

и, считая $\lambda \rightarrow 0^+$, получаем

$$Df(x; s) \geq x^* s, \text{ для } \forall s \in E^n.$$

Следующее утверждение является полезной характеристикой минимизирующих точек выпуклой функции.

Теорема 11. Пусть f — собственная. Тогда x^0 минимизирует f , если и только если

$$Df(x^0; s) \geq 0, \text{ для } \forall s \in E^n.$$

Доказательство. В силу теоремы 4 x^0 минимизирует f , если и только если $0 \in \partial f(x^0)$. В силу теоремы 10 это справедливо, если и только если указанное условие выполняется.

Определение 5. *Функцией-индикатором* выпуклого множества C , $\delta(\cdot|C)$ называется функция вида

$$\delta(x|C) = \begin{cases} 0, & x \in C, \\ +\infty, & x \notin C. \end{cases} \quad (16)$$

Ясно, что $\delta(\cdot|C)$ является выпуклой с эффективной областью C . Она является собственной, если и только если C непусто, и замкнутой, если и только если C замкнуто.

Определение 6. *Опорной функцией* выпуклого множества называется функция $\delta^*(\cdot|C)$, определенная следующим образом:

$$\delta^*(x^*|C) = \sup_{x \in C} (x^*x). \quad (17)$$

Так как

$$\sup_{x \in C} (x^*x) = \sup_{x \in E^n} (x^*x - \delta(x|C)),$$

$\delta^*(\cdot|C)$ есть сопряженная к $\delta(\cdot|C)$. Следовательно, она является выпуклой (теорема 7 Приложения 1), замкнутой и собственной, если и только если C непусто (теорема 9 Приложения 1). Далее, она является положительно однородной, поскольку, если $\lambda > 0$, то

$$\sup_{x \in C} ((\lambda x^*)x) = \lambda \sup_{x \in C} (x^*x).$$

Следующий результат играет важную роль в установлении явного выражения $Df(x; s)$.

Теорема 12. Пусть h — собственная, выпуклая, положительно однородная функция на E^n . Тогда множество

$$C = \{x^*/h(x) \geq x^*x \text{ для } \forall x \in E^n\} \quad (18)$$

является непустым, замкнутым, выпуклым и

$$cl h = \delta^*(\cdot|C). \quad (19)$$

Доказательство. В силу теоремы 8 Приложения 1 $cl h$ является дважды сопряженной к h , т. е. $cl h = f^{**}$, где

$$f^*(x^*) = \sup_x \{x^*x - h(x)\}$$

является, по теоремам 7 и 9 Приложения 1, замкнутой, собственной и выпуклой. Пусть $x = \lambda z$, $\lambda > 0$. Тогда

$$f^*(x^*) = \sup_z (\lambda x^*z - h(\lambda z)).$$

Используя положительную однородность h , мы получаем

$$f^*(x^*) = \lambda \sup_z (x^*z - h(z)) = \lambda f^*(x^*).$$

Таким образом, $f^*(x^*)$ должна быть либо нуль, либо $+\infty$ для любого $x^* \in E^n$. Поскольку она замкнута, выпукла и собственная, то является индикаторной функцией замкнутого, непустого, выпуклого множества. Это множество является множеством всех x^* , таких, что

$$f^*(x^*) = \sup_x (x^*x - h(x)) = 0,$$

т. е. множеством всех x^* , таких, что

$$x^*x - h(x) \leq 0 \text{ для } \forall x \in E^n,$$

т. е. множеством

$$C = \{x^*/h(x) \geq x^*x \text{ для } \forall x \in E^n\}.$$

Таким образом, $f^* = \delta(\cdot | C)$ и $\text{cl } h = \delta^*(\cdot | C)$.

Теоремы 10 и 12 вытекали бы сразу из выражения для $\text{cl}_s Df(x; s)$, если бы условия, гарантирующие, что $Df(x; s)$ удовлетворяет гипотезам теоремы 12, были бы известны. Мы уже знаем, что $Df(x; s)$ является выпуклой и положительно однородной по s . Следующая теорема показывает, что она является собственной.

Теорема 13. Пусть f — собственная. Тогда $\partial f(x)$ является непустым, если и только если $x \in \text{dom } f$ и $Df(x; s)$ правильна по s .

Доказательство. Необходимость. Выберем $x^* \in \partial f(x)$. Тогда

$$f(z) \geq f(x) + x^*(z - x) \text{ для } \forall z \in E^n.$$

Если мы выберем $z \in \text{dom } f$, то $f(x) < \infty$, так что $x \in \text{dom } f$. По теореме 10, $Df(x; s) > -\infty$ для всех $s \in E^n$. Так как $Df(x; 0) = 0$, то $Df(x; s)$ собственная по s .

Достаточность. В силу теоремы 12 множество $C = \{x^*/x^*s \leq Df(x; s)\}$ непусто. В силу теоремы 10 $\partial f(x)$ также непусто.

Далее мы приблизимся к главной цели этого приложения: к выводу выражения для $\text{cl}_s Df(x, s)$.

Теорема 14. Если $\partial f(x)$ непусто, то

$$\text{cl}_s Df(x; s) = \sup \{x^*s/x^* \in \partial f(x)\} = \delta^*(s/\partial f(x)). \quad (20)$$

Доказательство. Так как $\partial f(x) \neq \emptyset$, то $Df(x; s)$ является собственной, выпуклой и положительно однородной по s (теорема 13). Тогда по теореме 12 имеем

$$\text{cl}_s Df(x; s) = \sup \{x^*s/x^* \in C\},$$

где $C = \{x^*/Df(x; s) \geq x^*s \text{ для } \forall s \in E^n\}$. Но в силу теоремы 10, $C = \partial f(x)$.

Последний результат приводит к выражению для $Df(x, s)$ в точках x , таких, что эта производная замкнута по s . Найдем условия, гарантирующие, что это обеспечено.

Теорема 15. Пусть f — собственная. Тогда, если $x \in r \text{ int dom } f$, то $Df(x; s)$ замкнута и собственна по s .

Доказательство. По теореме 2 $\partial f(x) \neq \emptyset$, и по теореме 13 $Df(x; s)$ собственна по s . Для доказательства замкнутости положим $V = \{s/Df(x; s) < \infty\}$, что определяет эффективную область (по отношению к s) для $Df(x; s)$. Выберем s_1 и s_2 в V ; пусть α и β — неотрицательные вещественные числа, и рассмотрим точку

$$\begin{aligned} \alpha s_1 + \beta s_2 &= |\alpha|(\operatorname{sgn} \alpha) s_1 + |\beta|(\operatorname{sgn} \beta) s_2 = \\ &= (|\alpha| + |\beta|) \left(\frac{|\alpha|}{|\alpha| + |\beta|} \tilde{s}_1 + \frac{|\beta|}{|\alpha| + |\beta|} \tilde{s}_2 \right) = \gamma (\tilde{\alpha} \tilde{s}_1 + \tilde{\beta} \tilde{s}_2), \end{aligned}$$

где γ , $\tilde{\alpha}$ и $\tilde{\beta}$ положительные, $\tilde{\alpha} + \tilde{\beta} = 1$. Используем выпуклость и положительную однородность производной по направлению s :

$$\begin{aligned} Df(x; \alpha s_1 + \beta s_2) &= Df(x; \gamma (\tilde{\alpha} \tilde{s}_1 + \tilde{\beta} \tilde{s}_2)) = \gamma Df(x; \tilde{\alpha} \tilde{s}_1 + \tilde{\beta} \tilde{s}_2) \leq \\ &\leq \gamma [\tilde{\alpha} Df(x; \tilde{s}_1) + \tilde{\beta} Df(x; \tilde{s}_2)] = \gamma \tilde{\alpha} Df(x; \tilde{s}_1) + \gamma \tilde{\beta} Df(x; \tilde{s}_2). \quad (21) \end{aligned}$$

Для того чтобы выяснить, что происходит, если $\tilde{s}_1 = -s_1$, заметим, что если $Df(x; s)$ конечно, то $x + \lambda s$ должно быть в $\text{dom } f$ для всех $\lambda > 0$ и достаточно малых. Так как $x \in r \text{ int dom } f$, то $x - \lambda s$ также должно быть в $\text{dom } f$ для $\lambda > 0$ и достаточно малых, так что $Df(x; -s)$ является конечной (не может быть равно $-\infty$, так как является собственной). Следовательно, правая часть (21) конечна и $\alpha s_1 + \beta s_2 \in V$. Так как $0 \in V$, то V есть векторное подпространство E^n . Следовательно, все его точки являются относительно внутренними точками, его относительная граница пуста, и в силу теоремы 5 Приложения 1 имеем

$$\text{cl}_s Df(x; s) = Df(x; s).$$

В силу теорем 14 и 15, если $x \in r \text{ int dom } f$, то

$$Df(x; s) = \sup \{x^* s / x^* \in \partial f(x)\}. \quad (22)$$

По теореме 3, если $\text{dom } f$ имеет непустую внутренность и $x \in \text{int dom } f$, то $\partial f(x)$ компактно. Тогда супремум в (22) достигается в некоторой точке, принадлежащей $\partial f(x)$. Это устанавливает теорему, которая использовалась в главах VII и VIII.

Теорема 16. Пусть $\text{dom } f$ имеет непустую внутренность, и выбрано $x \in \text{int dom } f$. Тогда

$$Df(x; s) = \max \{x^* s / x^* \in \partial f(x)\}. \quad (23)$$

Этот результат может быть обоснован геометрическим путем из рассмотрения рис. П. 2.1. Пусть s исходит из x направо. Тогда $Df(x; s)$ является правосторонней производной f в точке x . Она равна наклону опорной линии, проходящей через x , которая по-

ворачивается против часовой стрелки настолько, насколько это возможно, т. е. $\max\{x^*/x^* \in \partial f(x)\}$. Аналогично $Df(x; s)$ есть $-\min\{x^*/x^* \in \partial f(s)\}$, как предсказывалось в (23).

Дальнейшее рассмотрение рис. П. 2.1 показывает, что $\partial f(x)$ состоит из единственной точки, если и только если f дифференцируема по x ; причем в этом случае $Df(x; s) = \nabla f(x)'s$ линейна по s . Справедливость этого доказана ниже.

Теорема 17. Пусть f — собственная. Тогда эквивалентны следующие утверждения: а) $x \in \text{int dom } f$ и f дифференцируема по x , б) $Df(x; s)$ линейна по s , в) $\partial f(x)$ содержит единственную точку x^* .

Доказательство. (а) $\Rightarrow$ (б); это следует немедленно, поскольку, при а) $Df(x; s) = \nabla f'(x)s$.

(б) $\Rightarrow$ (в): Пусть

$$Df(x; s) = a's \quad \text{для} \quad \forall s \in E^n. \quad (24)$$

Так как, в силу (22) $Df(x; s) = \sup\{x^*s/x^* \in \partial f(x)\}$, если $\partial f(x)$ содержит больше, чем одну точку, скажем x_1^* и x_2^* , то существуют векторы s_1 и s_2 , такие, что супремум для s_1 достигается в x_1^* и по s_2 в x_2^* . Тогда (24) будет нарушено.

(в) $\Rightarrow$ (а): В силу (в), имеем

$$Df(x; s) = x^*s. \quad (25)$$

Если $x \notin \text{int dom } f$, то существует $\tilde{s}$ такое, что $x + \lambda\tilde{s} \notin \text{dom } f$ для всех $\lambda > 0$. Это приводит к $Df(x; \tilde{s}) = +\infty$, что противоречит (25). Таким образом, $x \in \text{int dom } f$, и доказательство сводится к классическому случаю, рассмотренному в [7, стр. 85].

ЛИТЕРАТУРА

1. R. T. Rockafellar, Convex Functions and Dual Extremum Problems. Ph. D. Thesis, Harvard University, 1963.
2. R. T. Rockafellar, Nonlinear Programming. American Mathematical Society Summer Seminar on the Mathematics of the Decision Sciences, Stanford University, July—Aug. 1967.
3. M. Hamala, Geometric Programming in Terms of Conjugate Functions. Discussion Paper 6811, Center for Operations Research and Econometrics, Catholic University of Louvain, Nevele, Belgium, 1968.
4. H. G. Eggleston, Convexity, Cambridge University Press, New York, 1958.
5. S. Karlin, Mathematical Methods and Theory in Games, Programming and Economics, vol. 1, Addison—Wesley Publishing Company, Inc., Reading, Mass., 1959. [Русском переводе: С. Карлин, Математические методы в теории игр, программировании и экономике, «Мир», М., 1964.]
6. F. A. Valentine, Convex Sets, McGraw-Hill, Inc., New York, 1964.
7. W. Fenchel, Convex Cones, Sets, and Functions, lecture notes, Princeton University, 1951.
8. R. T. Rockafellar, Convex Analysis, Princ., N.-J., Princ. Univ. Press, 1970 [Русск. перев.: Р. Рокфеллар, Выпуклый анализ, «Мир», 1973.]

ДОПОЛНИТЕЛЬНЫЙ СПИСОК ЛИТЕРАТУРЫ

Библиография к книге Л. Лэсона охватывает период до 1970 года. К этому моменту основная «волна» теоретических исследований по вычислительным методам решения «больших» задач математического программирования уже прошла, и ссылки на работы американских авторов, приведенные в книге, можно считать достаточно полными и к настоящему времени. Однако исследования, проводимые в СССР, фактически совершенно не упомянуты, хотя они шли параллельно с американцами, и, в некоторых случаях, их опережали. Приводимый ниже список также не претендует на полноту. В частности, в нем почти не отражены работы, носящие экономический характер, где вычислительные алгоритмы интерпретируются (обычно, нестрого) как возможные механизмы децентрализованного планирования.

1. Байбородин Н. Е., Макаров А. А., Один алгоритм блочного линейного программирования, Сб. «Методы математического моделирования в энергетике», Вост.-Сиб. книж. изд., 1966.
2. Беленький В. З., Волконский В. А. (ред.), Итеративные методы в теории игр и программировании, «Наука», М., 1974.
3. Вахутинский И. Я., Дудкин Л. М., Щенников Б. А., Итеративное агрегирование в некоторых оптимальных экономических моделях, ЭММ¹⁾, 1973, т. 9, вып. 3.
4. Волконский В. А., Оптимальное планирование в условиях большой размерности (итеративные методы и принцип декомпозиции), ЭММ, 1965, т. 1, вып. 2.
5. Волконский В. А., Модель оптимального планирования и взаимосвязи экономических показателей, «Наука», М., 1967.
6. Вересков А. И., Использование овражного спуска в блочном программировании, ЭММ, 1973, т. 9, вып. 3.
7. Гольштейн Е. Г., Методы блочного программирования, ЭММ, 1966, т. 2, вып. 1.
8. Гольштейн Е. Г., Обобщенный градиентный метод отыскания седловых точек, ЭММ, 1972, т. 8, вып. 4.
9. Гольштейн Е. Г., Третьяков Н. В., Модифицированные функции Лагранжа, ЭММ, 1974, т. 10, вып. 3.
10. Гольштейн Е. Г., Юдин Д. Б., Новые направления в линейном программировании, «Сов. радио», М., 1966.
11. Грибов А. Б., Решение больших задач линейного программирования, Труды 2-й зимней школы по математическому программированию и смежным вопросам, Дрогобыч, 1969, вып. 1.
12. Звягина Р. А., Задачи линейного программирования с блочно-диагональными матрицами, Сб. «Оптимальное планирование», СО АН СССР, 1964, вып. 2 (алгоритмы).
13. Звягина Р. А., Задачи линейного программирования с матрицами произвольной блочной структуры, ДАН СССР, 1971, т. 196, № 4.

¹⁾ Здесь и далее ЭММ означает «Экономика и математические методы».

14. Звягина Р. А., Об общем методе решения задач линейного программирования блочной структуры, Сб. «Оптимизация», 1 (18), Новосибирск, 1971.
15. Коробочкин Б. И., Обобщенные функции Лагранжа и некоторые их свойства, Сб. «Оптимизация», 10 (27), Новосибирск, 1973.
16. Кутиков Л. М., Декомпозиция блочных задач линейного программирования со слабо связанными блоками, ЭММ, 1973, т. 9, вып. 4.
17. Малков У. Х., Ахметов П. А., Использование специфики условий задачи в мультипликативном алгоритме, ЭММ, 1972, т. 8, вып. 1.
18. Малинников В. В., Метод разложения в решении больших задач линейного программирования с блочной структурой, ЭММ, 1971, т. 7, вып. 5.
19. Мовшович С. М., Метод невязок для решения задач блочной структуры, ЭММ, 1966, т. 2, вып. 4.
20. Первозванская Т. Н., Первозванский А. А., Алгоритм поиска оптимального распределения централизованных ресурсов, Изв. АН СССР, Техническая кибернетика, 1966, № 3.
21. Первозванская Т. Н., Первозванский А. А., Распределение централизованных ресурсов между многими предприятиями, ЭММ, 1966, т. 2, вып. 5.
22. Первозванская Т. Н., Метод приближения в пространстве целевых функций при решении задач «на узкие места», Журнал вычислительной математики и математической физики, 1967, № 3.
23. Первозванская Т. Н., Первозванский А. А., Децентрализация оптимального планирования в сложной системе. Автоматика и телемеханика, 1968, № 7.
24. Полтерович В. М., Блочные методы вогнутого программирования и их экономическая интерпретация, ЭММ, 1969, т. 5, вып. 6.
25. Полтерович В. М., Математические модели перераспределения ресурсов, ЦЭМИ АН СССР, М., 1970.
26. Полтерович В. М., Об одной модели перераспределения ресурсов, ЭММ, 1970, т. 6, вып. 4.
27. Поляк Б. Т., Третьяков Н. В., Об одном итерационном методе линейного программирования и его экономической интерпретации, ЭММ, 1972, т. 8, вып. 5.
28. Поляк Б. Т., Третьяков Н. В., Метод штрафных оценок для задач на условный экстремум, ЖВМ и МФ, 1973, т. 13, № 1.
29. Сб. «Проблемы функционирования больших экономических систем», «Наука», М., 1969.
30. Рубинштейн Г. Ш., О решении задач линейного программирования большого объема, Сб. «Оптимальное планирование», СО АН СССР, 1964, вып. 2 (алгоритмы).
31. Третьяков Н. В., Метод штрафных оценок для задач выпуклого программирования, ЭММ, 1973, вып. 3.
32. Шор Н. З., Применение обобщенного градиентного спуска в блочном программировании, Кибернетика, 1967, № 3.

Л. С. Лэсдон
ОПТИМИЗАЦИЯ БОЛЬШИХ СИСТЕМ

М., 1975 г., 432 стр. с илл.

Редактор **Д. С. Фурманов**
Техн. редактор **В. Д. Элькинд**
Корректоры **Е. А. Белицкая, Н. Б. Румянцева**

Сдано в набор 11/IV 1975 г.
Подписано к печати 12/VIII 1975 г.
Бумага 60×90^{1/16}. Физ. печ. л. 27.
Условн. печ. лт. 27. Уч.-изд. л. 24,83.
Тираж 11 000 экз. Цена книги 2 р.
Заказ № 4518

Издательство «Наука»
Главная редакция
физико-математической литературы
117071, Москва, В-71, Ленинский проспект, 15

2 типография издательства «Наука».
Москва, Шубинский пер., д. 10.