



М.В. МЕЕРОВ
Б.А. ЛИТВАК

ОПТИМИЗАЦИЯ
СИСТЕМ
МНОГОСВЯЗНОГО
УПРАВЛЕНИЯ





ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ТЕХНИЧЕСКОЙ КИБЕРНЕТИКИ

ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1972

**М. В. МЕЕРОВ,
Б. Л. ЛИТВАК**

ОПТИМИЗАЦИЯ СИСТЕМ МНОГОСВЯЗНОГО УПРАВЛЕНИЯ

**ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1972**

Оптимизация систем многосвязного управления. Мееров М. В., Литвак Б. Л., Главная редакция физико-математической литературы изд-ва «Наука», М., 1972, 344 стр.

Книга посвящена проблеме оптимизации сложных многосвязных систем.

Устанавливаются основные свойства и особенности объектов многосвязного управления, качество которых определяется функционалом регулируемых величин. Рассмотрены типовые объекты изучаемого класса, для которых формируются обобщенные показатели качества и ограничения. Предложен специально разработанный метод и алгоритмы программирования, позволяющие на существующих вычислительных машинах успешно решать задачи оптимизации большой размерности. Рассматриваются технические, производственные и экономические объекты.

Книга рассчитана на инженеров и научных работников, занятых проектированием систем управления.

Илл. 36. Библ. 101 назв.

*Михаил Владимирович Мееров,
Борис Львович Литвак*

ОПТИМИЗАЦИЯ СИСТЕМ МНОГОСВЯЗНОГО УПРАВЛЕНИЯ

М., 1972., 344 стр. с илл.

Редактор Д. С. Фурманов

Техн. редактор Е. Н. Земская

Корректор Т. А. Панькова

Сдано в набор 27/VII 1972 г. Подписано к печати 17/XI 1972 г. Бумага 84×
×108¹/₃₂. Физ. печ. л. 10.75. Условн печ. л. 18.06. Уч.-изд. л. 17.44.
Тираж 6 000 экз. Т-16881. Цена книги 1 р. 34 к.
Заказ № 1450

Издательство «Наука»

Главная редакция физико-математической литературы
117071, Москва, В-71, Ленинский проспект, 15

Ордена Трудового Красного Знамени Первая Образцовая типография имени
А. А. Жданова Главполиграфпрома Государственного комитета Совета Министров СССР по делам издательств, полиграфии и книжной торговли.
Москва, М-54, Воровая, 28

Отпечатано во 2-ой типографии издательства «Наука».
Москва Г-99, Шубинский пер., 10.

ОГЛАВЛЕНИЕ

Предисловие	7
Введение	9
Г л а в а I. Объекты многосвязного управления. Примеры линейных задач оптимизации	20
§ 1. Нефтедобывающее предприятие (НДП) как объект оптимального управления	21
§ 2. Структурные представления многосвязных объектов. S -матрицы	39
§ 3. Объекты, описываемые некоторыми дифференциальными уравнениями в частных производных (уравнениями Лапласа, Пуассона и другими)	47
§ 4. Регулирование мощности в системе радиостанций	57
§ 5. Межотраслевой баланс	60
§ 6. Электрические цепи	69
Г л а в а II. Примеры нелинейных и некоторых динамических задач оптимизации	77
§ 1. Некоторые нелинейные задачи оптимизации многосвязных систем и S^k -задачи	77
§ 2. Марковское программирование	80
§ 3. Постановки задач оптимизации марковских процессов с доходами	87
§ 4. Задачи оптимизации марковских процессов с доходами при наличии переоценки	90
§ 5. Примеры задач оптимизации марковских процессов с доходами	96
§ 6. Заключительные замечания	102
Г л а в а III. Алгоритмы решения линейных задач оптимизации многосвязных систем	104
§ 1. Свойства S -задачи	106
§ 2. Алгоритмы решения S -задач. Примеры	120
§ 3. Различные приложения свойств S -задач. Некоторые замечания	138
§ 4. Приближенное решение	146

§ 5. Оптимальное управление режимом нефтяного пласта. Две постановки задачи	148
§ 6. Решение задачи оптимизации нефтедобычи по матрице коэффициентов взаимовлияния скважин по дебиту	156
§ 7. Решение задачи оптимизации нефтедобычи по матрице коэффициентов взаимовлияния скважин по депрессиям	172
§ 8. Симметричные задачи линейного программирования	180
§ 9. Некоторые обобщения	183
§ 10. Анализ устойчивости	193
§ 11. Обсуждение	197
§ 12. Некоторые соображения по динамической оптимизации многосвязных систем	205
Г л а в а IV. Алгоритмы решения S^k-задач	213
§ 1. Общие свойства S^k -задач	213
§ 2. Дополнительные свойства S^k -задач, возникающих при статической оптимизации нелинейных многосвязных систем	218
§ 3. Свойства S^k -задач, возникающих в марковском программировании	226
§ 4. Алгоритмы решения S^k -задач. I	229
§ 5. Алгоритмы решения S^k -задач. II	249
Г л а в а V. Решение некоторых дискретных задач	262
§ 1. Решение S -задач со снятием ограничений	262
§ 2. Решение частично целочисленных S -задач	267
§ 3. Пример частично целочисленной S -задачи	270
Г л а в а VI. Решение линейных задач оптимизации на электрических моделирующих устройствах	279
§ 1. Вступительные замечания	279
§ 2. Метод решения I	280
§ 3. Примеры решения задач по методу I	288
§ 4. Метод решения II	298
§ 5. Примеры решения задач по методу II	305
§ 6. Сравнение методов I и II	307
П р и л о ж е н и е. Некоторые сведения из линейного программирования	310
§ 1. Некоторые определения	310
§ 2. Теоретические основы формирования методов линейного программирования	315
§ 3. Вычислительные схемы	323
Л и т е р а т у р а	339

ПРЕДИСЛОВИЕ

В современной теории автоматического управления и приложениях многосвязные системы занимают особое положение. С одной стороны, они корнями глубоко уходят в традиционные проблемы управления и регулирования с четко очерченной проблематикой. Естественно, что при этом наличие большого числа связанных между собой управляемых и управляющих величин усложняет эти проблемы и ставит новые. Примером может служить известная проблема автономности. С другой стороны, проблематика, возникающая в связи с оптимизацией многосвязных систем, теснейшим образом связана с современными проблемами управления большими системами.

Такое положение обусловлено наличием двух типов многосвязных объектов. Как известно [56], к многосвязным относятся такие объекты, в которых количество управляемых и управляющих величин больше одной. Фактически число этих величин может быть достаточно большим, и характер связи между ними обусловлен свойствами объекта, управления и нагрузки. Из общего класса многосвязных систем можно выделить подкласс, для которого автономность не противоречит условиям нормального функционирования всей системы. Для этого подкласса обеспечение автономности для отдельных подсистем сводит решение задачи управления системой с n связными величинами к управлению n системами с одной управляемой величиной каждая. Теория автономности достаточно хорошо разработана; хотя доказано, что автономизация с последующей оптимизацией каждой автономной системы не всегда обеспечивает оптимум всей системы, все же такой путь решения задачи управления в ряде случаев является приемлемым.

Другой, наиболее важный и многочисленный подкласс многосвязных систем обладает тем свойством, что автономность противоречит условиям нормального функционирования системы или вообще не имеет смысла. Краеугольным здесь является то, что проблема оптимизации формулируется в виде требования экстремизировать функционал, зависящий одновременно от всех управляемых величин, при условии, что все величины и управления связаны между собой.

Настоящая работа посвящена разработке эффективных методов решения задач оптимизации многосвязных объектов второго типа.

Книга состоит из шести глав. Первые две главы посвящены описанию некоторых типовых объектов. Здесь основное внимание уделяется математической формулировке задачи оптимизации. Много внимания уделяется выяснению особенностей объектов многосвязного управления. В последующем эти особенности используются при разработке алгоритмов решения задачи. Главы III—VI посвящены разработке алгоритмов решения задач оптимизации.

Результаты теории и методы решения широко иллюстрируются практическими примерами из различных областей техники и экономики. Необходимые для понимания книги сведения из линейного программирования приведены в приложении.

Часть результатов, приведенных в книге, была опубликована в статьях авторов (49—52, 58—60, 100). Значительная часть результатов публикуется впервые. Обозначения матриц даются в книге полужирным шрифтом, векторы обозначаются, как правило, прописными (заглавными) буквами.

Авторы пользуются приятной возможностью выразить благодарность проф. А. А. Первозванскому, взявшему на себя труд по рецензированию книги, замечания и советы которого способствовали ее улучшению.

Авторы будут признательны всем, кто выскажет пожелания и замечания по работе.

М. Мееров, Б. Литвак

Апрель, 1971

ВВЕДЕНИЕ

Характерной особенностью современных технологических и производственных процессов как объектов управления является наличие в них большого числа управляемых и управляющих величин, которые влияют друг на друга в том смысле, что изменение какой-либо одной из них вызывает изменение всех других. Это обстоятельство требует специального подхода к решению задач управления такими объектами.

Здесь правомерны два подхода. Первый состоит в автономном управлении отдельными регулируемыми величинами (если это признано допустимым). Речь в данном случае идет о таких объектах, где взаимовлияние между регулируемыми величинами обусловлено физическими свойствами объекта, но не связано с результатами его функционирования. Для ясности приведем пример, ставший в настоящее время классическим примером многосвязного объекта данного класса: паровой котел, в котором регулируются температура, давление пара и уровень воды. Эти три переменные взаимосвязаны. Характер связи здесь подчинен определенному физическому закону. Однако, если ввести ограничения на пределы изменения этих величин, исходя из требований нормальной эксплуатации, то каждую из них можно регулировать отдельно. Для реализации такого регулирования необходимо, чтобы система управления обеспечивала бы «развязывание» или автономизацию регулируемых величин. Это может быть достигнуто или введением искусственных связей между регуляторами по различным регулируемым величинам так, чтобы скомпенсировать естественную связь между ними, или выбором такой структуры, для которой автономность является структурным свойством.

Объектам такого типа и системам управления, которые обеспечивают автономность, в литературе уделено достаточ-

но много внимания. В случае, когда автономность достигнута, задача сводится к обычным задачам автоматического регулирования для объектов с одной регулируемой величиной с использованием хорошо известных методов и средств. Вместо решения задачи управления многосвязным объектом с n регулируемыми величинами в этом случае решается задача управления n объектами с одной регулируемой величиной у каждого. Мы здесь не будем касаться вопроса о том, оптимально ли такое решение. В свое время нами было показано [56], что такое решение не является оптимальным с точки зрения обычных требований динамики управления. Важно, однако, то, что для этого подкласса обычно искусственное устранение влияния между регулируемыми величинами не противоречит технологическому процессу, реализуемому на данном объекте. Анализ этих систем не входит в задачу настоящей работы.

Второй подход состоит в одновременном учете влияния всех регулируемых величин и их взаимной связи и выяснении условий оптимальности функционирования объекта. Вообще говоря, такой подход правомерен и для рассмотренного ранее подкласса объектов. Однако класс объектов, которому будет уделено основное внимание в настоящей работе, именно потому и выделяется в отдельный класс, что с точки зрения технологии устранения связи между отдельными регулируемыми величинами недопустимо или лишено смысла. Конкретные объекты данного класса, их математическое описание и постановка задачи оптимизации будут подробно рассмотрены в главах I и II. Здесь мы пока рассмотрим только некоторые принципиальные вопросы, относящиеся к рассматриваемой проблеме в целом.

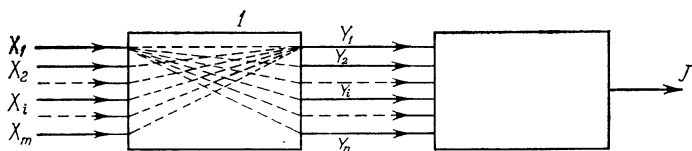


Рис. В.1. К определению многосвязных систем.

Многосвязный объект рассматриваемого типа может быть представлен схемой рис. В.1. Пусть вектор входных величин обозначен

$$X = [x_1, x_2, \dots, x_m]$$

и вектор выходных величин —

$$Y = [y_1, y_2, \dots, y_n];$$

при этом в общем случае $n \neq m$. Обобщенный показатель качества J функционирования объекта зависит одновременно от всех выходных величин Y . Так как в общем случае сами y_j являются функциями времени, то

$$J = \varphi [Y(t)]. \quad (B.1)$$

Характер изменения вектора Y зависит от закона изменения вектора входных величин $X(t)$ и от свойств части объекта, обозначенной на рис. В.1 значком 1. Без ограничения общности можно написать следующее выражение, определяющее поведение объекта:

$$F(X, Y, \psi[X, Y], t) = 0. \quad (B.2)$$

Здесь $\psi[X, Y]$ определяет свойства объекта. Кроме того, возможны случаи, когда переменные $X(t)$ и $Y(t)$ не должны выходить за некоторые заранее установленные пределы, так что дополнительно к (B.2) могут существовать еще условия типа

$$\text{и} \quad \left. \begin{array}{l} \bar{a} \leq X \leq \bar{b} \\ \bar{d} \leq Y \leq \bar{e}. \end{array} \right\} \quad (B.3)$$

Задача оптимизации может быть сформулирована следующим образом: экстремизировать ψ функционал (B.1) при соблюдении условий (ограничений) (B.2) и (B.3).

Мы получили вариационную задачу самого общего вида. Возможные пути решения существенно зависят от характера функций φ и F . Однако уже вид ограничений, представленный соотношениями (B.3), показывает, что здесь мы имеем дело с неклассической вариационной задачей. Далее, если входы — это управляющие величины, а выходы — управляемые, то здесь мы имеем ограничения как на управляемые, так и на некоторые фазовые координаты.

Здесь уместно подчеркнуть одну особенность систем многосвязного регулирования. Выходами многосвязного объекта являются регулируемые (или управляемые) величины и ограничения на эти величины обусловлены требованиями к процессу. Количество регулируемых величин в общем случае не равно количеству обобщенных координат объекта. Как следует из (B.1), обобщенный показатель качества

непосредственно зависит от тех обобщенных координат, которые являются выходами объекта, а не от всей совокупности обобщенных координат, которых может быть значительно больше, чем выходных величин.

Учитывая свойства объекта и постановку задачи оптимизации, легко понять, что решения лежат на пути использования методов математического программирования. Действительно, предположим, что в частном случае функционал (В.1) имеет вид *)

$$J = \int_0^t [Cf(\Theta)] d\Theta, \quad (\text{В.4})$$

и ограничения выражаются в форме

$$\int_0^t \varepsilon(t - \Theta) f(\Theta) d\Theta = \rho(t), \quad (\text{В.5})$$

$$f(\Theta) \geq 0; \quad t \in (0, T). \quad (\text{В.6})$$

Здесь C — постоянный вектор, $\rho(t)$ — заданная вектор-функция, $f(\Theta)$ — искомая вектор-функция. (Сформулированная задача относится, например, к минимизации себестоимости при заданном плане добычи нефти или к максимизации нефтедобычи при заданной себестоимости при учете технологических ограничений и при условии, что нефтяной пласт эксплуатируется в упругом режиме.) Задача (В.4) — (В.6) суть задача линейного программирования в функциональном пространстве.

Если (В.1) выражается линейной формой

$$J = \sum_{j=1}^n c_j y_j, \quad (\text{В.7})$$

а ограничения имеют, например, вид

$$\sum_{j=1}^n a_{ij} y_j \leq b_i, \quad i = 1, \dots, m, \quad (\text{В.8})$$

$$y_j \geq 0, \quad (\text{В.9})$$

то проблема оптимизации сводится к решению обычной задачи линейного программирования. (Такой случай имеет место, например, при оптимизации нефтедобычи в жестком

*) Подробно задача, которая приводится к (В.4)—(В.6), будет рассмотрена в главе I.

режиме работы пласта и некоторых дополнительных предположениях.) Если функционал содержит нелинейные элементы или нелинейности определенного типа содержатся в ограничениях, то при некоторых условиях задача оптимизации сводится к решению задач выпуклого программирования. (Такой случай имеет место, например, при учете нелинейности характеристик глубинных насосов при механизированной нефтедобыче или при учете нелинейностей при описании фонтанирующих скважин.) Ряд задач оптимизации рассматриваемого класса многосвязных объектов сводится к решению задач параметрического и целочисленного программирования.

Обратим внимание на следующее весьма важное обстоятельство. Свойства многосвязных объектов, как они были сформулированы выше, и постановка задачи оптимизации не связаны с какими-либо определенными физическими или техническими особенностями объектов. Объектом может быть и производственный процесс и экономическая система. В качестве функционала может быть как технологический показатель, например, качество выпускаемой продукции, так и экономический показатель, например, себестоимость. При этом в первом случае среди ограничений могут быть экономические ограничения, в то время как во втором в качестве ограничений могут быть технологические или производственные и т. д. Отсюда ясно, что класс рассматриваемых объектов весьма широк.

До сих пор речь шла об объектах, поведение которых определено; такие объекты называют *детерминированными*, при этом, что особенно существенно, мы предполагаем, что математическое описание объекта уже дано или может быть каким-либо образом получено. Большое число технологических объектов, принадлежащих к классу многосвязных, или не описывается математически строго из-за сложности явлений, или не может быть описано из-за отсутствия соответствующей измерительной техники. Т. о. речь идет об объектах, не относящихся к классу детерминированных.

Один из путей решения проблемы управления для указанных типов объектов связан с выделением конечного числа переменных, которые глобально определяют свойства объекта. По самой своей природе это будут взаимосвязанные величины и изменение какой-либо из них приводит к изменению других. В качестве примеров здесь могут служить

классические примеры статистической физики. Однако нахождение этих обобщенных переменных само по себе является задачей огромной трудности. Мы коснулись здесь этой проблемы только для того, чтобы подчеркнуть, что к многосвязным относится и этот большой и, вероятно, важнейший класс объектов.

Многосвязная система может трактоваться как коллективное поведение большого числа отдельных элементов,

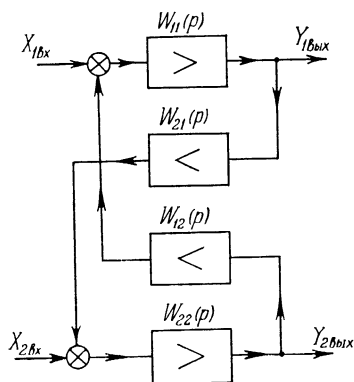


Рис. В.2. Система из двух регулируемых величин.

каждый из которых имеет свою цель, но реализация этой цели может быть достигнута только при условии учета влияния всех других элементов. Действительно, на рис. В.2 представлена структура системы регулирования, состоящая из двух регулируемых величин. По каждой регулируемой величине имеется свой замкнутый контур, задача которого состоит в том, чтобы изменить свою выходную величину в соответствии с изменением своего задания. Однако, как это видно из рис. В.2,

между отдельными контурами существует связь (в данном случае это будет перекрестная обратная связь), которая оказывает влияние на поведение регулируемой величины, в общем случае отличное от действия собственного задания. В системе с большим числом таких контуров каждый стремится реализовать свой собственный закон изменения, но при этом на каждый из контуров оказывают влияние переменные других контуров. Система может выполнить требуемые функции только при условии, что будет учтено не только влияние задания на каждом из контуров, но и взаимовлияние всех несобственных регулируемых переменных; при этом вся система в целом должна удовлетворять определенному критерию. Полученная ситуация характерна для коллективного поведения большого числа самостоятельных элементов.

Вся проблематика, возникающая в теории коллективного поведения большого числа элементов, в полной мере воз-

никает при рассмотрении многосвязной системы рассматриваемого класса. В частности, весьма интересным и важным для многосвязной системы является рассмотрение ее в конфликтной ситуации и рассмотрение систем с иерархией по обобщенным показателям качества (рис. В.3).

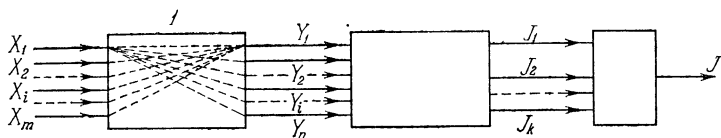


Рис. В.3. К определению сложных многосвязных систем.

Итак, мы охарактеризовали некоторые задачи и проблемы, с которыми приходится сталкиваться при оптимизации многосвязных систем. Приведенные рассуждения указывают, где следует искать пути и методы решения задачи. Такими путями и методами являются математическое программирование и теория игр. Однако успеха здесь можно добиться только при условии учета особенностей рассматриваемого класса систем и объектов. Это обстоятельство, которое является, с нашей точки зрения, фундаментальным, и положено в основу всей книги, заслуживает специального рассмотрения во введении с тем, чтобы была ясна основная идея книги.

Сложность является органической особенностью многосвязных систем. Методы исследования и оптимизации, хорошо разработанные в теории управления, оказываются беспомощными при сколько-нибудь сложной системе из-за того, что называют «проклятием размерности» [4]. Для многосвязных систем такая ситуация является не исключением, а правилом. По этой причине основной целью настоящей работы явилось именно преодоление трудностей, связанных с размерностью задач, не жертвуя при этом строгостью решения.

Мы уже выше указывали, что успеха можно добиться, если учесть особенности рассматриваемого класса систем. Что же является наиболее существенным для рассматриваемого класса объектов и систем? Ранее мы подчеркнули, что математическим аппаратом, адекватным данным задачам оптимизации, является аппарат математического программи-

рования. Большое число взаимосвязанных переменных приводит к тому, что матрица ограничений в задаче (B.1) — (B.3) может быть полностью заполненной и имеет высокую размерность. В качестве иллюстрации укажем, что нефтяной пласт может иметь до 1000 скважин. Если учесть производственные и плановые ограничения в задаче (B.7) — (B.9), то размерность матрицы ограничений может достичь 2000×1000 . Решать такую задачу обычным симплекс-методом практически невозможно даже на самых современных машинах. Здесь дело не только в том, что при решении придется многократно обращаться к внешней памяти машины, что также весьма существенно. Дело в том, что при решении задач такой размерности накапливаемая ошибка может свести к нулю само решение. Поэтому важнейшим является изыскание новых путей, новых методов решения задач большой размерности *). Многосвязные объекты обладают, вообще говоря, естественным и при этом весьма примечательным свойством, которое можно сформулировать следующим образом: самовлияние в данном контуре или канале обычно больше взаимовлияний на данный контур других контуров или каналов (матрица с диагональным преобладанием). Это фундаментальное свойство приводит к тому, что матрица, описывающая многосвязный объект, как правило, сводится к виду, при котором каждая строка и каждый столбец ее содержат не более одного положительного элемента, а остальные элементы неположительные. Существует также весьма широкий класс многосвязных объектов, которые в исходном виде описываются матрицей с указанной спецификой знаков. Это многосвязные объекты с прямыми отрицательными или с обратными положительными перекрестными связями. В дальнейшем мы рассмотрим большое число подобных объектов.

Для матриц ограничений указанного вида или матриц, приводящихся к ним, удалось найти алгоритмы, позволяющие решать задачи линейного программирования такой размерности, которая вполне достаточна при оптимизации реальных объектов. Удалось также решить некоторые задачи

*) Мы бы хотели подчеркнуть, что, с нашей точки зрения, поиск алгоритмов для решения задач большой размерности не связан с ограниченными возможностями машин данного поколения. Хорошо разработанные и удобные алгоритмы будут необходимы в любых, даже самых благоприятных условиях и совершенных вычислительных машинах.

выпуклого программирования и некоторые многоэкстремальные задачи на невыпуклых разрывных областях.

Таким образом, можно сказать, что задачи оптимизации многосвязных систем (и другие задачи, обладающие той же спецификой) выделяются в специальный класс задач математического программирования.

В настоящее время среди всей совокупности задач математического программирования можно (и должно) выделить несколько специальных классов задач, эффективно решаемых в условиях большой размерности. К ним относятся задачи транспортного типа *), задачи с блочно-диагональными матрицами и т. п. Специфика задач транспортного типа опирается на особенности значений элементов матрицы. Известны и другие классы задач, использующие специфику значений элементов матрицы (например, эти значения могут быть равны 0 или ± 1). Специфика задач с блочно-диагональными матрицами связана с неполным заполнением матрицы ненулевыми элементами.

Таким образом, в настоящее время известны специальные классы задач математического программирования, использующие *специфику значений элементов* матрицы ограничений, или *специфику заполнения матрицы* ограничений ненулевыми элементами.

А что же делать, если матрица имеет произвольные значения элементов и произвольное заполнение? Для рассматриваемого в книге класса задач оптимизации предлагается использовать *специальную знаковую структуру элементов* матрицы.

Подчеркивая этот факт, мы назовем *S-матрицей* матрицу, в которой каждый столбец и каждая строка содержат не более одного положительного элемента, а остальные элементы неположительные **) (*S* — по первой букве слова

*) О важности выделения специальных классов задач оптимизации говорит хотя бы такой часто приводимый факт, что из всех решенных в США задач линейного программирования 85% приходилось на задачи именно специального вида (задачи транспортного типа).

**) Ранее, в работах [49, 50, 52, 58, 59, 60] мы называли *S-матрицу* матрицей требуемого вида (ТВ-матрицей), а *S^k-матрицу* — матрицей квазипотребуемого вида (КТВ-матрицей). Заметим, что матрицу, подобную *S-матрице*, в том случае, когда она является квадратной, называют иногда *матрицей Метцлера* (или *М-матрицей*). Если же подобная матрица $A = [a_{ij}]$, кроме того, что является квадратной, обладает еще

sign — знак). В книге используется также матрица, состоящая из серии S -матриц, расположенных друг за другом по горизонтали или по вертикали. Эту матрицу мы назовем S^k -матрицей, где k — количество S -матриц. (Матрицу S^1 мы для простоты обозначили через S .) В матрице S^k каждая строка содержит не более k положительных элементов (если S -матрицы расположены по горизонтали), а каждый столбец содержит не более одного положительного элемента, остальные элементы неположительные.

Целесообразность выделения того или иного специального класса задач определяется двумя показателями: 1) широта класса задач, обладающих данной спецификой, и важность этих задач; 2) эффект от использования рассматриваемой специфики.

Эти вопросы подробно рассматриваются в книге. Здесь мы скажем только несколько слов.

Во-первых, в отношении круга объектов, обладающих указанными свойствами. Не прибегая к «опасному» слову *все*, мы отметим, что огромное большинство многосвязных объектов с сосредоточенными параметрами обладает этим свойством. Матрицами ограничений такого типа обладают объекты с распределенными параметрами, описываемые уравнениями Лапласа и типа Лапласа, Пуассона и некоторые другие объекты, описываемые дифференциальными уравнениями в частных производных эллиптического типа. Такими свойствами обладает широкий класс электрических цепей. Вообще, можно сказать, что к данному классу задач оптимизации относятся задачи определения оптимальных интенсивностей источников и стоков в некоторых сетях, частным случаем которых являются электрические цепи.

Весьма интересно также, что к данному типу задач относится такая классическая задача экономики, как задача о межотраслевом балансе — простая модель Леонтьева, модели обмена, задача об оптимальном энергетическом режиме n пар приемно-передающих радиоустройств и, конечно, задача об оптимальной работе нефтяного пласта в жестком режиме. Для указанных объектов решается задача статической оптимизации.

свойствами $a_{ii} > 0$, $a_{ij} \leq 0$ ($i \neq j$), $\sum_i a_{ij} > 0$ для всех j , то она называется *матрицей Минковского — Леонтьева*, или *матрицей Адамара*. Иногда M -матрицей называют матрицу Адамара.

Подобной структурой матриц ограничений обладают задачи динамической оптимизации систем, поведение которых описывается управляемым марковским процессом. Благодаря аналогии в структуре матриц эти задачи непосредственно включаются в рассматриваемый класс и могут быть использованы при динамической оптимизации соответствующих многосвязных систем. Класс этих объектов весьма широк: это многие динамические задачи, где нужно определить стратегию поведения в условиях неопределенности, некоторые задачи массового обслуживания и т. д. Подобного рода задачи называют *задачами марковского программирования*. Назовем только некоторые из задач, которые рассматриваются в книге в этом плане: 1) планирование деятельности предприятия; 2) задачи создания запасов; 3) задачи о замене оборудования и другие.

Из сказанного можно уяснить круг задач, который рассматривается в книге. Естественно, что мы не стремились в рамках данной монографии рассмотреть все задачи. Важно, однако, что предлагаемые методы решения могут быть использованы для весьма широкого класса задач.

Вопросы эффективности использования знаковой специфики, присущей задачам данного класса, рассматриваются в книге отдельно. Здесь мы напомним для иллюстрации только о задаче оптимизации режима нефтяного пласта, о которой мы уже говорили выше. Данная задача, сформулированная в виде общей задачи линейного программирования, может быть решена на машинах типа М-220 (без существенного обращения к внешней памяти) максимум для месторождений в 50—70 скважин. При использовании специфических особенностей матрицы эта задача решается на той же машине для месторождений с фондом до 1000 скважин, причем несравнимо быстрее.

ОБЪЕКТЫ МНОГОСВЯЗНОГО УПРАВЛЕНИЯ. ПРИМЕРЫ ЛИНЕЙНЫХ ЗАДАЧ ОПТИМИЗАЦИИ

В этой главе мы рассмотрим некоторые примеры объектов многосвязного управления. Совершенно очевидно, что в рамках одной книги нет возможности описать все объекты многосвязного управления и мы перед собой не ставим такой задачи. Вместе с тем мы постараемся привести примеры таких объектов, которые, с одной стороны, относятся к наиболее интересным с точки зрения современного состояния технического прогресса, с другой стороны, эти объекты избраны нами как в определенном смысле наиболее типичные в промышленной практике и в народном хозяйстве, и, наконец, в-третьих, мы обратимся к анализу объектов, структура которых позволяет с большей полнотой раскрыть предлагаемые методы оптимизации, границу их применимости и их возможности.

Кроме описания принципа действия объекта управления, необходимо сформулировать критерий оптимальности и установить все имеющиеся ограничения, без чего сама постановка задачи оптимизации теряет смысл. Именно в таком плане и будет построено изложение.

Основная идея данной главы заключается в том, чтобы рассмотреть ряд многосвязных объектов самой различной физической природы и выделить специфику, общую для всех рассматриваемых объектов. Эта специфика заключается в том, что при описании данных объектов используются S -матрицы (напомним, что S -матрицей мы называем матрицу, каждая строка и столбец которой содержат не более одного положительного элемента, остальные же элементы — не положительные; см. введение). Задачи оптимизации рассматриваемых объектов сводятся в данной главе к следующей задаче линейного программирования:

$$CX \rightarrow \max, \quad AX \leq B, \quad X \geq 0, \quad (0.1)$$

где A — S -матрица; $B \geq 0$ (C — вектор-строка коэффициентов целевой функции, X — вектор-столбец искомых пере-

менных, A — матрица ограничений, B — вектор-столбец правых частей ограничений). Задачу линейного программирования (0.1) назовем S -задачей. Заметим, что все специальные свойства S -задачи сохраняются и в том случае, когда имеются произвольные знаки ограничивающих неравенств, или когда задача ставится на минимум целевой функции.

Материал главы изложен следующим образом. Сначала в § 1 подробно рассматривается постановка задачи оптимизации для одного конкретного многосвязного объекта — нефтедобывающего предприятия. На примере данного объекта обсуждаются трудности, с которыми приходится сталкиваться при оптимизации. В § 2 на основе анализа характерных типов структур многосвязных объектов устанавливается специфическое свойство объектов данных типов. Это свойство заключается в том, что при описании и, соответственно, при оптимизации указанных объектов используется S -матрица. В остальных параграфах главы рассматривается появление S -матриц при описании и оптимизации конкретных объектов. В § 3 рассматривается важный класс объектов, описываемых уравнениями Лапласа, Пуассона, типа Лапласа, типа Пуассона. Описываются такие общие условия, при которых дифференциальные уравнения в частных производных сводятся после конечно-разностной аппроксимации к системам с S -матрицами. Дается постановка задачи оптимизации и сведение ее к S -задаче. В § 4 рассматриваются задачи регулирования мощности в коллективе радиостанций. Материал § 5 посвящен рассмотрению экономических межотраслевых моделей и моделей обмена. В § 6 на примере электрических цепей постоянного тока рассматривается задача определения оптимальных интенсивностей источников и стоков в некоторых сетях.

§ 1. Нефтедобывающее предприятие (НДП) как объект оптимального управления

В данном параграфе мы рассмотрим некоторые вопросы постановки и решения задач управления режимом нефтяного пласта. Прежде всего, в п. 1 будут приведены некоторые сведения об объекте и обсужден вариационный характер проблемы. Далее, в п. 2 рассматриваются постановки задач оптимизации данного объекта для линейного случая при учете и без учета переходных процессов, а также в случае учета нелинейности некоторых характеристик. В за-

ключение параграфа проводится обсуждение проблем, связанных с постановкой и решением отмеченных задач.

1. Режим и эксплуатация нефтяной залежи как вариационная проблема. Нефть, состоящая из твердых, жидких и газообразных углеводородов, находится в пористой среде и под действием пластовых сил давления приходит в движение и извлекается

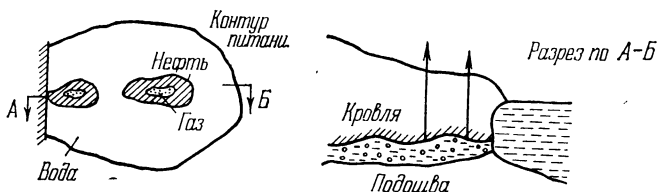


Рис. 1.1. Схема нефтеносного пласта.

на поверхность по стволам эксплуатационных скважин (рис. (1.1)). Для поддержания пластового давления, кроме эксплуатационных, бурятся нагнетательные скважины, через которые в пласт закачивается вода и таким образом в течение всего периода эксплуатации залежи искусственно поддерживается пластовое давление. Когда говорят о рациональной разработке нефтяного месторождения, то имеют в виду извлечение всех промышленных запасов нефти при возможно меньшей себестоимости и в возможно меньшее время.

Мы будем рассматривать проблему *) оптимальной эксплуатации нефтяной залежи в сложившихся условиях, когда пласт полностью разрушен и обустроен [61].

Нефтеносный пласт и скважины образуют единую гидродинамическую систему. Если изменять дебиты отдельных скважин, то это приводит к перераспределениям давлений и потоков внутри залежи в целом. Например, форсировка отборов по отдельным участкам пласта при неизменной закачке снизит с течением времени пластовое давление, и по

*) Вопросы рациональной разработки нефтяных месторождений ставятся как на этапе проектирования, так и на этапе эксплуатации месторождений. Технологические постановки задач оптимального проектирования разработки нефтяных месторождений опираются на работу [46]. Некоторые результаты решения этой большой и самостоятельной проблемы изложены, в частности, в [48].

этой причине многие скважины прекратят фонтанировать; более того, в зону пониженного давления может прорваться краевая вода и отдельные скважины могут преждевременно обводниться. Обводнение и прекращение фонтанирования скважин требуют повышенных затрат на эксплуатацию, и естественно, что с вопросами назначения режимов работы скважин связаны непосредственно экономические факторы. Если искусственно снять ограничения (что физически означает пуск всех скважин на полную нагрузку), то такой режим будет далеко не лучшим. Дело в том, что любой реальный пласт неоднороден по своим физическим и химическим свойствам, и поэтому потенциальные возможности каждой скважины различны. Кроме того, что весьма существенно, начиная с некоторого этапа разработки эксплуатационные скважины обводнены в различной степени. В этой связи ясно, что пуск скважин на полную нагрузку может привести к снижению темпов роста добычи нефти, к существенному повышению количества добываемой воды и, следовательно, к росту себестоимости продукции. Вместе с тем, можно так подобрать режимы работы скважин (уменьшить дебиты сильно обводненных скважин или даже выключить их), что в силу гидродинамической связи между скважинами удастся получить больше продукции с другой, менее обводненной части скважин, и увеличить суммарный дебит нефти.

Зависимость количества добываемой нефти (подчеркнем, — нефти, а не жидкости) от количества эксплуатационных скважин при условии поддержания постоянства давления на контуре и при наличии различной степени обводненности скважин представлена на рис. 1.2. Из этого рисунка, который носит сугубо качественный характер, видно, что

существует некоторое оптимальное число скважин, при котором количество добываемой нефти максимально. В каждом случае оптимальное число скважин, расположение скважин, находящихся в эксплуатации, и режимы этих

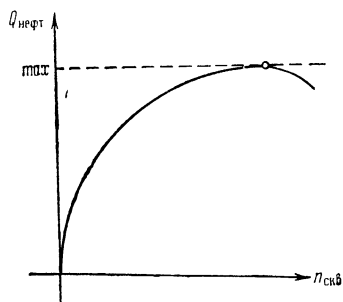


Рис. 1.2. Связь между количеством чистой нефти и количеством работающих скважин.

скважин зависят от конкретных условий: от обводненности, продуктивности каждой скважины и т. д. Важно, однако, что эксплуатационная характеристика пласта имеет явно выраженный экстремум.

Нетрудно видеть, что задача выбора режимов работы скважин при сложившейся системе разработки по своему существу вариационная. Как известно, решение вариационной задачи связано с экстремизацией некоторого функционала при заданных ограничениях. Для того чтобы поставить проблему, нужно определить содержание понятий функционала и ограничений применительно к задаче рациональной разработки нефтяного месторождения. В рассматриваемом случае правомерны две постановки задачи: а) назначить режимы работы скважин из условия обеспечения максимума суммарной добычи нефти при заданных технологических, производственных и экономических ограничениях; б) назначить режимы работы скважин из условия обеспечения минимальной себестоимости нефти при запланированном уровне добычи и заданных технологических и производственных ограничениях.

2. Математическое описание объекта, формулировка целевой функции и технологических ограничений, формулировка задачи оптимизации. Прежде, чем перейти к описанию объекта, поясним некоторые специальные технологические термины, используемые в дальнейшем. Режим неустановившейся фильтрации однородной жидкости в пласте называют *упругим режимом* фильтрации. Установившийся режим фильтрации называют *жестким режимом*. Фильтрационный приток жидкости к забою пробуренной скважины устанавливается в результате разности давлений на контуре питания пласта и давления в районе скважины. Эта разность давлений называется *депрессией*. Давление на контуре питания пласта называется *контурным давлением*, а давление у забоя скважины называется *забойным давлением*. Технологически режим работы каждой скважины характеризуется, в частности, дебитом — количеством жидкости, которое добывается из данной скважины, — и забойным давлением или депрессией.

а) *Упругий режим*. Мы начнем описание объекта управления с рассмотрения упругого режима, как наиболее общего случая. Допустим, что на рассматриваемом промежутке

времени проницаемость пласта, его упругость, а также вязкость жидкости, наполняющей пласт, непрерывные и достаточно число раз дифференцируемые функции времени; тогда распределение давлений в пласте описывается уравнением [89, 91]

$$\frac{\partial}{\partial x} \left(\frac{k_x}{\mu} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{k_y}{\mu} \frac{\partial p}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{k_z}{\mu} \frac{\partial p}{\partial z} \right) = \beta^* \frac{\partial p}{\partial t} + f(x, y, z) \quad (1.1)$$

и в плоском случае

$$\frac{\partial}{\partial x} \left(\frac{k_x}{\mu} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{k_y}{\mu} \frac{\partial p}{\partial y} \right) = \beta^* \frac{\partial p}{\partial t} + f(x, y), \quad (1.2)$$

где k_i — проницаемость пласта в направлении i ($i=x, y, z$), μ — вязкость жидкости, β^* — коэффициент упругости пласта, p — давление, t — время, x, y, z — координаты точки пласта, $f(x, y, z)$ — функция, равная нулю всюду, кроме мест расположения источников, стоков и других особенностей. Рассмотрим плоскую область V , не имеющую особенностей, для правильной подобласти которой справедливо уравнение (1.2). Внутри V расположено n точечных источников, каждый из которых соответствует идеализированной точечной скважине. Режим работы источников можно описать следующим образом:

$$\int_S \frac{k_L}{\mu} \text{grad}_L p dS - \int_V \beta^* \frac{\partial p}{\partial t} dV = F_V(t). \quad (1.3)$$

Здесь $F_V(t)$ зависит только от источников, попавших внутрь области V , не имеющей на своей границе S источников. Ясно, что если V не содержит источников, то $F_V(t) \equiv 0$. Обозначим через $F_V^i(t)$ функцию, которая будет стоять в правой части уравнения (1.3) в том случае, когда V содержит только i -й источник. Тогда, если заданы все $F_V^i(t)$, $i=1, 2, \dots, n$, мы получим $F_V(t)$ как сумму всех $F_V^i(t)$:

$$F_V(t) = \sum_{i \in V} F_V^i(t). \quad (1.4)$$

(Здесь суммирование ведется по тем индексам скважин, которые попали в область V .) Физически уравнение (1.4) определяет для заданной области V удельный мгновенный

расход жидкости, уходящей из этой области через источники. Если $F_V^i(t) > 0$, мы имеем сток, что соответствует эксплуатационной скважине; в случае $F_V^i(t) < 0$ — источник, или что то же самое — нагнетательную скважину.

Для удобства разделим граничные условия для всей области V на две части: внешние и внутренние. Под *внешними граничными условиями* мы будем понимать ту часть общего граничного условия, которая обусловлена внешними факторами, от нас не зависящими и недоступными нашему влиянию. Под *внутренними граничными условиями* мы будем понимать ту совокупность общих граничных условий, на которую мы можем воздействовать. Давление на контуре питания, например, это внешнее граничное условие, а режим работы скважины — внутреннее граничное условие. В свою очередь каждое внутреннее граничное условие можно разбить на такие части, каждую из которых можно варьировать только целиком (например, расход жидкости через скважину).

Для определенности будем считать, что функции $F_V^i(t)$ определяют все внутренние граничные условия. Предположим, что мы нашли решение уравнения (1.2) для случая, когда все внутренние граничные условия равны нулю. Обозначим это решение через p_0 . Обозначим, далее, через p_i решение уравнения (1.2) при нулевых начальных и внешних граничных условиях и при $F_V^j(t) = 0$ для всех $j \neq i$. В силу линейности уравнения (1.2) общее решение задачи при заданных граничном и начальном условиях можно записать в виде

$$p = p_0 + \sum_{i=1}^n p_i. \quad (1.5)$$

Обозначим через ρ_i решение p_i , при условии, что $F_V^i(t) = 1$. Применяя принцип суперпозиции, получаем

$$\begin{aligned} p_i &= F_V^i(0) \rho_i(M, t) + \int_0^t \rho_i(M, t-\theta) \frac{dF_V^i(\theta)}{d\theta} d\theta = \\ &= \int_0^t \frac{\partial \rho_i(M, t-\theta)}{\partial \theta} F_V^i(\theta) d\theta. \end{aligned} \quad (1.6)$$

Здесь M — точка из области V , для которой рассматривается

решение. Подставляя (1.6) в (1.5), получим

$$p = p_0 + \sum_{i=1}^n \int_0^t \frac{\partial \rho_i(M, t-\theta)}{\partial \theta} F_V^i(\theta) d\theta. \quad (1.7)$$

Мы получили выражение для описания давления в пласте. Следует иметь в виду, что в реальных условиях мы имеем дело не с точечными источниками и стоками. Однако для практических расчетов полученные выражения вполне приемлемы.

Имея выражение для распределения давления, можно приступить к формулировке ограничений. Мы будем в дальнейшем называть их *технологическими ограничениями*, понимая под этим ограничения, связанные с эксплуатацией собственно пласта. Первое — это ограничение по пластовым давлениям. Исходя из (1.7),

$$\sum_{i=1}^n \int_0^t \frac{\partial \rho_i(M_i, t-\theta)}{\partial \theta} F_V^i(\theta) d\theta \geq (\leq) p_{\text{пред}} - p_0, \quad (1.8)$$

где $p_{\text{пред}}$ — заданная величина, M_i — i -я координата точки пласта, в которой ограничивается давление. Весьма важным является ограничение на забойные давления. Оно имеет следующий вид:

$$\sum_{i=1}^n \int_0^t \frac{\partial \rho_i(M_i, t-\theta)}{\partial \theta} F_V^i(\theta) dt + r_i F_V^i(t) \leq (\geq) p_{\text{пред}} - p_0, \quad (1.9)$$

где r_i — гидравлическое сопротивление скважин. Кроме того, мы введем естественное ограничение, смысл которого очевиден:

$$\pm F_V^i(t) \geq 0, \quad (1.10)$$

где знак (+) берется для эксплуатационных скважин, а знак (—) для нагнетательных.

Из рассмотрения задачи рациональной разработки нефтяного месторождения вытекает формулировка целевой функции. Действительно, обозначая коэффициент содержания нефти в жидкости, добываемой из i -й скважины, через c_i^0 , мы можем описать общее количество нефти, добываемой за период T , выражением

$$l = \int_0^T \sum_{i=1}^n c_i^0 F_V^i(t) dt, \quad (1.11)$$

откуда ясно, что задача оптимизации состоит в максимизации целевой функции (1.11). Коэффициентам c_i^0 может быть придан и другой смысл; например, c_i^0 может выражать затраты по эксплуатации i -й скважины. Тогда оптимизация сводится к минимизации выражения (1.11). При этом, естественно, должны выполняться ограничения (1.8) — (1.10).

Для общей постановки задачи оптимизации следует рассмотреть ограничения, связанные с возможностями нефтепромыслового оборудования, а также с плановыми заданиями. Подробнее мы эти ограничения запишем ниже при рассмотрении жесткого режима работы пласта, так как эти ограничения в равной мере относятся как к жесткому, так и к упругому режимам. В общем же случае задача оптимизации пласта в упругом режиме может быть сформулирована в следующем виде:

Максимизировать (минимизировать) функционал

$$\left. \begin{aligned} l &= \int_0^T \left[\sum_{i=1}^n c_i^0 F_V^i(t) \right] dt \\ \text{при ограничениях} \\ \int_0^t K(t-\theta) F(\theta) d\theta + AF(t) \{=, \geq, \leq\} P(t), \\ t &\in (0, T), F(t) \geq 0, \end{aligned} \right\} \quad (1.12)$$

где $K(t-\theta)$ — матричное ядро, $F(t) = (F_V^{(1)}, F_V^{(2)}, \dots, F_V^{(n)})$, $P(t)$ — заданная m -мерная вектор-функция, $(0, T)$ — заданный отрезок времени, A — вещественная матрица. Символ $\{=, \geq, \leq\}$ означает, что часть ограничений имеет смысл равенства, часть — неравенства. Практически рассматриваемая задача состоит в требовании максимизации количества нефти или минимизации себестоимости добываемой продукции на заданном интервале времени, при ограничениях, учитывающих сложившуюся систему разработки, требования генерального плана разработки и требования планирующих органов.

Нетрудно видеть, что проблема оптимизации здесь сводится к решению задачи линейного программирования в функциональном пространстве. Действительно, задачу (1.12) можно привести к следующему виду: максимизировать

(минимизировать)

$$l = \int_0^T \left(\sum_{i=1}^N \gamma_i q_i \right) dt$$

при ограничениях

$$\left. \begin{aligned} \int_0^t K(t-\theta) Q(\theta) d\theta + A Q(t) &= P(t), \\ Q(t) = \{q_1(t), q_2(t), \dots, q_n(t)\} &\geq 0, \\ t \in (0, T), \end{aligned} \right\} \quad (1.13)$$

где соответствующие функции получаются из функций, входящих в формулировку задачи (1.12). Задача (1.13) и есть задача линейного программирования в декартовом произведении N функциональных пространств, построенных на отрезке $[0, T]$. Если рассматривать только функции, измеримые по Лебегу, то задача сводится к линейному программированию в пространстве Гильберта.

В абстрактном виде задача формулируется так [62]: максимизировать (минимизировать)

$$l = (\gamma Q)$$

при ограничениях

$$A Q (=) P, \quad Q \geq 0, \quad (1.14)$$

где Q — элемент из пространства Гильберта $H^{(n)}$, P — фиксированный элемент из пространства Гильберта $H^{(m)}$, γ — постоянная вектор-функция, A — линейный оператор из $H^{(n)}$ в $H^{(m)}$. Таким образом, в этом наиболее общем случае задача оптимизации сводится к решению задачи линейного программирования в функциональном пространстве Гильберта.

б) *Жесткий (стационарный) режим*. В жестком режиме все соотношения существенно упрощаются. Так, целевую функцию мы можем написать сразу: действительно, так как F в этом случае не зависит от времени (обозначим $F(t)$ для этого случая через Q), то

$$l = \sum_{i=1}^n c_i^0 q_i = C^0 Q. \quad (1.15)$$

Рассмотрим подробнее технологические ограничения. В жестком режиме уравнение распределения потенциала p

будет иметь следующий вид:

$$\frac{\partial}{\partial x} \left(g_1 \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(g_2 \frac{\partial p}{\partial y} \right) + \frac{\partial}{\partial z} \left(g_3 \frac{\partial p}{\partial z} \right) = f(x, y, z), \quad (1.16)$$

где через p обозначена функция, являющаяся решением (1.16), g_1, g_2, g_3 — известные функции координат, характеризующие неоднородность среды. Значение функции p для произвольной точки M пласта обозначим через $p_M(x, y, z)$. Тогда скорость фильтрации в направлении s_j запишется в виде

$$u_j = \frac{\partial p_M}{\partial s_{jM}}, \quad (1.17)$$

откуда, интегрируя, получаем

$$p_M = s_{Mj} u_j + C. \quad (1.18)$$

Уравнения (1.17) и (1.18) выражают известный линейный закон фильтрации Дарси. Если исходить из этого закона, то для произвольной точки M пласта и дебита j -й скважины получим соотношение

$$p_M = \beta_{Mj} q_j + C, \quad (1.19)$$

где β_{Mj} — известная функция от координат, характеризующая статические свойства неоднородной среды и имеющая размерность гидравлического сопротивления. Если имеется n источников и стоков, то давление в точке M опишется выражением

$$p_M = \sum_{j=1}^n \beta_{Mj} q_j + C. \quad (1.20)$$

При этом β_{Mj} следует брать со знаком (+) для источников и (—) для стоков, так как изменения q_j в нагнетательных и эксплуатационных скважинах приводят к приращениям давления с разными знаками. Если вместо произвольной точки M брать точки на стенке i -й скважины, то с учетом действия n скважин получим

$$p_i = \sum_{j=1}^n \beta_{ij} q_j + C, \quad i = 1, 2, \dots, n. \quad (1.21)$$

Освободимся от произвольной постоянной C ; для этой цели выпишем выражение для давления на контуре питания, полагая, что он расположен достаточно далеко от залежи.

Имеем

$$p_k = \sum_{j=1}^n \beta_{kj} q_j + C. \quad (1.22)$$

Вычитая (1.21) из (1.22), имеем

$$\sum_{j=1}^n (\beta_{kj} - \beta_{ij}) q_j = (p_k - p_i) \quad (1.23)$$

или, окончательно,

$$\sum_{j=1}^n a_{ij}^0 q_j = \delta p_i, \quad i = 1, 2, \dots, n. \quad (1.24)$$

Здесь $a_{ij}^0 = \beta_{kj} - \beta_{ij}$ — параметры, учитывающие гидравлические сопротивления и геометрию области неоднородного пласта, p_i — давление на забое i -й скважины, δp_i — депрессия, $\Delta p_i = p_k - p_i > 0$ — для эксплуатационных скважин, $\delta p_i = p_k - p_i < 0$ — для нагнетательных скважин. Имея в виду соответствующие теоремы существования и единственности решения уравнения типа Лапласа, нетрудно видеть, что для n действующих скважин уравнения системы (1.24) линейно независимы. Они определяют точку замкнутого выпуклого многогранника, составленного из ограничивающих гиперплоскостей и полупространств. Если варьировать p_i и q_j , то годограф вектора $Q = [q_j]$ заполнит некоторую область, содержащую все точки выпуклого множества. Учитывая (1.24), запишем для технологических условий типа $\delta p \leq \delta p^{\text{пред}}$ ограничения, определяющие это выпуклое множество:

$$\sum_{j=1}^n a_{ij}^0 q_j \leq \delta p_i^{\text{пред}}, \quad i = 1, \dots, n. \quad (1.25)$$

Таким образом, мы сформулировали функцию цели и технологические ограничения для жесткого режима эксплуатации пласта. Ограничения $\delta p_i^{\text{пред}}$ можно практически задавать, исходя из допустимых предельных забойных давлений. Кроме того, в некоторых случаях нужно учитывать предельные возможности нефтепромыслового оборудования. В этой связи мы получим дополнительные ограничения вида

$$\text{и} \quad \left. \begin{aligned} q_j &\leq q_j^{\text{пред}} \\ \sum_i q_i &\leq q_i^{\text{пред}} \end{aligned} \right\} \quad (1.26)$$

Условия (1.26) иногда называют *производственными ограничениями*. Остается рассмотреть экономические и плановые ограничения. В сложившейся системе эксплуатации можно полагать капиталовложения заданными постоянными величинами. Поэтому нам необходимо рассмотреть эксплуатационные расходы. Эти расходы по отдельным статьям могут быть представлены как функции дебитов скважин. Так, энергетические затраты можно представить как линейные функции дебитов

$$\sum_j b_j q_j \leq N, \quad (1.27)$$

где b_j — коэффициенты удельных затрат и N — энергетические ограничения. Наконец, запишем требования по плану добычи нефти в виде

$$\sum_{j=1}^n c_j^0 q_j \geq q_n. \quad (1.28)$$

В общем случае задача оптимизации приводится к задаче линейного программирования следующего вида: максимизировать (минимизировать)

$$I = C^0 Q$$

при ограничениях

$$K Q \leq \Gamma \quad \text{и} \quad Q \geq 0; \quad (1.29)$$

здесь K — матрица размерности $n \times m$, учитывающая все ограничения, Γ — соответствующий столбец с заданными элементами. Таким образом, для жесткого режима мы свели задачу оптимизации к задаче линейного программирования в конечномерном евклидовом пространстве.

в) *Нелинейная задача*. Рассмотрим НДП как объект многосвязного управления при учете некоторых нелинейностей. Учет некоторых нелинейностей позволяет приблизить математическую модель объекта к реальности. Основные типы нелинейностей здесь следующие:

1. Нелинейность характеристик погруженных электроцентробежных установок (ПЦН) скважин.
2. Нелинейность, появляющаяся при учете потерь на трение и местные сопротивления при подъеме жидкости из скважины.
3. Нелинейная зависимость критического давления фонтанирования от дебита для фонтанных скважин.

4. Нелинейность характеристик насосов кустовых насосных скважин (КНС) и характеристик водоводов при турбулентном режиме течения жидкости.

5. Нелинейности, определяемые увеличением скорости потока в призабойной зоне скважин.

6. Нарушение линейности закона фильтрации при снижении давления в забоях скважин ниже давления насыщения.

7. Нелинейности в целевой функции.

8. Нелинейная зависимость коэффициента нефтеотдачи от скорости фильтрации.

Рассмотрим подробнее указанные нелинейности и их влияние на ограничения при постановке задачи оптимизации. Ограничения, определяемые необходимым минимальным давлением ПЦН для подъема жидкости из скважины в случае линеаризованной характеристики насоса запишутся

$$\sum_{j=1}^n \alpha_{ij}^0 q_j + \alpha_i q_i \leq p_i^{\text{пред}}, \quad i = 1, \dots, m_1, \quad (1.30)$$

где α_i — коэффициент, определяющий наклон линеаризованной характеристики ПЦН, $p_i^{\text{пред}}$ — допустимые предельные значения давлений, m_1 — число скважин с ПЦН, n — общее число скважин месторождений. При учете действительной (нелинейной) характеристики ПЦН ограничения (1.30) запишутся в виде

$$\sum_{j=1}^n \alpha_{ij}^0 q_j + f_i^1(q_i) \leq p_i^{\text{пред}}, \quad i = 1, \dots, m_1, \quad (1.31)$$

где $p_i - f_i^1(q_i)$ — действительная характеристика ПЦН. Так как $f_i^1(q_i) \leq \alpha_i q_i$ для любого q_i , то ограничения (1.31) шире ограничений (1.30). Иначе говоря, постановка задачи при учете действительных характеристик ПЦН может существенно увеличить значения оптимальных дебитов скважин. Нелинейность, обусловленная учетом потерь на трение и местные сопротивления при подъеме жидкости из скважин, может быть суммарно учтена в члене $f_i^1(q_i)$ условий (1.31).

Ограничения забойного давления фонтанных скважин имеют вид

$$p_i^{\Phi} \geq p_i^{\Phi, \text{пред}}. \quad (1.32)$$

Благодаря нелинейной зависимости p_i^{Φ} пред от дебита условие (1.32) может быть записано в виде

$$p_i^{\Phi} \geq f_i^{(2)}(q_i). \quad (1.33)$$

Перейдя к выражениям для депрессий, получим ограничения по фонтанным скважинам:

$$\sum_{j=1}^n a_{ij}^0 q_j + f_i^{(2)}(q_i) \leq p_{*}. \quad (1.34)$$

Зависимость критического забойного давления от дебита фонтанной скважины может быть получена расчетным путем по формулам А. П. Крылова и С. С. Лутошкина [47].

Условия, связанные с ограниченными возможностями системы поддержания пластового давления при учете нелинейности характеристик КНС, аналогично (1.31) запишутся в виде

$$\sum_{j=1}^n a_{ij}^0 q_j + \sum_{j=v}^w a_{ij}^{(2)} q_j + f_i \left(\sum_{j=v}^w q_j \right) \leq p_i^{\text{пред}}, \quad i = v, \dots, w, \quad (1.35)$$

где $a_{ij}^{(2)}$ — коэффициент, учитывающий потери на трение на участке водовода от l -й КНС к j -й нагнетательной скважине; $v, \dots, w$ — номера нагнетательных скважин, при-

соединенных к l -й КНС; $p_i - f_i \left(\sum_{j=v}^w q_j \right)$ — действительная

характеристика насосов КНС. При турбулентном режиме ограничения (1.35) примут вид

$$\sum_{j=1}^n a_{ij}^0 q_j + f_i(q_v, \dots, q_w) + f_i \left(\sum_{j=v}^w q_j \right) \leq p_i^{\text{пред}}, \quad i = v, \dots, w, \quad (1.36)$$

где $f_i(q_v, \dots, q_w)$ — потери на трение при турбулентном законе фильтрации [79]. Наконец, учет нарушения линейного закона фильтрации в случае увеличения скорости потока в призабойной зоне скважин приводит к появлению нелинейности в диагональных членах матрицы технологических

ограничений задачи:

$$f_i^{(3)}(q_i) + \sum_{j=1}^n a_{ij}^0 q_j \leq p_i^{\text{пред}}, \quad i = 1, \dots, n. \quad (1.37)$$

Здесь $f_i^{(3)}(q_i)$ — член, определяющий искажение линейного закона фильтрации в призабойной зоне скважин [79]. Формулировка задачи оптимизации здесь, собственно, такая же, как ранее, а именно: найти вектор дебитов Q , экстремизирующий некоторую (в общем случае нелинейную) функцию цели, при ограничениях (1.31), (1.34), (1.36), (1.37). Нетрудно показать, что функции

$$f_i^{(1)}(q_i), f_i^{(2)}(q_i), f_i\left(\sum_{j=v}^w q_j\right), f_i(q_v, \dots, q_w) \text{ и } f_i^{(3)}(q_i) \text{ в ог-}$$

раничениях (1.31) — (1.37) являются выпуклыми. В связи с этим рассматриваемую задачу можно отнести к классу задач выпуклого программирования. Из рассмотрения выражений (1.31), (1.34) и (1.37) видно, что нелинейности в основном окажутся в членах главной диагонали матрицы ограничений задачи оптимального управления нефтедобывающим предприятием. Эту особенность задачи мы учтем в дальнейшем, при разработке алгоритма решения задачи.

3. З а м е ч а н и я п о п о с т а н о в к е з а д а ч и о п т и м и з а ц и и Н Д П. Чтобы завершить постановку задачи оптимизации режима работы нефтяного пласта, необходимо отметить некоторые весьма важные вопросы. При рассмотрении упругого, а также жесткого режима, мы предполагали, что так называемые коэффициенты взаимовлияния a_{ij} являются известными и заданными постоянными величинами. Кроме того, мы полагали, что коэффициенты c_j , определяющие содержание нефти в извлекаемой жидкости в каждой скважине, также являются величинами постоянными. Фактически дело обстоит значительно сложнее и для того, чтобы пользоваться этими данными, следует иметь в виду следующее. В существующей практике коэффициенты взаимовлияния определяются с помощью сеточной модели. Получаемые коэффициенты взаимовлияния являются некоторыми усредненными величинами. Поэтому в процессе работы необходимо систематически совершенствовать модель с учетом реальных замеров дебитов и давлений. С этой точки

зрения начальные данные о коэффициентах взаимовлияния следует считать первым приближением. Опыт, однако, показывает, что эти начальные сведения дают практически достаточно хорошее приближение к реальным условиям. При использовании сеточных моделей процесс оптимизации может производиться на стадии идентификации. Этому важному вопросу посвящена специальная глава книги. В процессе оптимизации с учетом изменения натуральных данных замеров могут усовершенствоваться данные модели. Это один из возможных путей уточнения параметров модели пласта.

Рассмотрим теперь проблему, связанную с коэффициентами целевой функции, определяющей процент содержания нефти в добываемой жидкости. Совершенно ясно, что в результате естественного процесса обводнения эти коэффициенты являются переменными величинами. Они меняются еще и по другой причине. Действительно, определив оптимальный процесс, мы определяем режимы скважин, которые отличны от режимов до оптимизации. В результате получаем перераспределение потоков и давлений в пласте. С течением времени начинают изменяться обводненности отдельных скважин. Таким образом, в результате указанных факторов коэффициенты целевой функции являются величинами переменными.

Однако и это обстоятельство не может служить препятствием для использования в этом случае методов программирования. Дело в том, что нефтяной пласт как объект управления является весьма инерционным. Процессы протекают медленно. По этой причине найденный оптимальный режим при заданных значениях коэффициентов взаимовлияния и заданных значениях коэффициентов целевой функции будет в течение определенного промежутка времени оставаться оптимальным. Этот промежуток времени тем больше, чем хуже коллекторские свойства пласта. Практически обнаружить изменения коэффициентов целевой функции можно непосредственными замерами степени обводненности добываемой жидкости в каждой скважине. Ясно, что если эти обводненности существенно изменились, то режим будет уже не оптимальным. Для получения оптимального режима нужно произвести расчет при учете новых коэффициентов целевой функции. Таким образом, оптимальный режим эксплуатации нефтяного пласта рассчитывается для каждого отдельного

отрезка времени, в течение которого ведется непрерывный контроль водного режима пласта и, при отклонениях обводненности скважин, выходящих за установленные пределы, производится перерасчет и находится новый оптимальный режим.

Как видно из данных замечаний, задача оптимизации режима нефтяного пласта в рассмотренной постановке является задачей не стратегической, а тактической. То есть ставится вопрос о назначении оптимальных режимов скважин на каждый период времени при выполнении условий, определяемых общим генеральным планом разработки данного месторождения. Условия конечной нефтеотдачи при этом могут быть учтены путем задания ограничений на дебиты и депрессии скважин, обеспечивающие привязку языков воды, требуемую скорость фильтрации и др.

4. З а м е ч а н и я п о р е ш е н и ю з а д а ч и о п т и м и з а ц и и Н Д П. Выше, в п. 2 мы рассмотрели постановки задачи оптимизации режимов скважин для различных случаев: а) в линейном виде с учетом переходных процессов; б) без учета переходных процессов; в) в нелинейном виде (при учете некоторых нелинейных характеристик). О решении задач (в) и (а) мы будем говорить в других главах книги. Здесь мы остановимся на задаче (б).

Решение этой задачи представляет наибольший практический интерес из следующих соображений, диктуемых реальными промысловыми условиями. Во-первых, точность модели пласта является весьма невысокой даже в линейной статической постановке. Это объясняется тем, что судить о параметрах пласта можно только по замерам в отдельных точках-скважинах, причем точность замеров весьма невысокая. Во-вторых, на функционирование НДП оказывают постоянное воздействие различные шумы (аварии на скважинах, в системе сбора и т. п.). Поэтому прежде, чем переходить от модели (б) к более сложным моделям, следует выяснить практическую целесообразность этого.

Итак, рассмотрим задачу линейного программирования (1.29). Размерность этой задачи в реальных условиях может значительно превосходить возможности вычислительных машин. Использование агрегирования или декомпозиции также не приводит в данном случае к ощутимому успеху. И вот почему. Под агрегированием понимается замена

исходных показателей задачи обобщенными, являющимися некоторыми функциями исходных. Агрегирование может проводиться на нескольких уровнях. Исходная задача в этом случае разбивается на несколько подзадач относительно переменных каждого уровня иерархии. Однако нельзя подходить к вопросу оптимизации иерархических структур упрощенно — решать задачу верхнего уровня (относительно переменных $Q_l^{\text{ар}}$, $l=1, \dots, N$, где N — число обобщенных показателей), а затем, используя полученное решение в качестве ограничений при решении N независимых подзадач нижнего уровня, находить оптимальные значения q_j в каждой l -й подзадаче. При таком подходе в точке решения будет получено значение целевой функции, меньшее действительного оптимума — за счет введения в задачу нижнего уровня ограничений $Q_l^{\text{ар}} = \sum q_j$, искусственно отсеивающих часть множества допустимых решений. Это может быть допустимо только в тех случаях, когда удастся доказать, что потеря в оптимуме будет незначительной. Эти частные случаи наблюдаются тогда, когда имеется возможность провести некоторую естественную агрегацию переменных. А такая возможность появляется в рассматриваемых задачах довольно редко.

Процедура оптимизации иерархических моделей в общем случае должна строиться на основе итерационной процедуры согласования моделей различных уровней. При этом информация подается по рангам иерархии, как «сверху вниз», так и «снизу вверх». Но использование процедуры итеративного агрегирования наталкивается на серьезные трудности, связанные как с теоретическим доказательством сходимости получаемых решений к оптимальному решению, так и с достижением приемлемой для практики скорости сходимости процесса.

Использование декомпозиции (разложения) может быть достаточно эффективно при блочной структуре основной части матрицы ограничений. В этом случае выделение связующей части матрицы с помощью различных процедур декомпозиции позволяет разбить задачу на отдельные, не связанные блоки. Но блочная структура наблюдается в рассматриваемых задачах только в отдельных частных случаях, например, тогда, когда ставится общая задача оптимизации режима сразу для нескольких гидродинамически не свя-

занных нефтяных пластов. В общем же случае структура матрицы ограничений в рассматриваемых задачах бывает произвольной. Применение многократного разложения для решения задач, обладающих произвольной структурой матрицы, невозможно, так как это во много раз уменьшает скорость сходимости и без того очень медленно сходящегося процесса декомпозиции.

Итак, мы выяснили трудности, с которыми приходится сталкиваться при решении рассматриваемой задачи оптимизации. Путь решения задачи мы обсудим ниже, после того как выясним в следующем параграфе некоторые общие черты, присущие широкому классу многосвязных объектов, к которому относится, в частности, и нефтяной пласт.

§ 2. Структурные представления многосвязных объектов. S-матрицы

В предыдущем параграфе мы на конкретном примере убедились в том, что одна из наибольших трудностей, возникающих при оптимизации сложных многосвязных систем, обусловлена размерностью решаемой задачи. Как правило, многосвязные объекты и системы имеют достаточно большое число регулируемых величин. Это приводит к большим вычислительным трудностям, часто непреодолимым даже при использовании самых совершенных, современных вычислительных машин. Выход в таких случаях лежит на пути использования особенностей объекта и разработки методов решения задачи программирования с использованием этих особенностей. По сути дела, вся книга посвящена этой основной теме. Здесь рассмотрим ту специфику, которая привносится в матрицу ограничений свойствами многосвязных объектов.

1. Особенности многосвязных объектов. Наиболее распространенные многосвязные объекты могут быть представлены двумя характерными структурами [56]. Первая разновидность структуры представлена на рис. 1.3 и носит название *структуры с прямыми перекрестными связями*. Как видно из рисунка, входные величины x_j действуют через коэффициенты взаимовлияния α_{ij} непосредственно на данную выходную величину y_i . Уравнение для любой выходной величины системы с отрицательными

прямыми связями запишется следующим образом:

$$y_i = \alpha_{ii}x_i - \sum_{\substack{j=1 \\ j \neq i}}^n \alpha_{ij}x_j, \quad i = 1, 2, \dots, n. \quad (2.1)$$

Здесь α_{ii} называют собственным коэффициентом влияния; α_{ij} — взаимным коэффициентом влияния. Коэффициенты α_{ii}

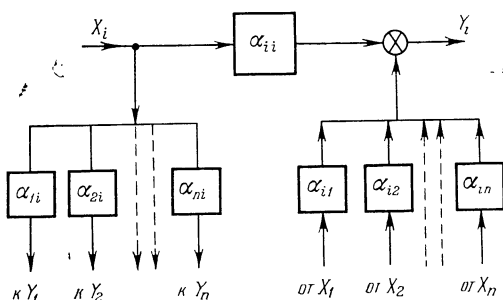


Рис. 1.3. Элемент многосвязного объекта с прямыми перекрестными связями.

и α_{ij} могут выражать как динамическую связь (тогда это будут некоторые операторы), так и статическую связь (тогда это будут постоянные коэффициенты). Динамические связи в установившемся режиме также определяются постоянными коэффициентами связи. Коэффициенты с одинаковыми индексами i, j определяют самовлияние, а с разными индексами — взаимовлияние. Как правило, собственное влияние всегда больше, чем взаимное. Обратим внимание на следующее весьма важное свойство многосвязных объектов с прямыми отрицательными связями: именно в матрице коэффициентов влияния все недиагональные элементы неположительные. Матрицу с такими элементами мы называли S -матрицей. Она имеет следующий вид:

$$A = \begin{bmatrix} +\alpha_{11} & -\alpha_{12} & -\alpha_{13} & \dots & -\alpha_{1n} \\ -\alpha_{21} & +\alpha_{22} & -\alpha_{23} & \dots & -\alpha_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ -\alpha_{n1} & -\alpha_{n2} & -\alpha_{n3} & \dots & +\alpha_{nn} \end{bmatrix}.$$

Если матрица имеет размерность $n \times m$, то положительными будут элементы на псевдодиагонали, остальные эле-

менты — отрицательные. Некоторые коэффициенты могут быть также нулевыми.

Вторая разновидность структуры представлена на рис. 1.4 и носит название *структуры с обратными перекрестными связями*. В этом виде структуры несобственные выходные величины $y_j (j \neq i)$ действуют через коэффициенты

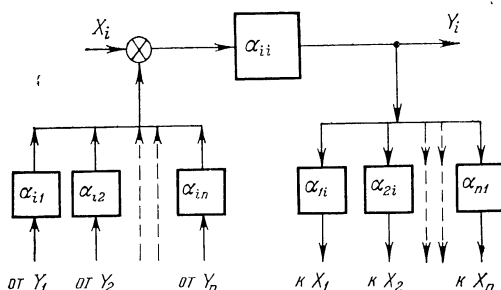


Рис. 1.4. Элемент многосвязного объекта с обратными перекрестными связями.

взаимовлияния α_{ij} на вход i -го сепаратного элемента многосвязной системы. Из рис. 1.4 ясно, что уравнения для выходных величин объекта с положительными обратными связями запишутся в виде

$$y_i = \alpha_{ii} \left(x_i + \sum_{\substack{j=1 \\ j \neq i}}^n \alpha_{ij} y_j \right), \quad i = 1, 2, \dots, n,$$

или

$$x_i = \frac{1}{\alpha_{ii}} y_i - \sum_{\substack{j=1 \\ j \neq i}}^n \alpha_{ij} y_j, \quad i = 1, 2, \dots, n. \quad (2.2)$$

Матрица системы (2.2) также является S -матрицей.

Уравнения, связывающие входные и выходные величины для многосвязных объектов с отрицательными обратными связями, имеют вид

$$x_i = \frac{1}{\alpha_{ii}} y_i + \sum_{\substack{j=1 \\ j \neq i}}^n \alpha_{ij} y_j, \quad i = 1, 2, \dots, n. \quad (2.3)$$

В то же время для объектов с положительными прямыми связями имеем

$$y_i = \alpha_{ii}x_i + \sum_{\substack{j=1 \\ j \neq i}}^n \alpha_{ij}x_j, \quad i = 1, 2, \dots, n. \quad (2.4)$$

Хорошо известно [65], что структуры с обратными перекрестными связями могут быть приведены к структурам с прямыми перекрестными связями и наоборот. Благодаря этому часто удается перейти от объекта с отрицательными обратными связями к объекту (2.1), а от объекта с положительными прямыми связями к объекту (2.2). Таким путем можно получить S -матрицу и в последних двух случаях.

В ряде случаев подобный переход может быть сделан чрезвычайно просто. Рассмотрим, например, объект с обратными отрицательными связями (см. рис. 1.4) и поменяем местами входы x_j и выходы y_i ($i=j$). Иначе говоря, изменим направление передачи сигнала: будем считать входами y_i , а выходами x_j . В ряде случаев такое изменение с точки зрения решаемой задачи имеет смысл. По отношению к новым входам и выходам структура является уже с прямыми перекрестными связями (см. рис. 1.3 и 1.4). Запишем уравнение (2.3) объекта в матричном виде:

$$x = A^0 Y, \quad (2.5)$$

где A^0 — неотрицательная матрица, X , Y — векторы входных и выходных величин объекта. После изменения направления передачи сигнала получим

$$Y = (A^0)^{-1} X. \quad (2.6)$$

Мы считаем в данном случае матрицу A^0 квадратной. В реальных задачах, рассматриваемых в данной главе, это предположение также можно оставить в силе, так как неквадратная матрица ограничений задачи оптимизации всегда может быть дополнена до квадратной путем введения заведомо «лишних» условий, не сужающих области допустимых точек. Исследуем структуру матрицы $(A^0)^{-1}$. Пусть одной входной величине x_k мы дали приращение δx_k , а все остальные $x_j = \text{const}$ ($j=1, 2, \dots, n, j \neq k$). Если при этом соответствующая выходная величина y_k получит положительное приращение $\delta y_k > 0$ (см. рис. 1.3), а все остальные выходные

величины y_i ($i=1, 2, \dots, n, i \neq k$) получают отрицательные (или неположительные) приращения $\delta y_i \leq 0$, значит, элементы $a_{ik} = \delta y_i / \delta x_k$ столбца k матрицы $(A^0)^{-1}$ имеют вид

$$A_k = \begin{bmatrix} -\delta y_1 / \delta x_k \\ -\delta y_2 / \delta x_k \\ \vdots \\ +\delta y_k / \delta x_k \\ \vdots \\ -\delta y_n / \delta x_k \end{bmatrix}. \quad (2.7)$$

Если при подаче скачка на любой из входов знаки приращений выходных величин подобны знакам элементов δy_i из (2.7), значит, матрица $(A^0)^{-1}$ является S -матрицей A , т. е. $(A^0)^{-1} = A$. Таким образом, из физических соображений можно установить, когда будет отрицательный характер связей при подобном переходе от структуры с обратными отрицательными связями к структуре с прямыми связями. Совершенно аналогично можно осуществить переход от объекта с положительными прямыми связями к объекту с обратными связями (см. рис. 1.3 и 1.4), и установить, сохраняется ли при этом положительный характер связей.

Для сведения к S -матрице неотрицательной матрицы, описывающей многосвязный объект с прямыми положительными или с обратными отрицательными связями, можно использовать также элементарные алгебраические преобразования, если эта матрица обладает диагональным преобладанием (т. е. если «самовлияние» больше «взаимовлияния»). Смысл этих преобразований заключается в вычитании из элементов исходной матрицы некоторого достаточно большого числа.

2. Применение к задаче оптимизации НДП. Рассмотрим в качестве примера структурные преобразования для многосвязного объекта, описанного в предыдущем параграфе, для нефтедобывающего предприятия (НДП). Если в качестве входов x_i считать депрессии по скважинам, а в качестве выходов y_i — дебиты скважин q_i , то нефтяной пласт можно рассматривать как многосвязный объект с отрицательными обратными связями (см. (1.24) и (2.3)). Уравнение (2.5) тогда будет иметь вид

$$\delta P = A^0 Q, \quad (2.8)$$

где $A^0 \geq 0$ — матрица коэффициентов взаимовлияния скважин по давлению. После перехода от данной структуры к структуре с прямыми связями получим

$$Q = (A^0)^{-1} \delta P,$$

где $(A^0)^{-1}$ — матрица коэффициентов взаимовлияния скважин по дебиту. Исследуем матрицу $(A^0)^{-1}$. Коэффициент a_{ik} этой матрицы показывает (2.7), насколько изменится дебит (δy_i) какой-либо i -й скважины от увеличения депрессии $\delta x_k > 0$ и, соответственно, от увеличения дебита $\delta y_k > 0$ данной k -й скважины. Физически очевидно, что увеличение дебита данной скважины $(\delta y_k > 0)$ может вызвать только уменьшение дебитов всех других скважин $(\delta y_i \leq 0)$. Иными словами, коэффициенты взаимовлияния по дебиту — неположительные числа, коэффициенты самовлияния — положительные числа. Таким образом, наблюдается отрицательный характер связей в получаемой структуре с прямыми связями; матрица $(A^0)^{-1}$ является S -матрицей:

$$(A^0)^{-1} = A.$$

С практической точки зрения переход в данной задаче от матрицы A^0 к матрице A не вызывает дополнительных трудностей, так как матрицу A можно отыскивать не как обратную матрице A^0 , а сразу определяя матрицу коэффициентов взаимовлияния скважин по дебиту вместо матрицы коэффициентов взаимовлияния скважин по давлению.

Мы рассмотрели переход к S -матрице в уравнениях объекта. Но, кроме этих уравнений, при постановке задачи оптимизации накладываются еще ограничения на переменные Q , δP . Основные технологические ограничения задачи (1.29) имеют вид

$$A^0 Q \leq \delta P^{\text{нр}}, \quad (2.9)$$

$$Q \geq 0. \quad (2.10)$$

Запишем систему (2.9) в виде равенств

$$A^0 Q + EX = \delta P^{\text{нр}}, \quad (2.11)$$

где X — n -мерный вектор-столбец дополнительных переменных, обращающих неравенства (2.9) в равенства; вектор X связан с вектором депрессий δP следующим образом:

$$X = \delta P^{\text{нр}} - \delta P$$

и имеет размерность депрессий; E — единичная матрица. Умножим условия (2.11) на матрицу $(A^0)^{-1}$ слева. Тогда

$$EQ + (A^0)^{-1}X = (A^0)^{-1} \delta P^{\text{нр}}$$

или, учитывая, что $(A^0)^{-1}A = I$ и обозначая $A\delta P^{\text{нр}}$ через B ,

$$EQ + AX = B, \quad (2.12)$$

где $B = A\delta P^{\text{нр}}$ — вектор дебитов скважин в случае, когда все они работают при предельных значениях депрессий. Заметим, что для реальных задач оптимизации нефтедобычи обычно выполняется достаточно широкое условие $B \geq 0$. Об этом говорится ниже, в § 1 пятой главы. Там же показывается, что в тех случаях, когда данное условие не выполняется, необходимо изменить саму формулировку задачи.

Итак, мы перешли от ограничения (2.9) к ограничению (2.12), использующему S -матрицу. Рассмотрим, как изменяется при подобном переходе целевая функция и остальные ограничения задачи (1.29). Целевая функция задачи относительно переменных X запишется в виде

$$L = CX \rightarrow \max, \quad (2.13)$$

где

$$C = -C^0 A. \quad (2.14)$$

Получить вектор C можно либо так, как это показано в (2.14) — производя умножение строки C^0 на матрицу A , либо решив систему $A^0 C = C^0$, что можно сделать непосредственно на модели пласта одновременно со снятием коэффициентов матрицы A (или матрицы A^0). Таким образом, задача оптимизации режимов нефтяного пласта при учете технологических ограничений (2.9), (2.10) сводится к следующей задаче линейного программирования с S -матрицей ограничений (S -задача):

$$L = CX \rightarrow \max, \quad AX \leq B, \quad X \geq 0. \quad (2.15)$$

Здесь ограничения получены из условий (2.12), записанных в виде неравенств, вектор C определяется из (2.14), вектор $B \geq 0$. Если в задаче оптимизации нефтедобычи (1.29), кроме основных ограничений (2.9), (2.10), имеются еще ограничения вида

$$A^{\text{д}} Q \leq B^{\text{д}}, \quad (2.16)$$

то после переформулирования (2.15) эти ограничения примут вид

$$A'X \leq B', \quad (2.17)$$

где

$$A' = -A^A A, \quad B' = B^A - (A^A A) \delta P^{\text{пр}}. \quad (2.18)$$

Получение выражений (2.17), (2.18), так же как и выражений (2.13), (2.14), становится понятным, если рассмотреть преобразование исходной матрицы задачи (1.29) по рекуррентным формулам модифицированных жордановых исключений, обобщенных на случай блочного направляющего элемента матрицы A^0 (см. § 1 приложения).

Часть условий (2.16) может быть также сведена к виду, использующему S -матрицу с помощью алгебраических преобразований. Данные преобразования могут заключаться, например, в следующем. Из матрицы A' выделяется подматрица с диагональным преобладанием, т. е. находится подматрица, в каждой s -й строке и l -м столбце которой имеется элемент a_{sl} , превосходящий по величине остальные элементы данной строки и столбца:

$$a_{sl} \geq \max_{i, j} \{a_{sj}, a_{il}\}, \quad j \neq l, i \neq s.$$

Тогда для сведения к S -матрице достаточно провести преобразования, в результате которых из каждой s -й строки данной подматрицы будет вычитаться элемент $a_{sl}^m = \left(\max_{i, j \neq l} a_{sj} \right) <$

$< a_{sl}$. При этом нужно учитывать, что вектор правой части ограничений S -задачи должен быть неотрицательным. Если же среди условий (2.16) имеются ограничения, которые не удастся свести к рассматриваемому специальному виду, то их следует выделить с помощью различных алгоритмов декомпозиции. В третьей главе книги рассматриваются некоторые случаи достаточно эффективного применения методов декомпозиции для решения S -задач с дополнительными ограничениями. Это позволяет расширить возможности применения специальных методов решения S -задач.

Подведем некоторые итоги. В данном параграфе на основе рассмотрения двух основных типов многосвязных объектов (с прямыми и с обратными перекрестными связями) мы отметили, что характерной особенностью подобных объектов

является использование при их описании матриц специального вида (S -матриц). Широкий класс многосвязных объектов с прямыми отрицательными связями и объектов с обратными положительными связями описывается уравнениями, уже в исходном виде содержащими S -матрицу. Описание многосвязных объектов с прямыми положительными связями и объектов с обратными отрицательными связями обычно также удается с помощью структурных или алгебраических преобразований свести к соответствующим объектам с обратными и прямыми связями, и тем самым получить в описании S -матрицу.

Мы рассмотрели также в данном параграфе использование структурных особенностей многосвязных объектов на примере конкретного объекта — нефтедобывающего предприятия, и показали, каким образом удастся свести данную задачу в статической постановке к специальному виду (к S -задаче).

§ 3. Объекты, описываемые некоторыми дифференциальными уравнениями в частных производных (уравнениями Лапласа, Пуассона и другими)

Ранее мы рассмотрели нефтяной пласт как объект многосвязного управления и убедились, что в жестком режиме его работы проблема оптимизации сводится к решению задачи линейного программирования; при этом матрица ограничений есть S -матрица. В этом параграфе мы покажем, что оптимизация объектов, которые описываются некоторыми дифференциальными уравнениями в частных производных эллиптического типа (в частности, уравнениями Лапласа, типа Лапласа, Пуассона, типа Пуассона), также сводится к задаче линейного программирования и матрицы ограничений являются S -матрицами.

1. Конечно-разностная аппроксимация уравнений объекта.

1. Пусть имеется объект, описываемый уравнением Лапласа. Для простоты изложения будем рассматривать двумерный случай. Тогда уравнения объекта запишутся в виде

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = 0. \quad (3.1)$$

Приведем запись данного уравнения в конечных разностях [7, 34]. После дискретизации имеем

$$\left. \begin{aligned} \left(\frac{\partial^2 p}{\partial x^2} \right)_{x,y} &\approx \frac{p(x+h, y) - 2p(x, y) + p(x-h, y)}{h^2}, \\ \left(\frac{\partial^2 p}{\partial y^2} \right)_{x,y} &\approx \frac{p(x, y+h) - 2p(x, y) + p(x, y-h)}{h^2}, \end{aligned} \right\} \quad (3.2)$$

где $h = \Delta x = \Delta y$ — шаг дискретизации по координатам x и y . Подставив (3.2) в (3.1), получим следующую систему линейных алгебраических уравнений:

$$4p(x, y) - p(x+h, y) - p(x-h, y) - p(x, y+h) - p(x, y-h) = 0$$

для всех внутренних узлов сетки $\{x, y\}$. Узлы сетки с координатами $\{x_l, y_t\}$ удобнее обозначать через $\{l, t\}$ (рис. 1.5), и вместо $p(x_l, y_t)$ писать p_{lt} . Пусть M — множество внут-

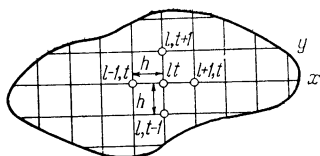


Рис. 1.5. Аппроксимация непрерывной области сеткой.

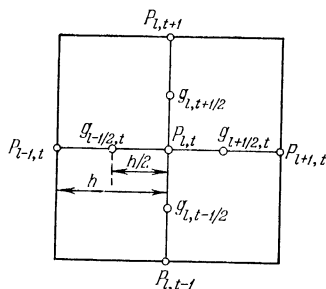


Рис. 1.6. Сеточная аппроксимация для уравнений типа Лапласа.

ренних точек, в которых определяются значения p_{lt} . Тогда рассматриваемая система уравнений примет вид

$$4p_{lt} - p_{l+1, t} - p_{l-1, t} - p_{l, t+1} - p_{l, t-1} = 0$$

$$\text{для всех узлов } \{l, t\} \in M. \quad (3.3)$$

Рассмотрим теперь объекты, описываемые уравнением типа Лапласа,

$$\frac{\partial}{\partial x} \left(g \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(g \frac{\partial p}{\partial y} \right) = 0, \quad (3.4)$$

где

$$g = g(x, y).$$

Раскрыв выражения в скобках как производные произведений двух функций, получим известное уравнение

$$g \frac{\partial^2 p}{\partial x^2} + g \frac{\partial^2 p}{\partial y^2} + g'_x \frac{\partial p}{\partial x} + g'_y \frac{\partial p}{\partial y} = 0. \quad (3.5)$$

После дискретизации (рис. 1.6) имеем

$$\left. \begin{aligned} \left(\frac{\partial p}{\partial x} \right)_{lt} &\approx \frac{p_{l+1, t} - p_{l-1, t}}{2h}; \\ \left(\frac{\partial p}{\partial y} \right)_{lt} &\approx \frac{p_{l, t+1} - p_{l, t-1}}{2h}, \\ \left(\frac{\partial^2 p}{\partial x^2} \right)_{lt} &\approx \frac{p_{l+1, t} - 2p_{lt} + p_{l-1, t}}{h^2}; \\ \left(\frac{\partial^2 p}{\partial y^2} \right)_{lt} &\approx \frac{p_{l, t+1} - 2p_{lt} + p_{l, t-1}}{h^2}, \end{aligned} \right\} \quad (3.6)$$

$$\left. \begin{aligned} g_{lt} &\approx \frac{g_{l+1/2, t} + g_{l-1/2, t}}{2} = \frac{g_{l, t+1/2} + g_{l, t-1/2}}{2}, \\ (g'_x)_{lt} &\approx \frac{g_{l+1/2, t} - g_{l-1/2, t}}{h}; \\ (g'_y)_{lt} &\approx \frac{g_{l, t+1/2} - g_{l, t-1/2}}{h}. \end{aligned} \right\} \quad (3.7)$$

Подставив выражения (3.6) в (3.5), получим

$$\begin{aligned} &\left[g_{lt} - \frac{h}{2} (g'_x)_{lt} \right] p_{l-1, t} + \left[g_{lt} + \frac{h}{2} (g'_x)_{lt} \right] p_{l+1, t} + \\ &+ \left[g_{lt} - \frac{h}{2} (g'_y)_{lt} \right] p_{l, t-1} + \left[g_{lt} + \frac{h}{2} (g'_y)_{lt} \right] p_{l, t+1} - \\ &- 4g_{lt} p_{lt} = 0 \quad \text{для всех узлов } \{l, t\} \in M. \end{aligned} \quad (3.8)$$

Подставив дискретизацию (3.7) в (3.8), и представив член $4g_{lt} p_{lt}$ в виде

$$\begin{aligned} 4g_{lt} p_{lt} &= (2g_{lt} + 2g_{lt}) p_{lt} = \\ &= (g_{l+1/2, t} + g_{l-1/2, t} + g_{l, t+1/2} + g_{l, t-1/2}) p_{lt}, \end{aligned}$$

получим

$$\begin{aligned}
 & -g_{l-1/2, t} p_{l-1, t} - g_{l+1/2, t} p_{l+1, t} - \\
 & \quad - g_{l, t-1/2} p_{l, t-1} - g_{l, t+1/2} p_{l, t+1} + \\
 & \quad + (g_{l-1/2, t} + g_{l+1/2, t} + g_{l, t-1/2} + g_{l, t+1/2}) p_{lt} = 0
 \end{aligned}$$

для всех узлов $\{l, t\} \in M$. (3.9)

Рассмотрим теперь объект, описываемый уравнением Пуассона:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = f(x, y) \quad (3.10)$$

или уравнением типа Пуассона

$$\frac{\partial}{\partial x} \left(g \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left(g \frac{\partial p}{\partial y} \right) = f(x, y) \quad (g = g(x, y)). \quad (3.11)$$

В конечных разностях уравнения (3.10) и (3.11) запишутся аналогично (3.3) и (3.9) с той лишь разницей, что в правой части появляется член $-h^2 f(x_l, y_t) = f_{lt}^1$, т. е. уравнение Пуассона принимает вид

$$4p_{lt} - p_{l+1, t} - p_{l-1, t} - p_{l, t+1} - p_{l, t-1} = f_{lt}^1$$

для всех $\{l, t\} \in M$, (3.12)

а уравнение типа Пуассона —

$$\begin{aligned}
 & -g_{l-1/2, t} p_{l-1, t} - g_{l+1/2, t} p_{l+1, t} - \\
 & \quad - g_{l, t-1/2} p_{l, t-1} - g_{l, t+1/2} p_{l, t+1} + \\
 & \quad + (g_{l-1/2, t} + g_{l+1/2, t} + g_{l, t-1/2} + g_{l, t+1/2}) p_{lt} = f_{lt}^1
 \end{aligned}$$

для всех $\{l, t\} \in M$. (3.13)

Как видно из полученных уравнений, матрицы коэффициентов для (3.3), (3.9), (3.12), (3.13) таковы, что положительными являются только элементы главной диагонали, а все остальные элементы — отрицательные или нулевые. Таким образом, объекты, описываемые уравнениями Лапласа, Пуассона, типа Лапласа, типа Пуассона после дискретизации могут быть представлены в виде системы линейных алгебраических уравнений, содержащей S -матрицу. Учет граничных условий 1-го и 2-го рода также не изменит специфику знаков элементов получаемой матрицы.

Интересно рассмотреть, в каком наиболее общем случае дифференциальное уравнение в частных производных сводится после конечно-разностной аппроксимации к системе линейных алгебраических уравнений с S -матрицей. (Мы не останавливаемся здесь на множестве проблем, связанных с использованием конечно-разностной аппроксимации. Нас интересует только специальная знаковая структура получаемой матрицы.) Пусть имеется линейное дифференциальное уравнение с частными производными общего вида

$$a \frac{\partial^2 p}{\partial x^2} + 2b \frac{\partial^2 p}{\partial x \partial y} + c \frac{\partial^2 p}{\partial y^2} + d \frac{\partial p}{\partial x} + e \frac{\partial p}{\partial y} + qp = f \quad (3.14)$$

в двумерной области R с координатами x, y , где a, b, c, d, e, q, f — функции x, y . Учитывая дискретизацию (3.6), а также замену

$$\left(\frac{\partial p}{\partial x \partial y} \right) \approx \frac{p_{l+1, t+1} - p_{l-1, t+1} - p_{l+1, t-1} + p_{l-1, t-1}}{(2h)^2},$$

мы можем записать уравнение (3.14) в разностном виде:

$$\begin{aligned} & (-2a - 2c + gh^2) p_{l, t} + \left(a + \frac{h}{2} d \right) p_{l+1, t} + \\ & + \left(a - \frac{h}{2} d \right) p_{l-1, t} + \left(c + \frac{h}{2} e \right) p_{l, t+1} + \left(c - \frac{h}{2} e \right) p_{l, t-1} + \\ & + \frac{1}{2} b p_{l+1, t+1} - \frac{1}{2} b p_{l-1, t+1} - \frac{1}{2} b p_{l+1, t-1} + \\ & + \frac{1}{2} b p_{l-1, t-1} = fh^2 \quad \text{для всех } \{l, t\} \in M. \end{aligned} \quad (3.15)$$

Значения a, b, c, d, e, q, f также берутся в точке $\{l, t\}$. Нас интересует, каким условиям должны удовлетворять коэффициенты при переменных p_{ij} в (3.15), чтобы матрица данной системы линейных алгебраических уравнений являлась S -матрицей. Из (3.15) очевидно, что это будет в том случае [7], если: 1) h достаточно мало; 2) $b=0$; 3) знаки при a и c одинаковые. Последнее условие выполняется, если уравнение (3.14) является уравнением эллиптического типа, для которого $ac > 0$.

Таким образом, в общем случае дифференциальное уравнение в частных производных может быть после конечно-разностной аппроксимации сведено к системе линейных алгебраических уравнений с S -матрицей, если это уравнение

эллиптического типа и не имеется члена с $\partial^2 p / \partial x \partial u$. При учете граничных условий 1-го и 2-го рода S -матрица в системе сохраняется [7].

2. **Постановка задачи оптимизации.** При заданных граничных условиях может быть получено решение рассматриваемых уравнений. Задача приобретает вариационный характер в том случае, когда имеется возможность выбирать граничные условия из некоторого множества, и, в зависимости от этого выбора, решение указанных уравнений будет давать то или иное значение критерия качества. Задача тогда будет заключаться в выборе оптимальных граничных условий. Варьироваться при оптимизации могут также некоторые параметры самого уравнения, например, функция стока f в уравнении Пуассона.

Сформулируем постановку задачи оптимизации. Пусть, записав в конечных разностях уравнения объекта и граничные условия, мы получим следующую систему линейных алгебраических уравнений:

$$GP = F, \quad (3.16)$$

где G — S -матрица *), P — вектор переменных, часть из которых управляется с помощью задания граничных условий 1-го рода, F — вектор переменных правой части. Кроме уравнений объекта (3.16), в задаче оптимизации могут быть различные ограничения на переменные. Если учитывать, например, ограничения на P типа $0 \leq P \leq P^{np}$ и ограничения на F типа $F \geq 0$ или $F \leq F^{np}$, то при линейной целевой функции задача оптимизации в прямой или в двойственной форме сводится к S -задаче линейного программирования:

$$CX \rightarrow \max, \quad AX \leq B, \quad X \geq 0, \quad (3.17)$$

где A — S -матрица, $B \geq 0$, X — вектор переменных, определяющих искомые переменные P, F . К виду (3.17) сводятся, естественно, и те задачи оптимизации, ограничения в которых можно рассматривать как частный случай нами приведенных, а также задачи с ограничениями $0 \leq F \leq F^{np}$, $P \geq 0$.

3. **Пример.** Рассмотрим в качестве примера уже

*) Заметим, что размерность рассматриваемой матрицы иногда удается значительно сократить, выразив часть переменных через управляемые переменные. Нетрудно показать, что получаемая при этом матрица также является S -матрицей, если решение задачи существует.

известную нам задачу оптимизации режима нефтяного пласта, поскольку данный объект описывается интересующими нас уравнениями. Пусть имеется пласт произвольной формы, дренируемый n эксплуатационными скважинами. Для поддержания высокого пластового давления производится нагнетание в пласт воды или газа, благодаря чему на контуре питания пласта поддерживается некоторое контурное давление p_k . Установившийся процесс фильтрации однородной жидкости в недеформируемой неоднородной по проницаемости пористой среде описывается уравнением (3.4) или (3.11). Для однородного по проницаемости пласта описание процесса фильтрации сводится к уравнениям (3.1) или (3.10). В данном случае $p(x, y)$ — давления в различных точках пласта, $g(x, y)$ — гидравлическая проводимость среды, $f(x, y)$ — функция источников (или стоков). Функция $f(x, y)$ в рассматриваемом случае будет равна нулю во всех

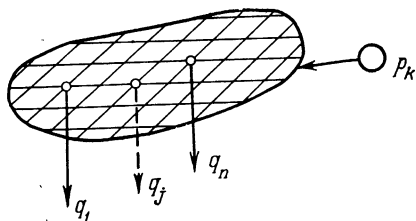


Рис. 1.7. Аппроксимация нефтяного пласта сеткой.

точках пласта, кроме точек расположения скважин. Будем рассматривать далее уравнение (3.11) как наиболее общее.

После конечно-разностной аппроксимации пласта сеткой, состоящей из N внутренних узлов, и задания для граничных узлов давлений p_k (рис. 1.7), получим систему N линейных алгебраических уравнений (3.13), которую запишем в виде

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} = \begin{bmatrix} -Q + P_1^k \\ P_2^k \end{bmatrix}, \quad (3.18)$$

где P_1 — n -мерный вектор-столбец давлений в тех узлах сетки, в которых расположены скважины; P_2 — $(N - n)$ -мерный вектор-столбец давлений в остальных внутренних узлах сетки; A_{11} , A_{12} — матрицы размера соответственно $(n \times n)$ и $n \times (N - n)$, состоящие из коэффициентов левой части уравнений (3.13) для узлов сетки, в которых расположены скважины; A_{21} , A_{22} — матрицы размера соответственно $(N - n) \times n$ и $(N - n) \times (N - n)$, состоящие из коэффициентов левой части уравнений (3.13) для остальных узлов сетки;

Q — n -мерный вектор-столбец дебитов скважин (значения функции f_{it}^I в точках расположения скважин); P_1^k, P_2^k —векторы размера соответственно n и $N - n$, элементы которых получены после переноса в правую часть уравнений (3.13) членов $g \cdot p$ с заданными значениями давлений $p = p_k$ для граничных узлов сетки. В целом матрица левой части условий (3.18), по аналогии с (3.16), может быть обозначена через G , а вектор правой части — через F^*).

Систему (3.18) удобнее переписать, взяв в качестве переменных не давления p_{ij} , а депрессии $\delta p_{ij} = p_k - p_{ij}$. Алгебраически для этого достаточно, например, прибавить и вычесть из каждого уравнения (3.13) член

$$(g_{l-1/2, t} + g_{l+1/2, t} + g_{l, t-1/2} + g_{l, t+1/2}) p_k.$$

Тогда получим

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \delta P_1 \\ \delta P_2 \end{bmatrix} = \begin{bmatrix} Q \\ 0 \end{bmatrix}, \quad (3.19)$$

где 0 — нулевой вектор. Система уравнений (3.19) связывает депрессии всех N узлов сетки. Однако, практически, вносить управляющие воздействия в режим работы пласта можно только через те n точек, в которых пробурены скважины. Ограничения также накладываются только относительно δP_1 и Q . Поэтому нас интересует не вся система уравнений (3.19), а только подсистема, характеризующая взаимосвязь дебитов скважин Q и депрессий δP_1 . Для получения этой подсистемы нужно в уравнениях $A_{11}\delta P_1 + A_{12}\delta P_2 = Q$ выразить значение δP_2 через значение δP_1 из условия $A_{21}\delta P_1 + A_{22}\delta P_2 = 0$. Тогда

$$A\delta P = Q, \quad (3.20)$$

где

$$A = A_{11} - A_{12}A_{22}^{-1}A_{21} \quad (3.21)$$

(по обобщенному алгоритму Гаусса), $\delta P = \delta P_1$. Матрица A коэффициентов взаимовлияния скважин по дебиту яв-

*) Заметим, что при написании конечно-разностной системы уравнений (3.18) следует учитывать несоизмеримость внутреннего контура скважины с внешним контуром питания пласта [10]. Сведение внутреннего контура скважины к точке может быть проделано, например, путем изменения сопротивления в зоне скважины.

ляется S -матрицей. Это очевидно как из физических соображений, так и аналитически. Физические соображения основаны на том, что увеличение дебита (и, соответственно, депрессий) на одной скважине вызывает уменьшение (точнее, не увеличение) дебитов всех остальных скважин (см. § 2). Аналитически, утверждение, что A есть S -матрица, следует из (3.21). Действительно, здесь по построению A_{11} — S -матрица, $A_{12} \leq 0$, $A_{21} \leq 0$ *). Матрица $A_{22}^{-1} \geq 0$, если для задачи линейного программирования, поставленной с использованием всей исходной матрицы G , имеет место ограниченность целевой функции на рассматриваемом множестве (это доказывается в главе III). В задачах оптимизации нефтедобычи значение целевой функции на рассматриваемом множестве не может быть бесконечным. Поэтому $A_{22}^{-1} \geq 0$. Отсюда, элементы матрицы A_{11} могут получить только неположительные приращения при выполнении преобразования (3.21), т. е. матрица A является S -матрицей.

Важно заметить, что переход от исходной матрицы G к матрице A не является обязательным при постановке задачи оптимизации. Задача не потеряет смысл и будучи поставленной относительно исходной матрицы G , т. е. если в ней, кроме независимых переменных, будут присутствовать зависимые переменные, единственным образом определяемые через независимые переменные. Переход от G к A имеет смысл делать только в том случае, когда это позволяет значительно сократить объем запоминаемой информации. Целесообразность перехода определяется исходя из следующих соображений. Матрица A по размеру может быть значительно меньше, чем матрица G . Однако матрица G является редкой матрицей — каждая строка содержит не более пяти ненулевых элементов. Матрица A , в отличие от матрицы G , является плотной матрицей — количество ненулевых элементов в строке может быть произвольным (хотя, учитывая физические соображения, каждая скважина имеет ненулевые коэффициенты взаимовлияния не более, чем с 20—30 близлежащими скважинами). Кроме того, при использовании матрицы G можно не записывать в память машины все ненулевые элементы g_{ij} , а запрограммировать только

*) Обозначение $A \leq 0$ означает, что матрица A имеет все неположительные элементы (неположительная матрица). В свою очередь матрица $A \geq 0$ — неотрицательная матрица [19].

зависимость $g(x, y)$ этих элементов от пространственных координат x, y . Тогда элементы g_{ij} будут определяться в явном виде только по мере надобности, а запись матрицы G будет значительно компактнее.

Особенно большие преимущества имеет использование матрицы G перед матрицей A в том случае, когда для решения задачи используются методы (например, итерационные), позволяющие не преобразовывать элементы исходной матрицы в процессе решения. (О таких методах речь пойдет далее — в третьей главе книги.)

Итак, пусть мы имеем уравнения объекта, записанные в виде (3.20) (случай использования матрицы G будем рассматривать как частный случай использования матрицы A при $n=N$, считая, что интенсивности стоков равны нулю в тех точках, в которых не пробурены скважины). Основные ограничения задачи имеют вид

$$\delta P \leq \delta P^{\text{пр}} \quad (3.22)$$

(давление p_i на забое i -й скважины не может быть понижено ниже некоторого предельного уровня $p_i^{\text{пр}}$; $\delta p_i = p_k - p_i$;

$$Q \geq 0 \quad (3.23)$$

(неотрицательность дебитов; неотрицательность депрессий при (3.22) выполняется автоматически: $\delta P = A^0 Q \geq 0$, так как $A^0 \geq 0$, $Q \geq 0$). Целевая функция имеет вид

$$C^0 Q \rightarrow \max. \quad (3.24)$$

Рассмотренная задача (3.20) — (3.24) может быть сведена к S -задаче. Заметим, что в предыдущем параграфе мы уже описывали сведение данной задачи к S -задаче для случая, когда она была сформулирована на основе матрицы A^0 коэффициентов взаимовлияния по депрессиям: $A^0 = A^{-1}$. Здесь мы покажем аналогичное сведение для задачи, сформулированной на основе матрицы A , что имеет место после конечно-разностной аппроксимации уравнений Пуассона.

Прежде всего исключим из рассмотрения n зависимых переменных, подставив Q из условий (3.20) в (3.23) и (3.24). Тогда получим

$$C^0(A\delta P) \rightarrow \max \text{ при } \delta P \leq \delta P^{\text{пр}}, A\delta P \geq 0. \quad (3.25)$$

В (3.25) произведем замену переменных δP на $\delta P = \delta P^{\text{пр}} - X$, где X — вектор дополнительных переменных,

обращающих условия (3.22) в равенства. После проведения простых алгебраических преобразований получим S -задачу

$$CX \rightarrow \max \text{ при } AX \leq B, X \geq 0, \quad (3.26)$$

где $C = -C^0 A$, $B = A \delta P^{\text{пр}} \geq 0$.

В данном параграфе мы рассмотрели уравнения Лапласа, Пуассона, типа Лапласа, типа Пуассона и могли видеть, что после конечно-разностной аппроксимации данные уравнения сводятся к системе алгебраических уравнений с S -матрицей. Далее, мы рассмотрели общие условия, при выполнении которых дифференциальное уравнение в частных производных сводится после дискретизации к системе с S -матрицей. Затем была описана постановка задачи оптимизации и S -задача линейного программирования, возникающая при этом.

§ 4. Регулирование мощности в системе радиостанций

Этот пример объекта представляет значительный интерес в связи с тем, что он в некотором роде являлся объектом исследования в качестве прообраза поведения коллектива автоматов [8]. На этом примере мы покажем, что выбор оптимального режима каждого самостоятельного элемента может быть произведен только при одновременном учете всех элементов системы в их взаимосвязи.

В качестве объекта рассматривается n пар радиопередающих и радиоприемных устройств. Отдельно приемно-передающее устройство будем называть элементом системы. Передающее устройство транслирует полезный сигнал на свое приемное устройство и создает помеху на все другие элементы системы. Рассмотрим математическую модель изучаемой системы. Допустим, что i -й передатчик для передачи информации i -му приемнику развивает мощность ε_i . При этом полагаем, что количество информации, которая в данном случае будет полезной, пропорционально мощности генератора; коэффициент пропорциональности обозначим через a_{ii} . Кроме того, будем полагать, что помеха от j -го элемента на i -й пропорциональна мощности j -го передатчика с коэффициентом пропорциональности a_{ij} . Все коэффициенты будем считать на заданном промежутке времени постоянными, положительными величинами, не зависящими от мощностей ε_j , $j=1, \dots, n$. Кроме того, будем считать, что на каждый приемник действует своя аддитивная помеха N_i , $i=1, \dots, n$.

где

$$E = [\varepsilon_1, \dots, \varepsilon_n], \quad N = [N_1, \dots, N_n]$$

и

$$M^n(\Lambda) = \begin{bmatrix} \lambda_1 a_{11} & -a_{12} & \dots & -a_{1n} \\ -a_{21} & \lambda_2 a_{22} & \dots & -a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \dots & \dots & \dots & \dots \\ -a_{i1} & \dots & \lambda_i a_{ii} & \dots & -a_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -a_{n1} & -a_{n2} & \dots & \lambda_n a_{nn} \end{bmatrix}. \quad (4.4)$$

Матрица (4.4) является S -матрицей, и мы можем рассматривать данную систему как объект многосвязного управления с прямыми отрицательными связями со структурой, приведенной на рис. 1.3. Задачу оптимизации можно сформулировать следующим образом: требуется определить вектор $E = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]$, минимизирующий в момент $t = t_0$ общую мощность всех генерирующих станций при выполнении заданного отношения шум/помеха: $\lambda_i \leq \lambda_i^{\text{задан}}$ и заданных пределов мощности передатчиков d_j^1 и d_j^2 . Математически задача запишется в следующей форме: минимизировать

$$L = \sum_{j=1}^n c_j \varepsilon_j \quad (4.5)$$

при выполнении ограничений

$$\left. \begin{aligned} \lambda_i^{\text{задан}} a_{ii} \varepsilon_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \varepsilon_j &\geq N_i, \\ d_j^1 &\leq \varepsilon_j \leq d_j^2, \end{aligned} \right\} \quad (4.6)$$

где λ_i, a_{ij}, N_i — заданные числа; $c_j \geq 0, \lambda_i^{\text{задан}} \geq 0, N_i \geq 0, a_{ij} \geq 0; i = 1, \dots, n; j = 1, \dots, n$. Покажем метод приведения задачи (4.5) и (4.6) к S -задаче. В матричном виде задача (4.5), (4.6) запишется в форме

$$CE \rightarrow \min \quad \text{при} \quad AE \geq N, \quad D^1 \leq E \leq D^2,$$

где A — S -матрица, состоящая из элементов $\lambda_i^{\text{задан}} \cdot a_{ii}$ (для $i=j$) и a_{ij} (для $i \neq j$). Произведя замену $E' = E - D^1$,

получим задачу с двусторонними ограничениями:

$$0 \leq E' \leq D',$$

где $D' = D^2 - D^1$. Перейдя теперь к двойственной задаче, получим

$$N'Y_1 - D'Y_2 \rightarrow \max \quad (4.7)$$

при

$$A^TY_1 - EY_2 \leq C, \quad Y_1 \geq 0, \quad Y_2 \geq 0.$$

Матрица A — S -матрица, E — единичная матрица, вектор $C \geq 0$, следовательно, задача (4.7) является S -задачей.

§ 5. Межотраслевой баланс

Рассмотрение межотраслевого баланса как многосвязного объекта оптимизации представляет большой интерес по крайней мере с двух точек зрения. Во-первых, сведение задачи к эквивалентному многосвязному объекту даст возможность использовать в полной мере все имеющиеся в теории управления методы и средства для наиболее важного экономического объекта *), и во-вторых, межотраслевой баланс, пожалуй, наиболее характерный объект высокой размерности. Эффективные методы решения задачи оптимизации межотраслевого баланса представляют самостоятельный большой интерес. Мы не будем вдаваться в подробности проблемы, а приведем только минимум сведений, необходимых для понимания постановки задачи.

Межотраслевым балансом производства и распределения продукции называют [98] систему таблиц, характеризующую производство и распределение продукции между отраслями, а также использование ресурсов в народном хозяйстве. Важность его состоит в том, что он вскрывает ключевые пропорции взаимодействия отраслей, позволяет определить величины, определяющие развитие экономики [67].

Для решения задачи согласования планов выпуска продукции отдельных отраслей составляется матрица межотраслевых связей «input — output» (затраты — выпуск), на основе которой затем проводится исследование. В данном

*) Вопросы оптимизации межотраслевого планирования явились первым применением методов линейного программирования в области экономики.

параграфе рассматривается линейная статическая модель межотраслевого планирования и указывается, что матрица межотраслевого баланса Леонтьева по структуре может быть отнесена к S -матрицам. Отмечается также то, что матрицы, используемые в ряде других экономических моделей, например, в моделях обмена, также являются S -матрицами. Далее записывается задача оптимизации межотраслевого планирования и показывается сведение ее к S -задаче линейного программирования.

1. О п и с а н и е м о д е л и. Допустим, что имеется n отраслей хозяйства, которые производят m видов продукции, причем каждая отрасль выпускает только один вид продукции и каждый продукт производится только в одной отрасли хозяйства (простая линейная модель производства, $n=m$). Пусть продукция i -й отрасли распределяется следующим образом:

$$x_i = \sum_{j=1}^n x_{ij} + b_i, \quad (5.1)$$

где x_i — продукция i -й отрасли хозяйства за рассматриваемый промежуток времени (например, за год), x_{ij} — поставки товаров i -й отраслью j -й отрасли (за год); b_i — конечный продукт, т. е. объем выпуска конечной продукции, запланированный i -й отраслью (за год).

Экономисты полагают возможным считать, что x_{ij} есть линейная однородная функция x_i . Это предположение и определяет линейный характер экономической модели межотраслевого баланса. Согласно принятому предположению

$$x_{ij} = a_{ij}x_i. \quad (5.2)$$

Коэффициент a_{ij} называют коэффициентом прямых затрат; он показывает, какое количество продукции отрасли i необходимо для производства единицы продукции отрасли j . Подставляя (5.2) в (5.1), получаем уравнение

$$\sum_{j=1}^n a_{ij}x_j + y_i = x_i, \quad i = 1, \dots, n. \quad (5.3)$$

Оно носит название *уравнения распределения продукции*

в межотраслевом балансе *). Систему уравнений (5.3) можно записать в форме

$$x_i - \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m. \quad (5.4)$$

Полученная система уравнений соответствует структуре многосвязного объекта с прямыми отрицательными связями. В матричной форме (5.4) запишется в виде

$$(E - A) X = B, \quad (5.5)$$

где E — единичная матрица. Матрица $(E - A)$ называется *матрицей межотраслевого баланса*, или *матрицей Леонтьева*. Так как $A \geq 0$, то матрица $(E - A)$ может быть отнесена по своей знаковой структуре к S -матрицам.

К S -матрицам относятся также матрицы, используемые в моделях обмена. Приведем один из примеров модели обмена [21]. Пусть имеется n стран $C_1, C_2, \dots, C_n$, торгующих друг с другом. Будем считать, что весь свой доход x_i i -я страна получает от продажи своих товаров либо внутри страны либо за рубежом. Обозначим через α_{ij} часть дохода x_j некоторой j -й страны, которая тратится на импорт из страны C_i . В линейных моделях α_{ij} является постоянной величиной, не зависящей от x_j . Экспорт из i -й страны в j -ю равен $\alpha_{ij} x_j$. Суммарный доход x_i для i -й страны можно подсчитать по формуле

$$x_i = \sum_{j=1}^n \alpha_{ij} x_j, \quad i = 1, 2, \dots, n, \quad (5.6)$$

где

$$\alpha_{ij} \geq 0, \quad \sum_{i=1}^n \alpha_{ij} = 1. \quad (5.7)$$

*) Коэффициенты прямых затрат обычно рассчитывают по формуле

$$a_{ij} = \frac{\sum_k a_{ij}^k x_j^k}{\sum_k x_j^k}.$$

Здесь коэффициент прямых затрат является средней арифметической нормой затрат на производство данного продукта [45].

Матрицу $A^0 = [\alpha_{ij}]$, элементы которой удовлетворяют условиям (5.7), называют *матрицей обмена* *). Выражение (5.6) в матричном виде запишется так:

$$X = A^0 X.$$

Очевидно, что матрица $(E - A^0)$ при векторе переменных X является S -матрицей.

2. П о с т а н о в к а з а д а ч и о п т и м и з а ц и и. Задача оптимизации межотраслевого планирования заключается в следующем [20, 92]. Полагая, что описание экономических процессов за данный промежуток времени остается справедливым и на следующий промежуток времени, требуется задать плановые объемы производства каждой отрасли таким образом, чтобы достигнуть экстремума некоторой целевой функции при удовлетворении имеющимся ограничениям, определяемым требованиями конечного спроса b_i и производственными возможностями отраслей. Если потребовать удовлетворения необходимых объемов конечной продукции каждой отрасли, то соответствующие ограничения запишутся с учетом (5.5) в виде

$$x_i - \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \dots, n. \quad (5.8)$$

Производственные возможности отраслей d_i определяют следующие ограничения на объемы выпускаемой продукции:

$$0 \leq x_i \leq d_i. \quad (5.9)$$

В качестве функции цели можно рассматривать минимизацию некоторого линейного функционала, имеющего смысл суммарных затрат:

$$L = \sum_{j=1}^n c'_j x_j, \quad (5.10)$$

где c'_j — себестоимость производства единицы продукции j -й отрасли. Задача оптимизации в этом случае запишется в виде (5.8) — (5.10). В работе [92] показано, что при выполнении естественных с экономической точки зрения

*) Подобные матрицы, которые в теории вероятностей называют *стохастическими*, широко применяются при описании марковских процессов. Ниже мы будем говорить о задачах марковского программирования, в которых также используются данные матрицы.

предпосылок:

$$\sum_{i=1}^n a_{ij} < 1, \quad j = 1, 2, \dots, n$$

и т.

$$\sum_{i=n}^n x_{ij} < x_j, \quad j = 1, 2, \dots, n,$$

решение задачи (5.8) — (5.10) находится в точке решения системы уравнений

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

если компоненты x_i^* решения этой системы удовлетворяют ограничениям (5.9). Если же компоненты x_i^* не удовлетворяют условиям (5.9), то это означает, что задача (5.8) — (5.10) не имеет решения.

Обычно на практике приходится сталкиваться с тем случаем, когда производственные ограничения нельзя не принимать во внимание. Тогда для того, чтобы получить решения задачи оптимизации межотраслевого планирования, необходимо расширить производственные возможности отраслей либо изменить формулировку задачи.

Мы рассмотрим случай, когда не представляется возможности расширить производственные мощности. В этом случае ограничения (5.9) ослабить не удастся. Остается только изменить ограничения (5.8). Например, можно потребовать, чтобы необходимые объемы конечной продукции удовлетворялись не для всех, а только для некоторых, особо важных отраслей (будем считать, что для этих отраслей $i = v_k$). В этом случае, естественно, не имеет смысла превышение необходимого объема конечной продукции по этим областям, т. е. не имеет смысла записывать условия (5.8) вида $\geq$. Достаточно записать ограничения по особо важным отраслям в виде равенств

$$x_i - \sum_{j=1}^n a_{ij} x_j = b_i, \quad i = v_k \quad (5.11)$$

и неравенства

$$x_i - \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i \neq v_k \quad (5.12)$$

для остальных отраслей. В качестве целевой функции при такой постановке задачи можно взять либо ущерб от недо-

получения требуемого объема конечной продукции [92, 93] либо условие максимизации суммарного дохода [20] (тогда коэффициенты c_j линейной формы будут означать доходы от производства единицы j -го продукта). В обоих случаях задача в конечном счете сводится к максимизации линейной формы

$$L = \sum_{j=1}^n c_j x_j \rightarrow \max \quad (5.13)$$

при ограничениях (5.9), (5.11), (5.12). Наличие условий-равенств (5.11) может привести к тому, что рассматриваемая задача, так же как и задача (5.8) — (5.10), может не иметь решения. Например, рассмотрим для иллюстрации такой случай. Пусть имеется некоторое k -е условие (5.11), где $b_k > 0$. Для выполнения этого условия необходимо иметь $x_k > 0$, так как иначе в левой части условия будет неположительное число. Минимальное значение $x_k > 0$ получается, если все остальные $x_j = 0$, $j \neq k$. Это минимальное значение будет следующим:

$$x_k^n = \frac{b_k}{1 - a_{kk}}.$$

Очевидно, что если k -е условие (5.9) не выполняется при x_k^n , то оно не будет выполняться ни при каком другом допустимом x_k , т. е. задача не имеет решения.

В то же время, как мы убедимся в дальнейшем (при рассмотрении свойств задач линейного программирования с S -матрицами), в оптимальной точке задачи при $C \geq 0$ для каждого i только одно из двух условий (5.9), (5.12) может обратиться в равенство. Иными словами, для каждого i может быть два случая: 1) производственных мощностей хватает для удовлетворения требованиям конечного спроса в i -м продукте (тогда i -е условие (5.9) выполняется как неравенство, а i -е условие (5.12) выполняется как равенство); 2) производственных мощностей не хватает для обеспечения требуемого объема i -го конечного продукта (тогда i -е условие (5.9) выполняется как равенство, а i -е условие (5.12) — как неравенство). Таким образом, если заменить систему условий (5.11), (5.12) условиями

$$x_i - \sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, \dots, n, \quad (5.14)$$

то автоматически устраняется случай, когда задача не имеет решения. В том случае, когда существующие производст-

венные мощности не могут обеспечить выполнение требований по объему некоторых конечных продуктов, задача будет иметь решение, но соответствующие условия (5.14) будут выполняться как строгие неравенства. Желание выполнить требования по необходимому объему конечного продукта для некоторых особо важных отраслей можно в некоторой степени отразить в целевой функции задачи с помощью введения больших весовых коэффициентов ущерба от недополучения конечного продукта в этих отраслях [92].

Таким образом, мы пришли к задаче [20] оптимизации межотраслевого планирования (5.9), (5.10), (5.14), которую называют иногда *статической моделью Леонтьева*. В матричной форме эта задача записывается в виде

$$L = CX' \rightarrow \max \quad (5.15)$$

при

$$(E - A)X \leq B, \quad (5.16)$$

$$X \leq D, \quad (5.17)$$

$$X \geq 0. \quad (5.18)$$

3. С в е д е н и е к S -задаче. Рассмотрим сведение задачи линейного программирования (5.15) — (5.18) к

$x_1 \quad x_2 \quad x_3 (\leq)$			S -задаче. Обычно $C \geq 0$. Тогда для приведения данной задачи к требуемому виду достаточно взять исходный псевдоплан в точке $x_j = d_j$ и решать задачу двойственной процедурой либо решать прямым алгоритмом типа симплексной задачи, двойственную к исходной, при начальном опорном плане $y_{i+n} = c_j (i=j)$, где y_{i+n} — переменные двойственной задачи, индексы которых соответствуют ограничениям (5.17).	
(5.16)	↑	$ E - A $		$ B$
		+	—	
	↓	—	+	
(5.17)	↑	$ E$		$ D$
		+		
	↓		+	
$\Delta = -C$		$-C$		L

Рис. 1.8. Знаковая структура исходной симплекс-таблицы для задачи (5.15) — (5.18).

(для $n=3$). Знаковая структура элементов исходной симплекс-таблицы для данной задачи имеет вид, изображенный на рис. 1.8. Перейдя к двойственной задаче, мы получим симплекс-таблицу, знаковая структура которой изображена

на рис. 1.9. Добавочные переменные y_d — это переменные, обращающие ограничения $\geq$ двойственной задачи в равенства. Оценки Δ равны коэффициентам целевой функции двойственной задачи с обратным знаком:

$$\Delta_1 = B^T, \quad \Delta_2 = D^T, \quad \Delta_d = 0$$

(рассматривается задача максимизации). Исходный опорный план удобно взять в точке $y_{i+n} = c_j$ ($i=j$). Знаковая структу-

Матрица $(E-A)^T$ условий (5.16)			Матрица E^T условий (5.17)			Матрица E при добавочных переменных			
y_1	y_2	y_3	y_4	y_5	y_6	y_d^1	y_d^2	$y_d^3 (=)$	
+	-	-	+			-			C
-	+	-		+			-		
-	-	+			+			-	
Δ_1			Δ_2			Δ_3			

Рис. 1.9. Знаковая структура симплекс-таблицы после сведения задачи (5.15) — (5.18) к требуемому виду.

ра элементов симплекс-таблицы остается при этом той же, что на рис. 1.9, а оценки имеют вид

$$\left. \begin{aligned} \Delta_1 &= B^T - D^T (E - A)^T, \\ \Delta_2 &= 0, \\ \Delta_d &= -D^T (-E) = D^T. \end{aligned} \right\} \quad (5.19)$$

Данная задача может быть записана следующим образом:

$$\left. \begin{aligned} & -\Delta_1 Y_1 - \Delta_d Y_d \rightarrow \max \\ \text{при} \quad & (E - A)^T Y_1 + E Y_2 - E Y_d = C, \\ & Y_1 \geq 0, Y_2 \geq 0, Y_d \geq 0. \end{aligned} \right\} \quad (5.20)$$

Задача (5.20) является S -задачей. Можно показать, что при $B \geq 0$ переменные Y_d в оптимальной точке всегда нулевые. Действительно, пусть мы начали решение с исходного опорного плана $y_{i+n} = c_j$, $i=j$ (т. е. $Y_2 = C$). Тогда исходная оценка $\Delta_d = D^T \geq 0$ (см. (5.19)). Если в процессе решения задачи оценка Δ_d останется не отрицательной, то это будет

означать, что переменные вектора Y_d не вводятся в число базисных переменных. Именно это нам и требуется показать. В дальнейшем, при рассмотрении алгоритмов решения S -задач, мы убедимся, что в процессе решения задачи (5.20) симплекс-методом вектор оценок Δ_d будет получать только отрицательные приращения за счет введения компонент вектора Y_1 в число базисных переменных.

Рассмотрим самый «худший» случай — т. е. случай, когда вектор Δ_d получает наибольшее отрицательное приращение. Это будет случай, когда все компоненты вектора Y_1 вводятся в число базисных. Тогда оценка Δ_d получит следующее приращение:

$$\Delta_d' = \Delta_d - \Delta_1 [(E - A)^T]^{-1} (-E)$$

или, учитывая (5.19),

$$\begin{aligned} \Delta_d' &= D^T + [B^T - D^T (E - A)^T] [(E - A)^T]^{-1} = \\ &= D^T + B^T [(E - A)^T]^{-1} - D^T = B^T [(E - A)^T]^{-1}. \end{aligned}$$

Матрица $(E - A)^T$ — матрица Леонтьева. Поэтому матрица $[(E - A)^T]^{-1}$ неотрицательна [40, 85]. Отсюда при $B^T \geq 0$ имеем $\Delta_d' \geq 0$, что и требовалось доказать.

Поясним физически, почему $y_d = 0$ при решении задачи (5.20). Элементы матрицы при переменных y_d в процессе решения задачи (5.20) будут получать, как и элементы вектора Δ_d , только неположительные приращения. Поэтому появление отрицательной компоненты вектора Δ_d означает неограниченность целевой функции двойственной задачи (5.20) на множестве ее планов или несовместность условий прямой задачи (5.15) — (5.18). Случай $B \geq 0$ — это один из случаев, когда условия задачи (5.15) — (5.18) совместны (при $B \geq 0$ очевиден исходный опорный план: $X = 0$, $X^d = \begin{bmatrix} B \\ D \end{bmatrix}$,

где X^d — вектор переменных, обращающих ограничения (5.16), (5.17) в равенства).

Поэтому при $B \geq 0$ не может быть случая неограниченности целевой функции двойственной задачи, $\Delta_d \geq 0$, и соответственно $Y_d = 0$. Полезно заметить также, что $Y_d = 0$ в задаче (5.20) означает $X^* > 0$ для прямой задачи (5.15) — (5.18). Иными словами, если условия задачи (5.15) — (5.18) совместны, то в оптимальной точке все компоненты вектора X^* являются базисными.

Итак, мы убедились, что при вполне естественной с экономической точки зрения предпосылке о неотрицательности вектора B компоненты вектора Y_d равны нулю и могут быть отброшены. Задача (5.20) в этом случае примет вид

$$-\Delta_1 Y_1 \rightarrow \max$$

при

$$(E-A)^T Y_1 + E Y_2 = C, \quad Y_1 \geq 0, Y_2 \geq 0, \quad (5.21)$$

где $\Delta_1 = B^T - D^T (E-A)^T$. В дальнейшем мы будем рассматривать задачи, в которых вектор $C \geq 0$. Это предпосылка обычно выполняется на практике.

§ 6. Электрические цепи

В данном параграфе показывается, что описание любой линейной электрической цепи постоянного тока приводит к S -матрице. Затем рассматривается пример постановки задачи оптимизации на электрической цепи.

1. Описание электрической цепи и S -матрицы. Прежде всего приведем некоторые сведения о расчете электрических цепей (см., например, [2, 53]). Структурно электрическая цепь состоит из трех основных типов элементов: ветвей, узлов и контуров. Соответственно уравнения электрической цепи распадаются на три основные группы: уравнения ветвей, уравнения узлов и уравнения контуров.

Уравнения ветвей (закон Ома):

$$i = j - yi \quad (6.1)$$

или

$$u = e - zi, \quad (6.2)$$

где i — ток ветви, u — напряжение ветви, j — интенсивность источника тока в ветви, e — э. д. с. источника напряжения, y, z — соответственно проводимость и сопротивление ветви.

Уравнения узлов (1-й закон Кирхгофа):

$$\sum_{s=1}^S i_s = 0, \quad (6.3)$$

где $s=1, 2, \dots, S$ — индексы ветвей, сходящихся в данном узле.

Уравнение контуров (2-й закон Кирхгофа):

$$\sum_{i=1}^L u_i = 0, \quad (6.4)$$

где $l=1, 2, \dots, L$ — индексы ветвей, относящихся к данному контуру. Пусть в цепи p ветвей, q узлов и k контуров. Если считать один узел за базисный, то число независимых узлов будет $q-1$. Число независимых контуров — соответственно $k=p-q+1$. Число уравнений (6.1) или (6.2) равно p , число уравнений (6.3) равно $q-1$, и число уравнений (6.4) равно $p-q+1$. Таким образом, $2p$ неизвестных u и i электрической цепи связаны $2p$ уравнениями (6.1) (или (6.2)), (6.3), (6.4). Однако для расчета электрической цепи не имеет смысла решать систему $2p$ линейных алгебраических уравнений. Достаточно определить из решения соответствующей системы только независимые переменные, через которые могут быть найдены все остальные токи или напряжения цепи.

Прежде всего сразу видно, что из решения системы уравнений (6.3), (6.4) можно найти p независимых токов или напряжений, подстановка которых в уравнения (6.1) или (6.2) однозначно определяет остальные p неизвестных. Например, можно решать систему (6.3), (6.4) только относительно токов i или только относительно напряжений u , а затем при известных токах определить напряжения из уравнений (6.1) или (6.2); если же известны напряжения, то из (6.1) или (6.2) могут быть определены токи.

Далее, систему p уравнений (6.3), (6.4) также не имеет смысла решать полностью. Достаточно решить систему меньшего порядка относительно некоторых независимых переменных, в качестве которых удобно брать узловые напряжения или контурные токи. Узловые напряжения — это напряжения в узлах цепи относительно некоторого базисного узла. Контурные токи — это условные токи контуров, принятые в предположении, что в каждом независимом контуре течет свой ток. Если в качестве переменных используются узловые напряжения, то расчет цепи строится не на основе уравнений (6.3), (6.4), а на основе $(q-1)$ уравнений (6.3). Физически это вполне понятно: если из решения

системы $(q-1)$ -го порядка определить узловые напряжения для всех $q-1$ независимых узлов, то напряжение каждой ветви легко находится как разность соответствующих узловых напряжений. Если же использовать в качестве переменных контурные токи, то вместо решения p уравнений (6.3), (6.4) расчет строится на основе $(p-q+1)$ уравнений (6.4). Определив токи $(p-q+1)$ независимых контуров, находят ток каждой ветви как сумму или разность соответствующих контурных токов.

Приведем описание линейной электрической цепи сначала относительно узловых напряжений, затем относительно контурных токов. Будем использовать матричные обозначения [53].

Пусть Π — матрица, показывающая, каким образом ветви электрической цепи соединяются в узлы. Каждому столбцу этой матрицы соответствует ветвь, каждой строке — узел. На пересечении столбца и строки ставится $+1$, если соответствующая ветвь входит в данный узел, ставится -1 , если эта ветвь выходит из данного узла и ставится 0 , если ветвь не принадлежит данному узлу. Вектор токов ветвей обозначим через I , вектор напряжений ветвей через U . Вектор источников тока в ветвях обозначим $J_{\text{в}}$. Матрицу собственных и взаимных проводимостей ветвей обозначим через $Y_{\text{в}}$. Для цепей постоянного тока матрица $Y_{\text{в}}$ — вещественная диагональная. В цепях переменного тока без индуктивных связей между ветвями матрица $Y_{\text{в}}$ комплексная, хотя и диагональная. Однако в предлагаемой работе нас интересуют только вещественные матрицы, поэтому цепи переменного тока нами не рассматриваются.

Учтя введенные матричные обозначения, запишем систему уравнений (6.1)

$$I = J_{\text{в}} - Y_{\text{в}} U \quad (6.5)$$

и систему (6.3)

$$\Pi I = 0. \quad (6.6)$$

Подставив (6.5) в (6.6) получим выражение для напряжений ветвей

$$(\Pi Y_{\text{в}}) U = \Pi J_{\text{в}}. \quad (6.7)$$

Перейдем от вектора напряжений ветвей U к вектору узловых напряжений V . Связь между векторами U и V очевидна из самого определения напряжения ветви как

разницы двух соответствующих узловых напряжений:

$$U = P^T V, \quad (6.8)$$

где P^T — транспонированная матрица P . Тогда, подставив (6.8) в (6.7), получим

$$(PY_B P^T) V = PJ_B. \quad (6.9)$$

Обозначим матрицу $(PY_B P^T)$ через Y , вектор PJ_B через J_Σ и запишем

$$YV = J_\Sigma. \quad (6.10)$$

Матрица Y — квадратная матрица собственных и взаимных проводимостей узлов электрической цепи размера $(q-1)$.

На практике матрица Y записывается обычно прямо по схеме электрической цепи. Элементы главной диагонали матрицы Y (собственные проводимости узлов Y_{ss}) определяются как сумма проводимостей ветвей, сходящихся в s -м узле. Взаимные проводимости узлов Y_{sw} , $s \neq w$, равны проводимостям ветвей между s -м и w -м узлами. Вектор J_Σ является вектором суммарных токов источников тока в ветвях, принадлежащих данному узлу. Каждая s -я компонента вектора J_Σ определяется как сумма токов короткого замыкания ветвей, сходящихся в s -м узле.

Приведем теперь аналогичный вывод уравнений, связывающих контурные токи электрической цепи. Пусть Γ — матрица, показывающая, каким образом ветви электрической цепи соединяются в контуры. Каждому столбцу этой матрицы соответствует ветвь, каждой строке — контур. На пересечении данного столбца и строки ставится 0, если соответствующая ветвь не принадлежит данному контуру; ставится +1 или -1, если соответствующая ветвь принадлежит данному контуру (+1 — когда направление ветви совпадает с направлением контурного тока, -1 — когда не совпадает). Вектор источников напряжения в ветвях обозначим через E_B , матрицу сопротивлений ветвей — через Z_B . Тогда уравнения (6.2) и (6.4) запишутся в виде

$$U = E_B - Z_B I, \quad (6.11)$$

$$\Gamma U = 0. \quad (6.12)$$

Подставив (6.11) в (6.12), получим

$$(\Gamma Z_B) I = \Gamma E_B. \quad (6.13)$$

Перейдем от вектора токов ветвей I к вектору контурных токов I_k . Этот переход очевиден, если учесть, что ток ветви определяется как сумма контурных токов, протекающих через рассматриваемую ветвь:

$$I = \Gamma^T I_k, \quad (6.14)$$

где Γ^T — транспонированная матрица Γ . Подставив (6.14) в (6.13), получим

$$(\Gamma Z_b \Gamma^T) I_k = \Gamma E_b. \quad (6.15)$$

Обозначим матрицу $(\Gamma Z_b \Gamma^T)$ через Z , вектор ΓE_b через E_Σ . Тогда

$$Z I_k = E_\Sigma. \quad (6.16)$$

Матрица Z — квадратная матрица собственных и взаимных сопротивлений контуров размера $k = p - q + 1$. Эта матрица обычно записывается прямо по схеме электрической цепи. Элементы главной диагонали матрицы Z (собственные сопротивления контуров Z_{kk}) определяются как сумма сопротивлений ветвей, принадлежащих k -му контуру. Взаимные сопротивления контуров Z_{kw} ($k \neq w$) равны сопротивлениям ветвей, общих k -му и w -му контурам. Вектор E_Σ — это вектор суммарных э. д. с. источников напряжений, действующих в контуре.

Описание вывода известных уравнений метода узловых напряжений (6.10) и метода контурных токов (6.16) понадобилось нам для того, чтобы получить возможность судить о структуре матриц Y и Z . Итак, мы имеем $Y = \Pi Y_b \Pi^T$ (см. уравнение (6.9)) и $Z = \Gamma Z_b \Gamma^T$ (см. уравнение (6.15)). Специальная структура матриц Y и Z основывается на специфике исходных матриц Π и Γ . Матрица Π характерна следующим:

1) каждый столбец содержит не более двух ненулевых элементов: $(+1)$ и (-1) ;

2) две различные строки имеют не более чем по одному ненулевому элементу, принадлежащему одному и тому же столбцу. Физически первое свойство очевидно: каждая ветвь может соединять не более двух узлов, причем из одного узла она выходит (-1) , а в другой узел входит $(+1)$. Второе свойство матрицы Π соответствует тому, что два неустраиваемых

узла электрической цепи могут быть соединены только одной ветвью *).

Умножение матрицы $\mathbf{P}$ справа на диагональную матрицу $\mathbf{Y}_v$ эквивалентно умножению элементов каждого j -го столбца матрицы $\mathbf{P}$ на j -й диагональный элемент матрицы $\mathbf{Y}_v$. Умножение матрицы $\mathbf{PY}_v$ справа на матрицу $\mathbf{P}^T$ всегда приводит к квадратной симметричной матрице, у которой все элементы главной диагонали положительны, а все остальные элементы — неположительны.

Это следует из специфических свойств матрицы $\mathbf{P}$. Действительно, при перемножении и последующем сложении элементов двух строк матриц $\mathbf{PY}_v$ и $\mathbf{P}$ (что и делается при определении $(\mathbf{PY}_v)\mathbf{P}^T$) вследствие свойств матрицы $\mathbf{P}$ получается неположительное число: $-y_i \cdot 1$ или $+y_i \cdot (-1)$, если рассматриваются строки с различными номерами; положительное число $\sum_{s=1}^S y_s$ получается, если рассматриваются строки матриц $\mathbf{PY}_v$ и $\mathbf{P}$ с одинаковыми номерами (здесь $s=1, \dots, S$ — индексы ненулевых элементов рассматриваемой строки; физически это номера ветвей, входящих в рассматриваемый узел). Отсюда видно также, что матрица $\mathbf{Y}=(\mathbf{PY}_v) \cdot \mathbf{P}^T$ является симметричной и удовлетворяет ослабленным условиям Адамара [19]. Таким образом, мы убедились, что матрица $\mathbf{Y}$, используемая при описании электрической цепи через узловые напряжения, является S -матрицей.

Аналогично можно показать, что матрица $\mathbf{Z}$, используемая при описании электрической цепи через контурные токи, также является S -матрицей, если исходная матрица (аналогично матрице $\mathbf{P}$) обладает двумя специальными свойствами, отмеченными выше. Интересующие нас свойства присущи матрице $\mathbf{G}$ для любой планарной электрической цепи, если

*) *Устранимым узлом* называют узел, в котором соединяются только две ветви (последовательное соединение). *Устранимым контуром* называют контур, которому принадлежат только две ветви (параллельное соединение). Устранимыми подобные узлы и контуры называют потому, что они могут быть легко удалены из электрической цепи эквивалентным пересчетом соответствующих проводимостей и источников. Если рассчитывается электрическая цепь, состоящая только из неустраимых узлов и контуров, то токи и напряжения устранимых ветвей легко находятся по формулам последовательного или параллельного соединений. Поэтому перед расчетом сложной электрической цепи из нее следует удалить все устранимые узлы и контуры и составлять матрицы $\mathbf{P}$ и $\mathbf{G}$ только для неустраимых узлов и контуров.

направления контурных токов выбраны по часовой стрелке *). Действительно, в планарной электрической цепи каждая ветвь может входить не более, чем в два контура, т. е. каждый столбец содержит не более двух ненулевых элементов. Если же выбрать направления всех контурных токов по часовой стрелке, то направление ветви, общей двум контурам, будет совпадать с направлением одного контурного тока, и будет противоположно направлению другого контурного тока. Это означает, что одним из двух ненулевых элементов столбца матрицы Γ будет $+1$, другим -1 . Отсюда ясно выполнение свойства 1. Второе свойство в матрице Γ соответствует тому, что два неустраимых контура электрической цепи могут иметь не более чем одну общую ветвь. Это свойство для матрицы Γ очевидно.

2. Постановка задачи оптимизации. При заданных параметрах электрической цепи (т. е. при заданных токах источников J_v и проводимостях Y) из решения системы линейных алгебраических уравнений (6.10) однозначно определяются U , а затем и все интересующие нас токи и напряжения цепи. Задача приобретает вариационный характер в том случае, когда задаваемые величины могут выбираться из некоторого множества, и в зависимости от этого выбора после расчета электрической цепи будет получено то или иное значение некоторой функции качества. В данных условиях естественна задача оптимального выбора задаваемых параметров электрической цепи.

Рассмотрим постановку одной задачи оптимизации на электрических цепях. Пусть имеется цепь, составленная из проводимостей и источников тока. (Если в цепи имеются источники напряжения, то их для данной задачи следует заменить источниками тока.) Далее, пусть значения токов источников тока могут варьироваться

$$J_v \geq 0, \quad (6.17)$$

но при этом должны выполняться некоторые ограничения на переменные, например, ограничения

$$V \leq V^{\text{пр}}, \quad (6.18)$$

где $V^{\text{пр}}$ — предельно допустимое значение V . Каждому

*) Естественно, имеется в виду матрица Γ , составленная после удаления из электрической цепи устранимых узлов и контуров.

варианту выбора значений J_v ставится в соответствие некоторое значение целевой функции. Если эта функция цели линейна и ограничения тоже линейны, то задача может быть сформулирована в терминах линейного программирования. Примером подобных задач может служить описанная выше задача оптимизации нефтедобычи, сформулированная в терминах электрических цепей. В данном случае, в соответствии с заменой уравнений типа Пуассона (см. § 3) электрической сеткой, составленной из активных сопротивлений и источников тока, получится следующее. Дебиты скважин Q заменяются токами источников J_v (J_v здесь равно J_Σ). Депрессии δP относительно давлений на контуре заменяются узловыми напряжениями V относительно базового узла (контура). Тогда системы (3.19) и (6.10) эквивалентны. Ограничения (3.22) и (6.18), (3.23) и (6.17) также эквивалентны. Таким образом, данная задача оптимизации электрической цепи может быть сформулирована в виде задачи линейного программирования (3.19), (3.22) — (3.24) и сведена к S -задаче. Аналогичная задача оптимизации электрической цепи может быть сформулирована и при использовании описания (6.16).

В заключение приведем следующее замечание общего характера. Электрическую цепь можно рассматривать как некоторую сеть. Описание сети нетрудно свести к виду, использующему S -матрицу (например, если использовать в описании матрицу PPP^*). Приведенную задачу можно рассматривать как довольно общую задачу определения оптимальных интенсивностей источников и стоков в сетях, имеющих единственное распределение потоков.

ПРИМЕРЫ НЕЛИНЕЙНЫХ И НЕКОТОРЫХ ДИНАМИЧЕСКИХ
ЗАДАЧ ОПТИМИЗАЦИИ

В предыдущей главе мы рассмотрели целый ряд примеров линейных многосвязных объектов и показали, что все эти объекты используют в описании специальную матрицу, названную нами *S-матрицей*. Задачи оптимизации для данных объектов сводились к задаче линейного программирования с *S-матрицей* ограничений (*S-задача*).

В данной главе мы рассматриваем примеры некоторых нелинейных и динамических объектов, использующих в описании матрицу, состоящую из серии *S-матриц*. Подобную матрицу мы назвали во введении *S^k-матрицей*. Если *S-матрицу* мы определяли как матрицу, каждая строка и каждый столбец которой содержат не более одного положительного элемента, то *S^k-матрицу* можно определить, как матрицу, каждая строка или каждый столбец которой содержат не более одного положительного элемента, а остальные элементы, как и в первом случае, неположительные. Задачи оптимизации для данных объектов будут сводиться к задаче линейного программирования с *S^k-матрицей* ограничений (*S^k-задача*).

Изложение в данной главе построено следующим образом. Сначала в § 1 рассматриваются некоторые нелинейные задачи оптимизации многосвязных систем и ряд сходных задач, в которых *S-матрица* используется неоднократно. Затем в § 2 подробно рассматривается наиболее важный класс задач, использующих *S^k-матрицу*, — задачи марковского программирования.

§ 1. Некоторые нелинейные задачи оптимизации многосвязных систем и *S^k-задачи*

Из описания статического режима многосвязного объекта (см. уравнения (2.1) — (2.5) главы I) видно, что в зависимости от типа коэффициентов связи между регулируемыми

величинами α_{ij} многосвязные объекты могут быть линейными (при $\alpha_{ij} = \text{const}$) или нелинейными (при $\alpha_{ij} = f(xy)$). Особенно важно учитывать нелинейность членов главной диагонали матрицы, описывающей многосвязный объект, так как эти члены обычно превосходят остальные элементы матрицы («самовлияние» больше «взаимовлияния»). Учет указанных нелинейностей при оптимизации многосвязных объектов приводит часто к следующей задаче выпуклого программирования:

$$\sum_{j=1}^n c_j x_j \rightarrow \max, \quad (1.1)$$

$$\sum_{j=1}^n a_{ij} x_j + \varphi_i(x_i) \leq b_i, \quad i = 1, 2, \dots, m_1, \quad (1.2)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = m_1 + 1, \dots, m, \quad (1.3)$$

$$0 \leq x_j \leq \beta_j, \quad j = 1, 2, \dots, n, \quad (1.4)$$

где $\varphi_i(x_i)$ выпуклые функции, $A = [a_{ij}]$ — S -матрица. Подобные задачи возникают, например, при оптимизации нефтяных месторождений с фонтанными скважинами. Можно провести кусочно-линейную аппроксимацию функций $\varphi_i(x_i)$ таким образом, что вместо каждого i -го ограничения системы (1.2) получим k_i условий вида

$$\sum_{j=1}^n a_{ij} x_j - \alpha_i^k x_i \leq b_i - p_i^k, \quad k = 1, 2, \dots, k_i, \quad (1.5)$$

где α_i^k — угловой коэффициент прямой, линеаризующей $\varphi_i(x_i)$ на k -м участке оси x_i ; p_i^k — отрезок, отсекаемый k -й линеаризующей прямой на оси $\varphi_i(x_i)$; k_i — число отрезков, на которые разбивается $\varphi_i(x_i)$ при кусочно-линейной аппроксимации. Нетрудно показать, что аппроксимированная задача эквивалентна исходной. Для этого достаточно убедиться в том, что на каждом t -м участке функции $\varphi_i(x_i)$ t -е условие (1.5) является наиболее «узким»:

$$-p_i^t + \alpha_i^t x_i \leq -p_i^k + \alpha_i^k x_i, \quad k = 1, 2, \dots, k_i, \quad k \neq t.$$

Это непосредственно следует из выпуклости функции $\varphi_i(x_i)$.

Отсюда ограничение системы (1.2) на каждом t -м участке аппроксимируется t -м ограничением (1.5), а все остальные ограничения (1.5) $k \neq t$ на этом участке оказываются лишними, что и доказывает эквивалентность между аппроксимированной и исходной задачами. Если составлять $(m \times n)$ -мерные матрицы A_i , в которые войдет по одному условию из каждой i -й системы (1.5) или (1.3), то исходная задача в матричной форме запишется так:

$$\left. \begin{aligned} CX &\rightarrow \max, \\ A_k X &\leq B_i, \quad k = 1, 2, \dots, k_i^m, \quad 0 \leq X \leq \beta, \end{aligned} \right\} \quad (1.6)$$

где A_k — S -матрица, k_i^m — максимальное число k_i . Матрица ограничений полученной задачи линейного программирования состоит из нескольких S -матриц. По аналогии с рассмотренными в первой главе задачами, мы называли подобные задачи S^k -задачами. S^k -задачи возникают не только при оптимизации многосвязных систем. К подобным задачам сводятся задачи марковского программирования, которые мы подробно рассмотрим в последующих параграфах. К S^k -задачам сводятся также многие задачи оптимизации объектов, рассмотренных в предыдущей главе, в том случае если S -матрица, описывающая объект, используется в задаче неоднократно. Рассмотрим в качестве примера модель межотраслевого баланса.

Обобщенная модель Леонтьева. В предыдущей главе, при рассмотрении модели межотраслевого планирования, предполагалось, что каждый технологический процесс может производить только один продукт, и каждый продукт может производиться только одним технологическим процессом. Такая модель называется *простой моделью Леонтьева*. В обобщенной модели Леонтьева [21] второе предположение снимается.

Пусть i -й продукт может производиться k_i технологическими процессами. Пусть, далее, x_i^k — интенсивность использования k -го процесса при производстве i -го продукта, а α_{ij}^k — затраты i -го продукта на производство единицы j -го продукта при использовании k -го процесса. Тогда выпуск i -го продукта в единицу времени

$$y_i = \sum_{k=1}^{k_i} x_i^k - \sum_{j=1}^n \sum_{k=1}^{k_j} \alpha_{ij}^k x_{ij}^k, \quad i = 1, 2, \dots, n. \quad (1.7)$$

Матрица системы (1.7) состоит из нескольких S -матриц, т. е. является S^k -матрицей.

Рассмотрение обобщенной модели Леонтьева в свете S^k -задач может оказаться весьма полезным не только в алгоритмическом отношении. Это позволяет, например, с иных позиций взглянуть на известную теорему замещения Самуэльсона. Доказательство данной теоремы в этом случае легко может быть получено как следствие некоторых свойств S^k -задач, о которых мы будем говорить в главе IV.

§ 2. Марковское программирование

В ряде задач динамической оптимизации используются матрицы требуемого вида. Подобной структурой матриц обладают, например, задачи динамической оптимизации систем, поведение которых описывается управляемым марковским процессом. Благодаря аналогии в структуре матриц эти задачи примыкают к рассмотренным в предыдущих параграфах задачам статической оптимизации многосвязных систем и могут быть использованы при динамической оптимизации соответствующих многосвязных систем.

Класс объектов, которые можно рассматривать в терминах марковских процессов с доходами, весьма широк. Это и многие динамические задачи выработки стратегии поведения в условиях неопределенности, некоторые задачи массового обслуживания и т. д. Подобные задачи оптимизации марковских процессов с доходами получили название *задач марковского программирования*. Ниже дается описание постановок и примеров задач оптимизации марковских процессов с доходами, рассматриваемых с точки зрения линейного программирования. В данном параграфе приводятся некоторые определения, необходимые для понимания последующего изложения. Рассмотрению различных постановок задач оптимизации марковских процессов с доходами посвящены §§ 3 и 4. В § 5 описываются примеры конкретных задач марковского программирования. В заключение приводится краткое обсуждение.

Некоторые определения. Процесс, протекающий в физической системе, называется *марковским*, если для каждого момента времени вероятность любого состояния системы в будущем зависит только от состояния системы в

настоящий момент времени и не зависит от того, каким образом система пришла в это состояние.

Марковский процесс характеризуется *переходной вероятностью* $p_{ij}(t_1, t_2)$, т. е. вероятностью того, что если в момент времени t_1 система находится в состоянии i , то в некоторый последующий момент времени t_2 система окажется в состоянии j . Марковский процесс может иметь как конечное, так и бесконечное число состояний. В дальнейшем мы будем рассматривать только процессы с конечным числом состояний. В примерах с бесконечным числом состояний проводится дискретизация числа состояний процесса с тем, чтобы свести анализ к случаю конечного числа состояний. Марковский процесс может быть как дискретным, так и непрерывным во времени. Будем в дальнейшем рассматривать процессы с дискретным временем (марковские цепи), однако приводимый подход может использоваться [87] и для рассмотрения процессов с непрерывным временем.

Однородность. Марковский процесс называется *однородным*, если вероятности $p_{ij}(t_1, t_2)$ зависят только от разности $t = t_1 - t_2$, т. е. $p_{ij}(t_1, t_2) = p_{ij}(t)$.

Для марковского процесса с дискретным временем переходная вероятность обозначается $p_{ij}(n_1, n_2)$, где n_1 и n_2 — количество переходов от начала процесса. Для однородного марковского процесса с дискретным временем $p_{ij}(n_1, n_2) = p_{ij}(n)$, где $n = n_1 - n_2$. Если в качестве n рассматривать один шаг, то можно сказать, что однородный марковский процесс — это процесс, для которого вероятность $p_{ij}(1)$ перехода из i -го состояния в j -е за один шаг не изменяется по шагам. Вероятность $p_{ij}(1)$ будем в дальнейшем обозначать p_{ij} .

Стационарность. Если вероятности $p_{ij}(n)$ не меняются при изменении n , то процесс является *стационарным*. Вероятности переходов в этом случае называются *предельными*.

Эргодичность. Если предельная вероятность π_j попадания в j -е состояние после достаточно большого числа переходов не зависит от исходного состояния i , то процесс называется *эргодическим*. В этом случае существует предел

$$\lim_{n \rightarrow \infty} p_{ij}(n) = \pi_j.$$

Уравнения процесса. Пусть за $n+1$ шаг процесса делается переход из некоторого 0-го исходного состояния в j -е состоя-

ние. Будем считать, что этот переход осуществляется следующим образом: за первые n шагов система переходит из 0-го состояния в одно из возможных состояний i ($i=1, \dots, N$, где N — число состояний процесса), а на последнем, $(n+1)$ -м шаге осуществляется переход из i -го состояния в j -е. Тогда в соответствии с формулой полной вероятности можно записать:

$$p_{0j}(n+1) = \sum_{i=1}^N p_{0i}(n) p_{ij} \quad \text{для всех } 0, j, n. \quad (2.1)$$

Кроме того, можно записать условие

$$\sum_{i=1}^N p_{0i}(n) = 1 \quad \text{для всех } 0, n, \quad (2.2)$$

означающее, что после n шагов система будет находиться в одном из N возможных состояний. Для стационарного эргодического процесса уравнения (2.1), (2.2) примут вид

$$\pi_j = \sum_{i=1}^N \pi_i p_{ij} \quad \text{для всех } j (j=1, \dots, N), \quad (2.3)$$

$$\sum_{i=1}^N \pi_i = 1. \quad (2.4)$$

Уравнения (2.3) можно записать в матричном виде:

$$\Pi = \Pi P, \quad (2.5)$$

где Π — N -мерный вектор; P — матрица переходных вероятностей размера $N \times N$. Для определения предельных вероятностей π_j попадания в j -е состояние ($j=1, \dots, N$) в случае стационарного эргодического процесса достаточно решить систему (2.3), (2.4) относительно N неизвестных π_i . Ранг системы $(N+1)$ уравнений (2.3), (2.4) не выше N , так как $\sum_j p_{ij} = 1$ и уравнения системы (2.3) линейно зависимы.

Управляемые марковские процессы. Марковский процесс называют *управляемым*, если переходные вероятности зависят от некоторого параметра k , который может быть выбран наблюдателем.

Параметр k является в данном случае управляющим воздействием и может быть назван *стратегией управления*. Вы-

бор стратегии в общем случае зависит не только от настоящего состояния, но и от всего предыдущего поведения системы. Стратегия называется *марковской*, если она зависит только от настоящего состояния процесса. Выбор стратегии может меняться также во времени. Стратегия называется *однородной марковской*, если она определяется только настоящим состоянием и не меняется по шагам процесса.

Немарковская стратегия определяет немарковский управляемый процесс. Чтобы процесс был марковским, необходимо использовать марковскую стратегию. То же можно сказать относительно свойства однородности: неоднородная марковская стратегия определяет неоднородный марковский процесс; чтобы процесс был однородным марковским, необходима однородная марковская стратегия.

Для управляемого марковского процесса уравнения (2.5) примут вид

$$\Pi = \Pi P^k, \quad (2.6)$$

где P^k — матрица переходных вероятностей размера $N \times N$, столбцы P_i^k которой определяются в зависимости от выбора k -й стратегии в i -м состоянии. При записи уравнений (2.6) предполагалось, что из множества возможных стратегий в каждом i -м состоянии выбирается только одна фиксированная стратегия. Это есть случай, когда используются так называемые «чистые» стратегии, т. е. стратегии, выбираемые с вероятностью, равной 1. Однако в вероятностных задачах зачастую оказывается более эффективным выбирать не одну фиксированную стратегию с вероятностью 1, а давать решение в виде набора различных рандомизированных стратегий, т. е. стратегий, используемых с некоторой вероятностью, которая может быть меньшей, чем 1. Подобный набор (вектор) стратегий называется *смешанной стратегией*.

Рассмотрим случай использования смешанных стратегий в описываемой задаче. Пусть l_i^k — вероятность использования k -й стратегии в i -м состоянии, а $\pi_i^k = l_i^k \pi_i$ — предельная вероятность того, что после достаточно большого числа переходов система окажется в i -м состоянии и будет принята k -я стратегия. В матрицу P^k в данном случае будет входить для каждого i -го состояния уже не по одному столбцу, соответствующему одной стратегии, выбираемой в i -м состоянии, а группа из $k \in K_i$ столбцов, соответствующих набору стратегий, используемых в i -м состоянии. Уравнения (2.3), (2.4)

управляемого марковского процесса тогда примут вид

$$\sum_{k \in K_j} \pi_j^k - \sum_{k \in K_i} \sum_{i=1}^N \pi_i^k p_{ij}^k = 0, \quad j = 1, 2, \dots, N, \quad (2.7)$$

$$\sum_{k \in K_i} \sum_{i=1}^N \pi_i^k = 1. \quad (2.8)$$

В последующем, при описании свойств рассматриваемых задач, мы убедимся, что всегда существует хотя бы одна чистая стратегия, являющаяся оптимальной. Иначе говоря, рассмотрение смешанных стратегий ничего не дает в отношении улучшения решения, хотя и необходимо для формулировки задачи.

Марковские процессы с доходами. Пусть каждому переходу из i -го состояния в j -е ставится в соответствие некоторый доход r_{ij} . Тогда доход, получаемый при переходе из i -го состояния, средневзвешенный по вероятностям попадания из i -го состояния в любое j -е, запишется как

$$q_i = \sum_{j=1}^{N|} p_{ij} r_{ij}.$$

Доход q_i назван в [87] *непосредственно ожидаемым* доходом при одном переходе из i -го состояния. Далее, пусть выбор стратегий в управляемом марковском процессе влияет на получаемые доходы. Тогда доход, получаемый при переходе из i -го состояния в j -е, при использовании k -й стратегии будет обозначаться r_{ij}^k , а непосредственно ожидаемый доход при одном переходе из i -го состояния запишется как

$$q_i^k = \sum_{j=1}^N p_{ij}^k r_{ij}^k. \quad (2.9)$$

Непосредственно ожидаемый доход является средним доходом, получаемым при одном переходе из i -го состояния в *рассматриваемый* момент времени. Нас будет в дальнейшем интересовать также средний доход, получаемый за один переход процесса, начатого с i -го состояния, *после достаточно большого* числа шагов (теоретически бесконечного).

Пусть после достаточно большого числа шагов система совершила переход из i -го исходного состояния в j -е. Доход,

получаемый за один последующий переход из j -го состояния, равен q_j . С учетом вероятности попадания из i -го исходного состояния в j -е, этот доход составит $\pi_{ij}q_j$. Просуммировав по всем j , получим средневзвешенный по предельным вероятностям доход за один шаг процесса, начатого с i -го состояния, после достаточно большого числа переходов:

$$g_i = \sum_{j=1}^N \pi_{ij} q_j.$$

Доход g_i называют *средним доходом за один шаг*, или *прибылью процесса* бесконечной длительности. В случае эргодического процесса доходы g_i не зависят от того, с какого именно исходного состояния i начался процесс. Тогда

$$g = \sum_{j=1}^N \pi_j q_j. \quad (2.10)$$

Для управляемого марковского процесса доход g запишется в виде

$$g = \sum_{k \in K_j} \sum_{j=1}^N \pi_j^k q_j^k. \quad (2.11)$$

Таким образом, для определения прибыли процесса g за один шаг нужно определить величины π_j , решив систему уравнений (2.3), (2.4), а затем найти g по формуле (2.10). В случае управляемого марковского процесса прибыль процесса находится по формуле (2.11), величины π_j^k в которой выбираются таким образом, чтобы удовлетворились уравнения (2.7), (2.8).

Рассмотрим теперь другой путь расчета марковских процессов с доходами, позволяющий найти g без предварительного определения предельных вероятностей. Пусть $v_i(n+1)$ — полный доход, получаемый за $(n+1)$ переход процесса, начатого с i -го исходного состояния. Можно записать так:

$$v_i(n+1) = \sum_{j=1}^N p_{ij}^k [r_{ij}^k + v_j(n)]. \quad (2.12)$$

Поясним формулу (2.12). Если система, начиная с i -го состояния делает переход в некоторое j -е состояние, а затем делает еще n переходов, то получаемый при этом доход равен

$r_{ij}^k + v_j(n)$. Полный доход за $n+1$ переход берется как сумма доходов $r_{ij}^k + v_j(n)$, взвешенных по вероятностям p_{ij}^k попадания в j -е состояние на первом шаге, откуда и получается формула (2.12). Учтя (2.9), уравнение (2.12) можно записать в следующем виде:

$$v_i(n+1) = q_i^k + \sum_{j=1}^N p_{ij}^k v_j(n). \quad (2.13)$$

При рассмотрении процесса достаточно большой длительности уравнение, определяющее полный доход процесса, приобретает асимптотический характер [87]:

$$v_i(n+1) = (n+1)g_i + v_i. \quad (2.14)$$

Выражение (2.14) интуитивно вполне понятно: g_i показывает, насколько увеличивается полный доход за один шаг процесса, $(n+1)$ — число шагов, v_i — отрезок, отсекаемый асимптотой на координатной оси полных доходов v_i (величину v_i называют [87] *весом i -го исходного состояния*). В случае эргодического процесса уравнение асимптоты (2.14) запишется в форме

$$v_i(n+1) = (n+1)g + v_i. \quad (2.15)$$

Подставив асимптотическое выражение $v_i(n+1)$ и $v_i(n)$ в уравнение (2.13), проведя элементарные алгебраические преобразования и учтя, что $\sum_{j=1}^N p_{ij}^k = 1$, получим

$$g + v_i = q_i^k + \sum_{j=1}^N p_{ij}^k v_j, \quad i = 1, \dots, N. \quad (2.16)$$

Система из N линейных алгебраических уравнений (2.16) позволяет при заданных q_i^k и p_{ij}^k определить величины g и v_i , характеризующие марковские процессы с доходами. При этом, чтобы число неизвестных $g, v_i, i=1, \dots, N$ равнялось числу уравнений, систему координат выбирают такой, где одно из v_i равнялось бы нулю [87].

Итак, в данном параграфе мы кратко рассмотрели некоторые используемые в дальнейшем определения [87, 3, 83, 70], относящиеся к марковским процессам, и могли видеть, что интересующие нас характеристики марковских процессов с доходами могут быть рассчитаны двумя пу-

тиями: 1) нахождением предельных вероятностей π_j из решения системы (2.3), (2.4) с последующим определением прибыли процесса g по формуле типа (2.10); 2) непосредственным нахождением прибыли процесса g и весов v_i из решения системы (2.16).

§ 3. Постановки задач оптимизации марковских процессов с доходами

Задача оптимизации управляемого марковского процесса заключается в определении стратегий, которые следует избрать, находясь на данном шаге в данном состоянии. Оптимальными считаются стратегии, при которых достигается максимум математического ожидания дохода.

Две схемы расчета марковских процессов с доходами, рассмотренные в предыдущем параграфе, определяют две постановки задачи оптимизации.

1. Пусть имеется объект или система, описываемые дискретным однородным марковским процессом с конечным числом состояний ($i=1, 2, \dots, N$). В каждом состоянии может быть выбрана некоторая стратегия поведения с индексом k . Стратегии выбираются из множества, которое задается конечным числом точек либо выпуклым многогранником ($k \in K_i$) [18]. Предполагается, что известны вероятности p_{ij}^k переходов и доходы r_{ij}^k . Требуется определить оптимальные стратегии, которые следует избрать, находясь в i -м исходном состоянии ($i=1, \dots, N$), если предполагается, что система будет работать достаточно длительный период времени (теоретически бесконечный), и процесс при этом — стационарный, эргодический.

При оптимизации процесса бесконечной длительности полный доход за все время процесса неограничен (это видно из выражения (2.14) при $n \rightarrow \infty$). Поэтому в качестве целевой функции принимается максимизация среднего дохода g за один шаг асимптотического процесса. Рассматриваемая задача формулируется в виде задачи линейного программирования [18, 99]. Ограничениями в этой задаче линейного программирования являются уравнения (2.7), (2.8), описывающие управляемый марковский процесс, дополненные естественными условиями неотрицательности предельных вероятностей. Целевой функцией является максимизация линейной формы (2.11).

Таким образом, постановка задачи оптимизации, являющаяся естественным продолжением той схемы расчета марковских процессов с доходами, в которой используется определение предельных вероятностей π_j , имеет вид

$$\left. \begin{aligned} & \sum_{k \in K_j} \sum_{i=1}^N q_i^k \pi_j^k \rightarrow \max \\ \text{при} \quad & \sum_{k \in K_j} \pi_j^k - \sum_{k \in K_i} \sum_{i=1}^N p_{ij}^k \pi_i^k = 0, \quad j = 1, \dots, N, \\ & \sum_{k \in K_i} \sum_{i=1}^N \pi_i^k = 1, \\ & \pi_i^k \geq 0, \quad i = 1, \dots, N, \quad k \in K_i. \end{aligned} \right\} \quad (3.1)$$

2. Рассмотрим теперь постановку задачи оптимизации, основанную на использовании другой схемы расчета марковских процессов с доходами. Полный доход за $n+1$ переход марковского процесса, начатого с i -го состояния, определяется выражением (2.13). Условие максимизации полного дохода запишется в виде

$$v_i(n+1) = \max_k \left[q_i^k + \sum_{j=1}^N p_{ij}^k v_j(n) \right], \quad i = 1, \dots, N. \quad (3.2)$$

Выражение (3.2) — это рекуррентное соотношение динамического программирования, позволяющее определить для каждого состояния i стратегию поведения последовательно по шагам процесса. Сначала для $n=0$ при заданных значениях доходов $v_i(0)$ определяются стратегии для каждого состояния, позволяющие получить оптимальные доходы $v_i(1)$, $i=1, \dots, N$. Затем при известных значениях $v_i(1)$ определяются оптимальные стратегии для получения $v_i(2)$ и т. д. В данном случае n определяет число переходов, оставшихся до конца процесса.

Определение оптимальных стратегий поведения по формуле (3.2) при достаточно больших n затруднительно из-за неограниченного возрастания времени счета. Поэтому в литературе рассматривались вопросы определения оптимальных решений непосредственно для асимптотики процесса. Если рассматривается процесс достаточной длительности (теоретически бесконечной), то оптимальная стратегия при

$n \rightarrow \infty$ будет асимптотически приближаться к некоторой наилучшей стратегии для каждого состояния [3, 14], т. е. оптимальная стратегия не будет зависеть от номера шага процесса, а будет определяться состоянием i .

Рассмотрим постановку задачи определения оптимальных стратегий поведения для системы с бесконечным временем функционирования. Запишем асимптотическое выражение полных доходов процесса (2.16) в виде

$$g = q_i^k + \sum_{j=1}^N p_{ij}^k v_j - v_i, \quad i = 1, 2, \dots, N. \quad (3.3)$$

Полный доход процесса при $n \rightarrow \infty$ неограничен, поэтому, как и ранее, задача будет заключаться в определении стратегии для каждого i -го состояния, максимизирующей средний доход g за один шаг. Учтя выражение (3.3), запишем условия максимизации:

$$g = \max_k \left\{ q_i^k + \sum_{j=1}^N p_{ij}^k v_j - v_i \right\}, \quad i = 1, \dots, N$$

или, иначе,

$$g \geq q_i^k + \sum_{j=1}^N p_{ij}^k v_j - v_i, \quad i = 1, \dots, N, \quad k \in K_i. \quad (3.4)$$

Определить индексы оптимальных стратегий это значит выбрать из условий (3.4) для каждого i -го состояния ($i = 1, \dots, N$) не менее чем по одному условию, обращающемуся в равенство, с тем, чтобы сформировать систему уравнений (3.3) для расчета процесса. Причем осуществить этот выбор нужно таким образом, чтобы прибыль g , получаемая из решения системы (3.3), была максимальна; для этого необходимо выполнение всех остальных условий (3.4). В терминах линейного программирования наша задача будет выглядеть следующим образом: условия (3.4) рассматриваются как ограничения, а целевая функция должна быть такой, чтобы не менее, чем N условий (3.4) обращались в равенства. Рассмотрим, каков должен быть вид целевой функции данной задачи линейного программирования. Пусть мы имеем решение $g', v_1', v_2', \dots, v_N'$, обращающее N условий (3.4) в равенства. Возьмем любое другое решение $g'', v_1'', v_2'', \dots, v_N''$, удовлетворяющее условиям (3.4), где $g'' = g' + \delta$,

$v'_i = v'_i, i = 1, \dots, N$. Подставим это решение в (3.4). Имеем

$$g' + \delta \geq q_i^k + \sum_{j=1}^N p_{ij}^k v'_j - v'_i, \quad i = 1, \dots, N, k \in K_i. \quad (3.5)$$

Вычтем условия (3.3), обращаясь в равенства при решении $q', v'_1, \dots, v'_N$, из соответствующих условий (3.5). Тогда получим

$$\delta \geq 0.$$

Соответствующие N условий обращаются в равенства при $\delta = 0$. Это будет минимальное g , удовлетворяющее условиям (3.5), так как $g'' \geq g'$ при $\delta \geq 0$. Отсюда следует достаточно очевидный факт: для того чтобы иметь условия, обращаясь при решении в равенства, нужно в качестве целевой функции взять условие минимизации g .

Таким образом, задача формулируется в следующем виде:

$$\left. \begin{array}{l} g \rightarrow \min \\ \text{при} \\ g + v_i - \sum_{j=1}^N p_{ij}^k v_j \geq q_i^k, \quad i = 1, \dots, N, k \in K_i. \end{array} \right\} \quad (3.6)$$

Задачу (3.6) можно также получить как двойственную к рассмотренной ранее задаче (3.1). Отсюда ясна связь между двумя описанными постановками исходной задачи оптимизации марковских процессов с доходами. Заметим также, что из рассмотрения задачи (3.6) как двойственной по отношению к задаче (3.1), становится понятным, почему в данном случае условие максимизации прибыли процесса g заложено в ограничениях задачи (3.6), а в качестве целевой функции рассматривается условие $g \rightarrow \min$.

§ 4. Задачи оптимизации марковских процессов с доходами при наличии переоценки

В ряде практических задач необходимо учитывать изменение покупательской способности денег во времени. В этом случае будущие доходы нужно приводить к настоящему времени с помощью *коэффициента переоценки* β , $0 \leq \beta \leq 1$. Коэффициент β можно рассматривать также как вероятность того, что процесс будет продолжен, если исходная

задача такова, что процесс может в некоторый момент времени прекратиться с вероятностью $(1-\beta)$.

Аналогично процессам без переоценки, постановки задач оптимизации процессов с переоценкой будем вести на основе двух схем расчета марковских процессов с доходами, рассмотренных в § 2.

1. Рассмотрим сначала постановку задачи оптимизации, использующую ту схему расчета марковских процессов с доходами, где определяются предельные вероятности π_j^k .

Постановка задачи оптимизации процесса бесконечной длительности затруднена тем, что в данном случае величины π_j^k стремятся к нулю. Действительно, пусть коэффициент β есть вероятность продолжения процесса ($0 \leq \beta \leq 1$). Тогда уравнения процесса (2.1) запишутся в виде

$$p_j(n+1) = \beta \sum_{i=1}^N p_i(n) p_{ij}$$

или, в матричной форме,

$$p(n+1) = \beta p(n) P,$$

где $P=[p_{ij}]$ — матрица размера $N \times N$. Для $n=1, 2, \dots$, имеем [86]

$$p(1) = \beta p(0) P,$$

$$p(2) = \beta p(1) P = p(0) (\beta P)^2,$$

$$p(3) = \beta p(2) P = p(0) (\beta P)^3,$$

$$\dots \dots \dots$$

$$\pi_j = \lim_{n \rightarrow \infty} p_j(n) = p(0) (\beta P)^\infty \rightarrow 0.$$

Интуитивно это вполне понятно, так как предельная вероятность попадания в некоторое состояние после бесконечного числа переходов будет стремиться к нулю, если имеется вероятность прекращения процесса на n -м шаге.

Чтобы обойти трудности в постановке задачи, связанные с тем, что величины π_j^k стремятся к нулю, перейдем на время к рассмотрению иной задачи. Предположим, что мы изменили условия таким образом, что процесс не может прекратиться. Будем, например, считать, что если процесс прекратился на некотором переходе, то он в этот же момент времени вновь возобновляется с какого-либо j -го состояния с вероятностью ε_j [86]. Очевидно, вероятность возобновления

процесса с любого состояния

$$\sum_{j=1}^N \varepsilon_j = 1 - \beta.$$

В стационарном режиме в состояние j можно тогда попасть, если процесс продолжается, либо если он возобновлен на данном переходе с j -го состояния:

$$\pi_j = \beta \sum_{i=1}^N \pi_i p_{ij} + \varepsilon_j, \quad j = 1, \dots, N. \quad (4.1)$$

В данном случае процесс не может прекратиться, т. е.

$$\sum_{i=1}^N \pi_i = 1. \quad (4.2)$$

При учете смешанных стратегий уравнения (4.1), (4.2) запишутся в виде

$$\sum_{k \in K_j} \pi_i^k - \beta \sum_{k \in K_i} \sum_{i=1}^N p_{ij}^k \pi_i^k = \varepsilon_j, \quad j = 1, \dots, N, \quad (4.3)$$

$$\sum_{k \in K_i} \sum_{i=1}^N \pi_i^k = 1. \quad (4.4)$$

Задача в этом случае заключается в максимизации линейной формы (2.11) при ограничениях (4.3), (4.4) и естественном условии неотрицательности предельных вероятностей

$$\pi_i^k \geq 0, \quad i = 1, \dots, N, \quad k \in K_i. \quad (4.5)$$

Однако данная задача может быть поставлена и без условия (4.2). Для того чтобы процесс не прекратился, достаточно выбрать ε_j в условиях (4.1) таким образом, чтобы $\sum_{j=1}^N \varepsilon_j = 1 - \beta$. Это можно легко проверить. Просуммировав условия (4.1) по всем j и учтя, что $\sum_{j=1}^N p_{ij} = 1$, получаем

$$\sum_{j=1}^N \pi_j - \beta \sum_{j=1}^N \sum_{i=1}^N p_{ij} \pi_i = \sum_{j=1}^N \varepsilon_j$$

или

$$\sum_{j=1}^N \pi_j - \beta \sum_{i=1}^N \pi_i \sum_{j=1}^N p_{ij} = 1 - \beta,$$

откуда

$$\sum_{i=1}^N \pi_i = 1.$$

Иными словами, если выбрать ε_i таким образом, что $\sum_{j=1}^N \varepsilon_j = (1 - \beta)$, то условие $\sum_{i=1}^N \pi_i = 1$ является линейной комбинацией условий (4.1) и его можно не записывать.

Итак, при условии, что процесс не может прекратиться, задача заключается в максимизации целевой функции (2.11)

$$\sum_{k \in K_j} \sum_{j=1}^N q_j^k \pi_j^k \rightarrow \max \quad (4.6)$$

при ограничениях (4.3), (4.5), причем ε_j в условиях (4.3) нужно выбирать таким образом, чтобы

$$\sum_{j=1}^N \varepsilon_j = 1 - \beta.$$

В дальнейшем при рассмотрении алгоритмов решения задач марковского программирования будет показано следующее важное свойство задач типа (4.3), (4.5), (4.6): качественное решение задачи не зависит от значений компонент ε_j правой части ограничений. Под качественным решением в данном случае понимается определение индексов столбцов, входящих в оптимальный базис. Отметим, что во всех задачах оптимизации управляемых марковских процессов нас интересует именно качественное решение, т. е. номера стратегий (соответственно столбцов), являющихся оптимальными.

Вернемся теперь к исходной задаче. В данном случае процесс может прекратиться, т. е. не требуется выполнения равенства

$$\sum_{j=1}^N \varepsilon_j = 1 - \beta.$$

Однако задача (4.3), (4.5), (4.6), сформулированная таким образом, что процесс не может прекратиться,

позволяет благодаря своим специфическим свойствам решать задачи оптимизации марковских процессов с переоценкой, так как качественное решение этих задач совпадает (не зависит от правой части).

Таким образом, определение оптимальных стратегий управления системой, описываемой марковским процессом с доходами, при наличии переоценки сводится к качественному решению задачи линейного программирования (4.3), (4.5), (4.6) при произвольных значениях компонент $\varepsilon_j \geq 0$. Подчеркнем еще раз, что решение задачи (2.11), (4.3), (4.5) дает возможность определить только номера оптимальных стратегий для каждого состояния. Значения переменных в данном случае физического смысла не имеют.

2. Рассмотрим теперь постановку задачи оптимизации процесса с переоценкой, основываясь на втором пути расчета марковского процесса с доходами.

Уравнение (2.12), определяющее полный доход за $n+1$ переход процесса, начатого с i -го состояния, примет следующий вид при наличии переоценки:

$$v_i(n+1) = \sum_{j=1}^N p_{ij}^k [r_{ij}^k + \beta v_j(n)]. \quad (4.7)$$

Рекуррентное соотношение динамического программирования для оптимизации процессов с переоценкой, по аналогии с (3.2), запишется в виде

$$v_i(n+1) = \max_k \left[q_i^k + \beta \sum_{j=1}^N p_{ij}^k v_j(n) \right].$$

Будем, как и ранее, рассматривать асимптотику процесса. Полный доход процесса с переоценкой при бесконечном числе переходов равен конечному числу [87, 86]

$$\lim_{n \rightarrow \infty} v_i(n) = v_i,$$

т. е. уравнение асимптоты имеет вид

$$v_i(n) = v_i.$$

В данном случае v_i имеет смысл предельных доходов. Подставив уравнение асимптоты в (4.7), получим

$$v_i = q_i^k + \beta \sum_{j=1}^N p_{ij}^k v_j, \quad i = 1, \dots, N. \quad (4.8)$$

Система (4.8) позволяет при выбранных стратегиях рассчитать марковскую цепь с доходами при наличии переоценки, т. е. позволяет определить предельные доходы v_i при заданных q_i^k , β и p_{ij}^k . Задача оптимизации процесса с переоценкой для случая бесконечной длительности заключается в следующем: требуется для каждого i -го состояния выбрать стратегию управления k таким образом, чтобы получить максимум предельных доходов v_i :

$$v_i = \max_k \left[q_i^k + \beta \sum_{j=1}^N p_{ij}^k v_j \right], \quad i = 1, \dots, N. \quad (4.9)$$

Иначе выражение (4.9) можно записать в форме

$$v_i \geq q_i^k + \beta \sum_{j=1}^N p_{ij}^k v_j, \quad i = 1, \dots, N, k \in K_i. \quad (4.10)$$

Решить данную задачу оптимизации — это значит определить индексы строк системы (4.10), обращающиеся в равенства, чтобы сформировать систему (4.8). При этом условия (4.10) следует рассматривать как ограничения некоторой задачи линейного программирования, так как только при выполнении условий (4.10) достигается максимум дохода v_i (4.9). А целевую функцию задачи линейного программирования следует выбрать таким образом, чтобы в оптимальной точке не менее, чем N условий (4.10) обращались в равенства. Рассматривая данную задачу как двойственную к задаче (4.3), (4.5), (4.6), получим

$$\left. \begin{array}{l} \sum_{j=1}^N \varepsilon_j v_j \rightarrow \min \\ \text{при} \\ v_i - \beta \sum_{j=1}^N p_{ij}^k v_j \geq q_i^n, \quad i = 1, \dots, N, k \in K_i, \end{array} \right\} \quad (4.11)$$

где ε_j — произвольные неотрицательные числа. Задача (4.11) может быть получена и независимо от задачи (4.3), (4.5), (4.6), для чего достаточно показать [86], что при целевой функции $\sum_{j=1}^N \varepsilon_j v_j$, для каждого i не менее, чем по одному условию (4.10) обращается в равенства.

§ 5. Примеры задач оптимизации марковских процессов с доходами

Для постановки задачи динамической оптимизации некоторой системы в терминах марковских процессов с доходами необходимо следующее:

а) убедиться, что поведение системы может быть описано марковским процессом;

б) установить, какие физические величины принимаются за состояния марковского процесса, какие — за стратегии управления (при этом следует учитывать, что возможность практического решения задачи оптимизации определяется в основном числом состояний);

в) определить вероятности p_{ij}^k перехода из i -го состояния в j -е при выборе k -й стратегии, и доходы r_{ij}^k , получаемые при этом.

Если эти три пункта выполнены, то можно приступить к постановке задачи. При этом следует установить, какой марковский процесс рассматривается (однородность, эргодичность, наличие переоценки, дискретность во времени, задание множества возможных стратегий и т. д.).

Рассмотрим некоторые примеры, следуя приведенному плану формализации задачи.

1. П л а н и р о в а н и е д е я т е л ь н о с т и п р е д п р и я т и я. Предположим, что некоторое предприятие выпускает продукцию, которая может либо пользоваться спросом либо не пользоваться спросом [87]. В начале каждого года рассматривается вопрос о том, как действовать далее. Если продукция пользуется спросом, то можно: а) продолжать выпуск той же продукции; б) перейти к выпуску новой продукции. Если продукция не пользуется спросом, то можно: а) усовершенствовать выпускаемую продукцию; б) перейти к выпуску новой продукции.

Прежде чем перейти к постановке задачи, убедимся в том, что выполнены предпосылки, необходимые для постановки проблемы в терминах марковского программирования.

а) Рассматриваемый процесс является марковским, так как решение, принимаемое в данный момент времени (и, соответственно, будущее состояние) определяется только настоящим состоянием системы, а не прошлым.

б) Поведение системы может быть описано двумя состояниями i : состояние 1 — продукция пользуется спросом;

состояние 2 — продукция не пользуется спросом. Решение (а) или (б) для каждого состояния является в данном случае стратегией управления k .

в) Заданы вероятности переходов p_{ij}^k и доходы r_{ij}^k .

Вероятности p_{ij}^k считаются неизменными по шагам, т. е. процесс является однородным. При достаточно большом числе шагов из любого состояния можно попасть в i -е состояние, причем, состояния не являются ни периодическими, ни нулевыми [83]. Это говорит о том, что процесс эргодический. Заметим также, что процесс является дискретным во времени (с интервалом в один год), а стратегии управления задаются конечным множеством (в данном случае всего две возможные стратегии: (а) и (б)). Задача заключается в том, чтобы определить оптимальную стратегию управления в данный момент времени, если система находится в одном из возможных состояний и предполагается, что предприятие будет работать достаточно длительное время (теоретически — бесконечное). Задача может ставиться без учета переоценки будущих доходов относительно настоящего времени — в этом случае формулировка задачи будет записываться в виде (3.2) или в виде задач линейного программирования (3.1), (3.6). При учете коэффициента переоценки задача будет записываться в виде (4.9) или в виде задачи линейного программирования (4.11) (или (4.3), (4.5), (4.6)).

2. Задачи управления запасами. Научное управление запасами является в настоящее время самостоятельной теорией, которой посвящены многочисленные статьи и монографии (см., например, [6, 18, 73, 86, 99]). Задачи планирования запасов возникают в самых различных сферах деятельности человека: в экономике, в военном деле, в промышленности, в торговле, в медицинском обслуживании населения и т. д. При этом трудности планирования запасов обусловлены неопределенностью природы снабжения и спроса.

Рассмотрим следующую задачу [18, 99]. Имеется некоторый запас продукта. Спрос на продукт задан распределением вероятностей. В начале каждого планируемого периода времени (например, месяца) решается вопрос о производстве дополнительных запасов. Как при недостатке продукта, так и при его избытке предприятие несет потери. Требуется определить оптимальный объем запасов, который необходимо произвести в данном месяце, если предполагается, что

предприятие будет функционировать достаточно длительное время (теоретически — бесконечное) и известен запас, имеющийся на начало месяца. Оптимальность понимается в смысле минимизации математического ожидания общих потерь за весь период функционирования.

Предпосылки, необходимые для постановки задачи марковского программирования, выполнены.

а) Процесс в данном случае марковский.

б) В качестве состояния процесса i имеет смысл рассматривать объем запаса, имеющийся в начале месяца. В общем случае при этом будет получен марковский процесс с бесконечным множеством состояний. При проведении дискретизации следует ограничивать число состояний вычислительными возможностями ЦВМ. В качестве стратегии управления k рассматривается объем дополнительных запасов, производимых в данном месяце.

в) Вероятности переходов p_{ij}^k определяются дискретным распределением спроса из зависимости

$$i(t+1) = i(t) + k(t) - h(t), \quad (5.1)$$

где $i(t)$, $i(t+1)$ — объем запаса, имеющегося в начале t -го и $(t+1)$ -го месяцев; $k(t)$ — объем дополнительных запасов, производимых в t -м месяце (в единицах продукта); $h(t)$ — спрос на продукт в t -м месяце (в единицах продукта). Доходы r_{ij}^k определяются конкретно для каждой задачи, исходя из доходов от продажи продукта, расходов на хранение в течение месяца запаса, не реализованного в прошлом месяце, затрат на производство дополнительных запасов в данном месяце, штрафа за неудовлетворенный спрос и так далее.

Обычно считается, что функция распределения спроса одинакова для каждого месяца. Процесс в этом случае однородный. Все состояния эргодические. Множество возможных стратегий для каждого состояния может быть задано конечным множеством точек либо выпуклым многогранником.

Условие максимизации общих доходов (или минимизации потерь) соответствует в данном случае условию (4.6) и задача может быть сформулирована в виде задачи линейного программирования (3.1). Рассматриваемая задача может быть записана также в любом другом виде, приведенном в предыдущих параграфах.

3. Оптимизация ирригационных систем с регулируемым стоком. Рассмотрим систему, состоящую из орошаемого массива и водохранилища, регулирующего сток реки [12, 101]. Полезный объем воды в водохранилище $i(t)$ в начале некоторого периода времени t зависит от притока $h(t)$, который носит случайный характер. Поэтому объем воды в водохранилище является случайной величиной. Вода водохранилища используется для орошения. Чем больше воды отпускается на орошение, тем больший доход может быть получен от земледелия, так как большая часть площади может быть орошена. Однако если в рассматриваемый период времени много воды водохранилища будет отдано на орошение, то в последующие периоды времени воды может не хватить. Требуется определить оптимальное количество воды водохранилища $k(t)$, отпускаемое на орошение в рассматриваемый период времени, при учете того, что система будет функционировать достаточно длительное время.

Предпосылки, необходимые для постановки задачи в терминах марковского программирования, выполнены, так как:

а) процесс работы системы «водохранилище — орошаемый массив» можно рассматривать как управляемый марковский процесс;

б) полезный объем воды в водохранилище имеет смысл принять за состояние процесса, а количество воды, идущее на орошение — за стратегию управления. Так как процесс при этом имеет бесконечное множество состояний, то следует провести дискретизацию числа состояний, учтя физические соображения и вычислительные возможности ЦВМ;

в) вероятности переходов p_{ij}^k определяются из соотношения, связывающего величины $i(t)$, $h(t)$ и $k(t)$:

$$i(t+1) = \min \{ [i(t) + h(t) - k(t)], i^{\text{np}} \}, \quad (5.2)$$

где $i(t)$, $i(t+1)$ — полезный объем воды в водохранилище в начале t -го и $(t+1)$ -го периодов времени; i^{np} — предельно возможный полезный объем воды в водохранилище. Случайная величина притока $h(t)$ задана рядом распределения. Отсюда легко найти и вероятности переходов p_{ij}^k .

Из приведенного соотношения видна связь данной задачи с рассмотренной ранее задачей управления запасами. Задача управления ирригационной системой в данном виде

может рассматриваться как одна из задач создания запасов, если учесть отличие в знаках, стоящих перед значениями $h(t)$ и $k(t)$ в выражениях (5.1) и (5.2).

Доходы r_{ij}^k определяются возможностью орошения тех или иных культур при данном количестве воды k , идущем на орошение. Конкретно, доходы r_{ij}^k находятся следующим образом: при заданном количестве воды, идущем на орошение, решается задача распределения выделенной воды на орошение различных культур таким образом, чтобы достигнуть максимума суммарного дохода, получаемого от земледелия. Это задача распределения ресурсов при линейной функции цели [12], которая решается элементарно. Оптимум целевой функции данной задачи и берется в качестве дохода r_{ij}^k .

Время t в данном случае целесообразно считать дискретным с шагом в один год. В этом случае марковский процесс будет однородным, так как вероятности переходов будут определяться вероятностями годового притока воды, которые считаются неизменными по шагам. Все состояния процесса эргодические. При постановке задачи имеет смысл учитывать переоценку будущих доходов относительно настоящего времени.

Изложенное выше позволяет записать задачу оптимизации ирригационной системы в виде (4.9) или в виде задач линейного программирования (4.11) и (4.3), (4.5), (4.6).

4. Задача о замене оборудования. В процессе эксплуатации оборудование изнашивается, удорожается его эксплуатация и понижается эффективность. Ставится вопрос — когда целесообразно заменить оборудование? Подобные задачи возникают не только непосредственно при замене оборудования, но и в случае замены сырья, решения вопроса о времени ремонта оборудования и т. д. Например, в нефтехимических процессах [73] к этому классу задач относятся задачи о смене катализатора, очистке теплообменников и удалении отложений со стенок реактора.

Задача о замене оборудования решается методами динамического программирования. В [87] показано, что данная задача может быть сформулирована в терминах марковского программирования.

В качестве состояния процесса рассматривается «возраст» оборудования. Пусть, например $i=0, 1, 2, \dots, N$ лет. Замена оборудования i -го возраста оборудованием k -го возраста принимается за стратегию управления ($k=0, 1, 2, \dots$

..., N). В случае, если $k=i$, сохраняется прежнее оборудование.

Вероятности переходов p_{ij}^k определяются следующим образом. Имеется вероятность p_i сохранения работоспособности («выживания») оборудования от i -го до $(i+1)$ -го возраста. Тогда

$$p_{ij}^k = \begin{cases} p_k & \text{при } j = i + 1, \\ 0 & \text{при остальных } j. \end{cases}$$

Например, если сохраняется прежнее оборудование ($k=i$), то

$$p_{ij}^k = \begin{cases} p_i & \text{при } j = i + 1, \\ 0 & \text{при остальных } j. \end{cases}$$

Доходы r_{ij}^k в зависимости от стратегии определяются либо стоимостью r_i эксплуатации имеющегося оборудования возраста « i » (при $k=i$) в течение года, либо ценой r_k (при $k \neq i$) замены оборудования и его эксплуатации:

$$r_{ij}^k = \begin{cases} r_k & \text{при } j = i + 1, \\ 0 & \text{при остальных } j. \end{cases}$$

В случае сохранения прежнего оборудования ($k=i$):

$$r_{ij}^k = \begin{cases} r_i & \text{при } j = i + 1, \\ 0 & \text{при остальных } j. \end{cases}$$

Естественно, доход в данном случае может иметь отрицательный знак. Нетрудно видеть, что теперь имеется все необходимое для записи задачи в терминах марковских процессов с доходами.

Приведенные нами примеры, разумеется, не претендуют на охват всего класса задач, которые могут быть сформулированы в терминах марковского программирования*), и по замыслу авторов должны лишь в определенной мере отметить само разнообразие возможных задач, решаемых средствами марковского программирования.

*) Например, в [87] приводится задача массового обслуживания, рассмотренная в данных терминах.

§ 6. Заключительные замечания

Задачам оптимизации марковских процессов с доходами посвящено много работ. В [3] данные задачи рассматривались с точки зрения динамического программирования. В [87] был предложен метод, специально предназначенный для решения рассматриваемых задач оптимизации при бесконечном времени планирования. Постановка данных задач в терминах линейного программирования описывалась, например, в работах [99, 97, 18, 96, 86].

В данной главе мы попытались с единой точки зрения описать различные постановки задач марковского программирования, рассматривая их как естественное продолжение двух известных методов расчета марковских процессов с доходами. Расчету процесса, основанному на определении предельных вероятностей π_i , соответствует задача (3.1), если не учитывается переоценка доходов относительно настоящего времени, и соответствует задача (4.3), (4.5), (4.6), если учитывается переоценка доходов. Расчету марковского процесса с доходами, основанному на определении доходов v_i , соответствует задача (3.6), если не учитывается переоценка будущих доходов, и соответствует задача (4.11), если учитывается переоценка. Полученные задачи линейного программирования (3.1) и (3.6) (для процессов без переоценки), а также задачи (4.3), (4.5), (4.6) и (4.11) (для процессов с переоценкой) являются *двойственными* друг относительно друга. Отсюда становится понятной связь описанных постановок задач.

Рассмотрение задач оптимизации марковских процессов с точки зрения линейного программирования оказывается полезным не только в отношении единого подхода к различным постановкам задачи, но и в алгоритмическом отношении. Специфика получаемых задач линейного программирования (матрицы ограничений являются S -матрицами) позволяет упростить алгоритмы решения. Алгоритмы решения задач марковского программирования будут приведены в главе IV.

В данной главе мы подробно останавливались на постановках задач оптимизации только для однородных эргодических марковских процессов. Однако класс рассматриваемых задач, относящихся к марковскому программированию, может быть значительно расширен. Задачи оптимизации неэргодических процессов также можно рассматривать с по-

зиций линейного программирования (см., например, в [86]), что может оказаться полезным как в постановочном, так и в алгоритмическом отношении. Задачи оптимизации неоднородных марковских процессов рассматриваются при конечном времени функционирования системы. В этом случае задачи также могут ставиться в терминах линейного программирования [86]. Рассмотрение можно расширить также на полумарковские процессы принятия решений [36, 75], т. е. на такие процессы, в которых время между переходами является случайной величиной. Мы не останавливались также на доказательствах существования решений задач оптимизации марковских процессов с доходами. Данные вопросы рассматриваются в [3, 5, 14, 64, 69, 74] и других работах. Следуя характеру предлагаемой работы, мы рассмотрели в данной главе вопросы постановки задач марковского программирования, а далее, в главе IV остановимся на вычислительных аспектах решения данных специальных задач линейного программирования.

АЛГОРИТМЫ РЕШЕНИЯ ЛИНЕЙНЫХ ЗАДАЧ ОПТИМИЗАЦИИ
МНОГОСВЯЗНЫХ СИСТЕМ

В первых главах книги рассматривалось большое число объектов, которые могут быть отнесены к многосвязным объектам, работающим в статическом режиме. Специфика всех этих объектов заключается в том, что они описываются системой линейных алгебраических уравнений с матрицей, каждый столбец и каждая строка которой содержат не более одного положительного элемента, а остальные элементы матрицы — неположительные. Подобные матрицы были названы нами S -матрицами. Из рассмотренных примеров видно, что S -матрицы используются при описании достаточно широкого класса объектов. Это объекты, описываемые уравнениями Лапласа, типа Лапласа, Пуассона, типа Пуассона и некоторыми другими дифференциальными уравнениями в частных производных (после конечно-разностной аппроксимации), электрические цепи, балансовые модели экономики, объекты, поведение которых описывается управляемым марковским процессом, и многие другие. Были приведены формулировки задач оптимизации данных объектов в терминах линейного программирования.

Однако практическое решение полученных задач линейного программирования наталкивается на значительные трудности, связанные с большой размерностью. Например, в объектах, описываемых уравнениями типа Лапласа или Пуассона, большая размерность задачи вызвана тем, что после дискретизации физическое поле заменяется сеткой, состоящей из большого числа узлов. В реальных задачах оптимизации поля давлений нефтяного пласта число переменных может достигать нескольких тысяч даже в том случае, если учитывать не все узлы сетки, а только те узлы, в которых расположены скважины. Хорошо известно, как остро стоит проблема размерности в задачах оптимизации межотраслевого баланса и многих других. Известно также, что матрицы ограничений получаемых задач линейного програм-

мирования имеют в общем случае произвольное, неблочное заполнение ненулевыми значениями.

Таким образом, из предыдущего изложения мы убедились в том, что рассматриваемый класс задач оптимизации, с одной стороны, достаточно широк, а с другой стороны, — решение данных задач реальной размерности наталкивается на значительные трудности. Все это показывает целесообразность выделения задач оптимизации рассматриваемых объектов в отдельный специальный класс задач линейного программирования, для которых могут быть предложены эффективные алгоритмы решения. Разработке некоторых алгоритмов решения данных задач и посвящена эта глава.

Рассмотрение ведется на основании следующей задачи линейного программирования, названной нами S -задачей:

$$L = \sum_{j=1}^n c_j x_j \xrightarrow{\text{max}} \quad (0.1)$$

при

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, \dots, m, \quad (0.2)$$

$$x_j \geq 0, \quad j = 1, \dots, n, \quad (0.3)$$

где $A = [a_{ij}]$ — S -матрица, $B = [b_i] \geq 0$. К виду (0.1) — (0.3) сводятся задачи оптимизации, рассмотренные в первой главе. В данной главе приводятся также некоторые обобщения задачи (0.1) — (0.3). Показывается, например, что специфика задачи, позволяющая строить эффективные алгоритмы ее решения, не нарушится при любом знаке условий (0.2) ($\leq$, $=$, $\geq$), и при постановке задачи на минимум целевой функции.

Изложение строится следующим образом. В § 1 рассматриваются свойства S -задач, обусловленные их спецификой. На основе данных свойств в § 2 предлагаются некоторые специальные алгоритмы решения S -задач, приводятся примеры. При этом рассматриваются как конечные, так и итерационные процедуры решения. В § 3 обсуждаются различные применения свойств S -задач и некоторые замечания вспомогательного характера. В § 4 показывается возможность получения приближенного решения S -задач без проведения каких-либо вычислений либо с помощью небольшого объема вычислений.

Для иллюстрации применения предлагаемых алгоритмов в § 5, 6 и 7 приводятся примеры подробного решения задачи оптимизации нефтедобычи. В трех последующих параграфах рассматриваются некоторые обобщения и замечания. В § 11 проводится обсуждение — рассматриваются преимущества S -задач перед общими задачами линейного программирования. На этом заканчивается рассмотрение алгоритмов решения линейных статических задач оптимизации многосвязных систем. В § 12 рассматриваются некоторые аспекты решения линейных динамических задач оптимизации многосвязных систем.

Заметим, что разделение на статические и динамические задачи в данном случае проводится с той точки зрения, что объект описывается как объект многосвязного регулирования с соответствующей передаточной функцией (см. § 2 первой главы). Но это не означает, что все объекты из первой названной нами группы являются статическими по своей природе. Например, ранее мы рассматривали некоторые объекты, описываемые дифференциальными уравнениями в частных производных, которые вполне могут быть по своей природе динамическими.

§ 1. Свойства S -задачи

Запишем задачу (0.1) — (0.3) в каноническом виде:

$$L = \sum_{j=1}^n c_j x_j \rightarrow \max \quad (1.1)$$

при

$$\sum_{j=1}^n a_{ij} x_j + x_i^0 = b_i, \quad i = 1, \dots, m, \quad (1.2)$$

$$x_j \geq 0, \quad x_i^0 \geq 0, \quad (1.3)$$

где $b_i \geq 0$, $a_{ij} \leq 0$ для $i \neq j$. Очевидно, что данная задача включает как частный случай любую запись S -задачи (в том смысле, что задача может ставиться как на максимум, так и на минимум целевой функции и знаки ограничивающих неравенств также могут быть любыми $\leq$, $=$, $\geq$). Если задача ставится на минимум, то она сводится к задаче максимизации изменением знаков целевой функции на противополо-

ложные. При ограничениях вида $\leq$ переменные x_i^0 являются дополнительными переменными, обращающими данные условия в равенства. Для тех случаев, когда имеются ограничения $=$, $\geq$, переменные x_i^0 являются искусственными. Особенности решения S-задач в этом случае будут отмечаться ниже, в § 9 данной главы.

Эффективные алгоритмы решения могут быть построены на основе использования специальных свойств S-задач. Опишем эти свойства.

Свойство 1. В любом опорном плане задачи (1.1) — (1.3) для каждого i положительна не более, чем одна из переменных x_i , x_i^0 .

Доказательство. Возьмем произвольный опорный план, при котором имеется r положительных переменных x_i , x_i^0 . Рассмотрим соответствующую систему из m_1 линейно независимых условий (1.2), включающую только столбцы при положительных переменных. Для выполнения каждого i -го условия этой системы хотя бы одна из двух переменных x_i , x_i^0 должна входить в рассматриваемую систему (т. е. должна быть положительна). В противном случае в i -м уравнении слева будет отрицательное число $\sum_{j \neq i} a_{ij}x_j < 0$ (если имеются $a_{ij} \neq 0$, $i \neq j$), а справа — неотрицательное число $b_i \geq 0$, и условия этого уравнения выполняться не могут. Если же в i -м уравнении все $a_{ij} = 0$, $i \neq j$, то без элементов $a_{ii}x_i$, x_i^0 данная строка является нулевой (линейно зависимой), и не включается в рассматриваемую систему по построению.

Как известно из теории линейного программирования, система уравнений, соответствующая опорному плану с r положительными переменными, содержит r линейно независимых условий: $m_1 = r \leq m$. Таким образом, мы имеем r линейно независимых условий, для выполнения каждого i -го из них хотя бы одна из двух переменных x_i , x_i^0 должна быть положительна, а общее количество положительных переменных равно r . Отсюда для каждого i , соответствующего номеру условия, входящего в рассматриваемую систему, только одна из двух переменных x_i , x_i^0 положительна. Если же учитывать не только r линейно независимых, но все m условий системы (1.2) (где $r \leq m$), то для каждого i положительна будет не более чем одна из двух переменных x_i , x_i^0 , что и требовалось доказать.

Свойство опорных планов S -задач имеет очевидную геометрическую интерпретацию при $n=2$. На рис. 3.1 проиллюстрирован простейший пример:

$$L = -x_1 + x_2 \rightarrow \max$$

при

$$\begin{aligned} (1) \quad & 2x_1 - x_2 \leq 2, \\ (2) \quad & -x_1 + 3x_2 \leq 3, \\ (3) \quad & -x_1 - x_2 \leq 1, \\ & x_1 \geq 0, \quad x_2 \geq 0. \end{aligned}$$

В каноническом виде условия (1) — (3) записываются так:

$$\begin{aligned} 2x_1 - x_2 + x_1^0 &= 2, \\ -x_1 + 3x_2 + x_2^0 &= 3, \\ -x_1 - x_2 + x_3^0 &= 1. \end{aligned}$$

Условие (3) является излишним. Область допустимых точек — многогранник $OABC$. Выполнение свойства 1 очевидно:

в точке A	в точке B
$\begin{cases} x_1 = 0, \quad x_1^0 > 0, \\ x_2 > 0, \quad x_2^0 = 0; \end{cases}$	$\begin{cases} x_1 > 0, \quad x_1^0 = 0, \\ x_2 > 0, \quad x_2^0 = 0; \end{cases}$
в точке C	в точке O
$\begin{cases} x_1 > 0, \quad x_1^0 = 0, \\ x_2 = 0, \quad x_2^0 > 0; \end{cases}$	$\begin{cases} x_1 = 0, \quad x_1^0 > 0, \\ x_2 = 0, \quad x_2^0 > 0. \end{cases}$

Некоторые алгоритмы решения S -задач можно построить путем существенного упрощения процедуры симплекс-метода применительно к данной задаче. Эти упрощения основываются на ряде утверждений, которые мы докажем.

Свойство 2. При решении S -задачи симплекс-методом

- а) каждый столбец матрицы ограничений только один раз может быть разрешающим;*
- б) разрешающие элементы всегда находятся на псевдодиагонали *) матрицы A .*

*) Предполагается, что строки и столбцы матрицы A пронумерованы таким образом, что положительные элементы находятся на диагонали, выходящей из левого верхнего угла матрицы. Эту диагональ мы называем псевдодиагональю. Для квадратных матриц псевдодиагональ совпадает с главной диагональю.

Доказательство. Исходная точка при решении задачи (1.1) — (1.3) симплекс-методом очевидна: $x_i^0 = b_i$. Обозначим через $A^{(p)}$ — матрицу коэффициентов выражения внебазисных векторов через векторы базиса, получаемую на p -й итерации решения задачи симплекс-методом. Для удобства примем форму записи симплекс-таблицы из [39]. Тогда записываются только внебазисные векторы, и матрица $A^{(p)}$ получается после каждой итерации симплекс-метода на

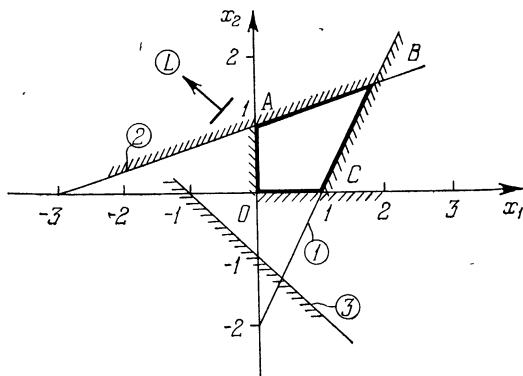


Рис. 3.1. Геометрическая иллюстрация S-задачи.

месте матрицы A . При этом вектор, выведенный из базиса на данной итерации, записывается на месте вектора, введенного в базис. Преобразования совершаются по рекуррентным формулам модифицированных жордановых исключений (м. ж. и.):

$$\left. \begin{aligned} a_{ij}^{(p+1)} &= a_{ij}^{(p)} - \frac{a_{ir}^{(p)} \cdot a_{rj}^{(p)}}{a_{rr}^{(p)}} && \text{для } i \neq r, j \neq r, \\ a_{ir}^{(p+1)} &= -\frac{a_{ir}^{(p)}}{a_{rr}^{(p)}} && \text{для } i \neq r, \\ a_{rj}^{(p+1)} &= \frac{a_{rj}^{(p)}}{a_{rr}^{(p)}} && \text{для } j \neq r, \\ a_{rr}^{(p+1)} &= \frac{1}{a_{rr}^{(p)}}, \end{aligned} \right\} \quad (1.4)$$

где индексом r обозначается разрешающая строка или столбец (рис. 3.2) на p -й итерации. Оценки векторов условий относительно данного базиса обозначим через $\Delta f^{(p)}$. Так как

рассматривается задача максимизации, то в базис будут вводиться векторы, имеющие отрицательные оценки $\Delta_j^{(p)}$. При этом j -й столбец матрицы $A^{(p)}$ будет разрешающим. Преобразования оценок по итерациям совершаются аналогично (1.4):

$$\left. \begin{aligned} \Delta_j^{(p+1)} &= \Delta_j^{(p)} - \frac{\Delta_r^{(p)} a_{rj}^{(p)}}{a_{rr}^{(p)}}, & j \neq r, \\ \Delta_r^{(p+1)} &= -\frac{\Delta_r^{(p)}}{a_{rr}^{(p)}}. \end{aligned} \right\} \quad (1.5)$$

Необходимые обозначения и термины пояснены. Приступим к доказательству. На первой итерации, очевидно, направляющий элемент может находиться только на псевдодиагонали,

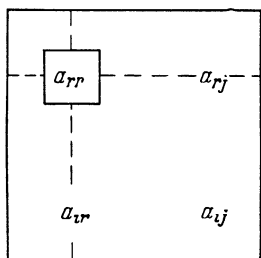


Рис. 3.2. Индексация элементов при преобразованиях матрицы по методу Гаусса.

так как все остальные элементы a_{ij} в S -матрице неположительные. После преобразования (см. (1.4)) элементы разрешающего столбца ($j=r$) станут неотрицательными, а элементы всех остальных столбцов получат неположительные приращения:

$$\delta a_{ij}^{(1)} = \frac{a_{ir}^{(1)} \cdot a_{rj}^{(1)}}{a_{rr}^{(1)}} \quad \text{для } i \neq r.$$

Знаки элементов разрешающей строки ($i=r$) остаются после преобразования теми же.

На второй итерации разрешающий элемент вновь может быть только на псевдодиагонали. Приращения $\delta a_{ij}^{(2)} \geq 0$, $\delta \Delta_j^{(2)} \geq 0$, $i \neq r$, $j \neq r$ для столбца, который уже использовался ранее (на первой итерации) в качестве разрешающего; $\delta a_{ij}^{(2)} \leq 0$, $i \neq r$, $j \neq r$ для элементов остальных столбцов матрицы. Элементы разрешающего столбца данной итерации становятся после преобразования неотрицательными, а элементы разрешающей строки сохраняют прежние знаки.

Если рассматривать процесс далее, то по индукции мы убедимся в том, что $a_{ij}^{(p)} \geq 0$ и $\Delta_j^{(p)} > 0$ для столбцов, которые были разрешающими (для них $\delta a_{ij}^{(p)} \geq 0$, $i \neq r$, $\delta \Delta_j^{(p)} \geq 0$) и $a_{ij}^{(p)} \leq 0$, $i \neq j$ для столбцов, которые не использовались в качестве разрешающих (для них приращения $\delta a_{ij}^{(p)} \leq 0$, $i \neq r$, $j \neq r$).

Отсюда можно заключить следующее:

1) j -й столбец матрицы $A^{(p)}$, который был однажды разрешающим, больше не будет в роли разрешающего, так как оценка $\Delta_j^{(p)}$ этого столбца остается на последующих итерациях положительной, что доказывает пункт а) свойства 2;

2) для j -го столбца, который не был разрешающим, положительные элементы $a_{ij}^{(p)}$ могут находиться только на псевдо-диагонали, что доказывает пункт б) свойства 2.

Свойство 2б можно рассматривать как интерпретацию свойства 1 в терминах симплекс-метода.

На основании приведенного утверждения можно доказать также следующие свойства S-задачи.

Свойство 3. Если вектор A_j матрицы ограничений S-задачи имеет отрицательную оценку Δ_j хотя бы на одной итерации решения задачи симплекс-методом, то он обязательно войдет в оптимальный базис.

Иначе свойство 3 можно записать следующим образом: *Вектор, однажды введенный в базис при решении S-задачи симплекс-методом, больше не будет выведен из базиса.*

Доказательство. Введем обозначения: A_j ($j=1, \dots, n$) — столбцы коэффициентов при переменных x_j ; A_j^0 ($j=n+1, \dots, n+m$) — столбцы коэффициентов при дополнительных переменных x_j^0 ($j=n+1, \dots, n+m$).

В исходной точке задачи (1.1) — (1.3) все векторы A_j — внебазисные, все векторы A_j^0 — базисные. Пусть на некоторой итерации решения S-задачи симплекс-методом вектор A_k имеет отрицательную оценку Δ_k . Введем A_k в базис. На основании свойства 1 при этом из базиса будет выведен вектор A_{n+k}^0 . В дальнейшем, при решении задачи заменить вектор A_k в базисе может только тот же вектор A_{n+k}^0 . Но на основании пункта а) свойства 2 столбец A_{n+k}^0 больше не может быть разрешающим — вектор A_{n+k}^0 больше не будет введен в базис.

Поэтому вектор A_k , однажды введенный в базис, больше не будет выведен из базиса. А так как при решении задачи симплекс-методом в базис может быть введен любой вектор, имеющий отрицательную оценку Δ_j , то свойство 3 доказано.

Свойство 4. Оптимальный базис S-задачи не зависит от величины компонент вектора B правой части ограничений.

Доказательство. На итерациях решения задачи линейного программирования симплекс-методом величина компонент вектора правой части ограничений определяет

разрешающую строку. На основании же пункта б) свойства 2 индекс разрешающей строки равен индексу разрешающего столбца, т. е. выбор оптимального базиса не зависит от правой части ограничений, что и требовалось доказать.

Геометрически свойство 4 проиллюстрировано на рис. 3.1 для случая $n=2$. Изменение компоненты b_i вектора правых частей ограничений соответствует параллельному сдвигу i -й гиперплоскости. Из рис. 3.1 видно, что, как бы мы ни сдвигали параллельно самим себе стороны AB или BC многоугольника $OABC$, оптимум останется по-прежнему в точке A (здесь $b_i \geq 0$ по определению S -задачи).

Аналитически свойство 3 является следствием пункта а) свойства 2, а свойство 4 является следствием пункта б) свойства 2. Как следствие пункта б) свойства 2, можно также утверждать следующее.

Свойство 5. При решении S -задачи симплекс-методом заикливание теоретически невозможно.

Для того чтобы убедиться в этом, достаточно вспомнить, что заикливание характеризуется появлением неоднозначности выбора вектора, подлежащего исключению из базиса [20]. (Заикливание возможно только в том случае, когда в текущем опорном плане не менее двух базисных переменных являются нулевыми, а это вызывает неоднозначность выбора исключаемого из базиса вектора.) На основании же пункта б) свойства 2 вектор, выводимый на данной итерации из базиса, однозначно определяется выбором вводимого в базис вектора, что и доказывает свойство 5.

Свойство 6. Многогранное множество, определяемое ограничениями (1.2), (1.3), не содержит вырожденных опорных планов, если $B > 0$.

Доказательство этого свойства аналогично доказательству свойства 2. Действительно, при решении S -задачи симплекс-методом разрешающий столбец данной итерации имеет все элементы неположительные ($a_{ir}^{(p)} \leq 0$), кроме разрешающего ($a_{rr}^{(p)} > 0$). Поэтому компоненты вектора $B = [b_i]$, $i \neq r$ будут получать в процессе решения только неотрицательные приращения:

$$\left. \begin{aligned} b_i^{(p+1)} &= b_i^{(p)} - \frac{b_r^{(p)} \cdot a_{ir}^{(p)}}{a_{rr}^{(p)}}, & i \neq r, \\ b_r^{(p+1)} &= \frac{b_r^{(p)}}{a_{rr}^{(p)}}, \end{aligned} \right\} \quad (1.6)$$

где $a_{ir}^{(p)} \leq 0$, $a_{ir}^{(p)} > 0$. Отсюда для $b_i > 0$ при $p=1$, $b_i^{(2)} > 0$ и соответственно по индукции $b_i^{(p+1)} > 0$. По определению, опорный план, не содержащий нулевых базисных компонент $x_{i_6}^{(p)} = b_i^{(p)}$, является невырожденным, что и требовалось доказать.

Прежде чем перейти к рассмотрению алгоритмов решения S-задач, нам остается выяснить, в каких случаях целевая функция задачи будет *неограниченной* на допустимом множестве планов. При использовании симплекс-метода целевая функция задачи неограничена на рассматриваемом множестве, если среди компонент $a_{ij}^{(p)}$ вектора, вводимого в базис, не имеется положительных. Заметим, что в S-задаче у вектора, вводимого в базис, положительным может быть только элемент $a_{ij}^{(p)}$ (см. доказательство свойства 2). Для получения компоненты $a_{ij}^{(p)}$ необходимо выразить вводимый в базис вектор через векторы текущего базиса. При решении S-задачи, как мы убедимся в дальнейшем, можно построить алгоритмы решения, в которых не будет требоваться вычисления компонент $a_{ij}^{(p)}$. Поэтому мы рассмотрим некоторые другие критерии, по которым можно будет судить о неограниченности целевой функции S-задачи.

Введем необходимые обозначения. Пусть M — множество индексов векторов A_j матрицы $A = [a_{ij}]$, которые в процессе решения задачи (1.1) — (1.3) симплекс-методом были разрешающими. Обозначим через $\tilde{A}$ квадратную матрицу, состоящую из элементов матрицы A , находящихся на пересечении столбцов и строк с индексами $j \in M$, $i \in M$, $i=j$. Соответственно через $\tilde{C}$ обозначим вектор, состоящий из компонент вектора $C = [c_j]$ с индексами $j \in M$, а через $\tilde{B}$ обозначим вектор, составленный из элементов вектора $B = [b_i]$ с индексами $i \in M$. Введенные обозначения пояснены рис. 3.3. На этом рисунке столбцы с индексами $j \in M$ перенесены в левую часть матрицы, а строки с индексами $i \in M$ перенесены вверх.

Поясним введенные обозначения. Пусть базисная матрица A_b составлена из векторов $\tilde{A}_j$, соответствующих базисным переменным x_j , и единичных векторов, соответствующих базисным переменным x_j^0 (рис. 3.4). Подматрицу векторов $\tilde{A}_j$ удобно разделить на две части: подматрицу $\tilde{A}$, составленную из строк с индексами $i \in M$, и подматрицу $\tilde{A}_b$, составленную из строк индексами $i \notin M$. Что дает такое

разделение матрицы? Вектор Λ_6 двойственных оценок условной задачи, как известно, вычисляется следующим образом:

$$\Lambda_6 = C_6 A_6^{-1},$$

где C_6 — вектор, составленный из компонент вектора целевой функции, соответствующих базисным переменным. Вектор Λ_6 можно находить также из решения системы

$$\Lambda_6 A_6 = C_6.$$

При решении задачи линейного программирования нас интересуют ненулевые компоненты вектора Λ_6 . С помощью матрицы $\tilde{A}$ мы как раз и будем получать подвектор Λ ,

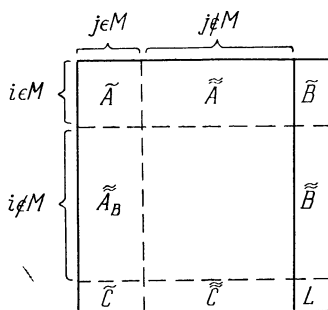


Рис. 3.3. Разделение матрицы при решении S -задачи.

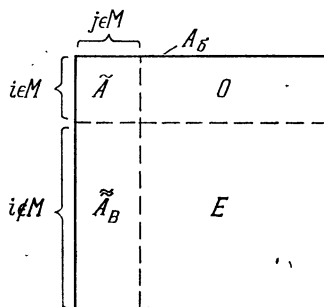


Рис. 3.4. Связь матрицы $\tilde{A}$ с базисной матрицей $\tilde{A}_6$.

составленный только из ненулевых компонент вектора Λ_6 , оперируя тем самым с матрицей $\tilde{A}$, размерность которой обычно значительно меньше размерности матрицы A_6 : $\Lambda = \tilde{C} \tilde{A}^{-1}$ или $\Lambda \tilde{A} = \tilde{C}$. Точно так же при вычислении вектора $X = [x_j]$ использование матрицы $\tilde{A}$ позволяет оперировать с системой значительно меньшего размера, вычисляя при том только подвектор $\tilde{X}$ базисных переменных.

Сформулируем теперь критерии неограниченности целевой функции S -задачи.

Свойство 7. Целевая функция S -задачи ограничена на рассматриваемом множестве планов, если в процессе решения задачи а) вектор $\tilde{X} \geq 0$, где $\tilde{X}$ находится из решения системы $\tilde{A} \tilde{X} = \tilde{B}$; или б) вектор $\Lambda > 0$, где Λ находится из решения сис-

темы $\bar{A}\bar{A}=\bar{C}$, или в) матрица $\bar{A}^{-1}\geq 0$. Если же условие (а) или (б) или (в) не выполняются, или если $|\bar{A}|=0$, то целевая функция S-задачи неограничена на рассматриваемом множестве.

Доказательство. Будем доказывать каждый пункт отдельно.

а) Если в оптимальной точке вектор $\tilde{X}_{\text{опт}}$, полученный при решении системы $\bar{A}_{\text{опт}}\tilde{X}_{\text{опт}}=\bar{B}_{\text{опт}}$, является неотрицательным, значит, оптимум существует — целевая функция задачи ограничена на рассматриваемом множестве планов (т. е. все $a_{ij}^{(p)}>0$). Если же оптимальный вектор $\tilde{X}_{\text{опт}}$ или вектор $\tilde{X}$, получаемый в процессе решения задачи, имеют отрицательные компоненты, значит, имеются члены $a_{ij}^{(p)}<0$ и целевая функция задачи неограничена на рассматриваемом множестве.

Действительно, компоненты $\tilde{x}_j$ вектора базисных переменных x_j при решении задачи симплекс-методом равны соответствующим значениям компонент правой части на данной p -й итерации: $\tilde{x}_j=b_i^{(p)}$, где $j=i$. Из рассмотрения формул (1.6) преобразования компонент правой части следует, что $b_i^{(p+1)}\geq 0$, если $a_{ir}^{(p)}>0$, и наоборот, $a_{ir}^{(p)}>0$, если $b_i^{(p+1)}\geq 0$. Элементы $b_i^{(p+1)}$ могут иметь отрицательные значения только в том случае, если на некоторой итерации появится отрицательный элемент $a_{ir}^{(p)}$. Отсюда ясно, что целевая функция S-задачи неограничена на рассматриваемом множестве планов, если вектор $\tilde{X}$, получаемый из решения системы $\bar{A}\tilde{X}=\bar{B}$, имеет отрицательные компоненты. Если же вектор $\tilde{X}\geq 0$ в процессе решения задачи (т. е. и $\tilde{X}_{\text{опт}}\geq 0$), значит, целевая функция S-задачи ограничена на рассматриваемом множестве планов.

Если система является неопределенной, то это также будет свидетельствовать о неограниченности целевой функции S-задачи. Это вытекает из следующего: случай $|\bar{A}|=0$ означает, что среди разрешающих элементов $a_{ir}^{(p)}$ имеются нулевые, так как определитель

$$|\bar{A}| = a_{11}a_{22}^{(1)}a_{33}^{(2)}, \dots, a_{ll}^{(l-1)},$$

где l — количество строк (столбцов) матрицы A . Что и требовалось доказать.

б) Если в оптимальной точке вектор $\Lambda_{\text{опт}}$, полученный из решения системы $\Lambda_{\text{опт}} \tilde{A}_{\text{опт}} = \tilde{C}_{\text{опт}}$, является неотрицательным, значит, существует оптимальный план двойственной задачи, а целевая функция прямой задачи ограничена на рассматриваемом множестве планов. Если же оптимальный вектор $\Lambda_{\text{опт}}$, или вектор Λ , получаемый в ходе решения задачи из системы $\Lambda \tilde{A} = \tilde{C}$, имеют отрицательные компоненты, то это означает неограниченность целевой функции задачи на рассматриваемом множестве планов.

Действительно, компоненты λ_i вектора Λ на данной p -й итерации равны соответствующим значениям оценок $\Delta_j^{(p)}$ векторов условий, выведенных из базиса в процессе решения, где $i=j$ (см. свойство 2). В то же время из рассмотрения формул (1.5) преобразования оценок $\Delta_j^{(p)}$ мы ранее (при доказательстве свойства 2) убедились в том, что если вектор однажды был выведен из базиса, то его оценка $\Delta_j^{(p+1)}$ остается положительной на всех последующих итерациях (при $a_{rr}^{(p)} > 0$), т. е. $\lambda_i > 0$. И наоборот, $a_{rr}^{(p)} > 0$, если $\Delta_j^{p+1} > 0$, $j=r$ (или $\lambda_i > 0$, $j=r$). Отрицательные значения λ_i могут появиться только в том случае, если на некоторой итерации имеется значение $a_{rr}^{(p)} < 0$.

Таким образом, целевая функция S -задачи неограничена на рассматриваемом множестве планов, если вектор Λ , получаемый из решения системы $\Lambda \tilde{A} = \tilde{C}$, имеет отрицательные компоненты. Если же вектор $\Lambda \geq 0$ в процессе решения задачи, значит, целевая функция ограничена на рассматриваемом множестве, что и требовалось доказать.

в) Из пункта а) следует, что в случае ограниченности целевой функции

$$\tilde{X} = \tilde{A}^{-1} \tilde{B} \geq 0$$

при любом $\tilde{B} \geq 0$. Пусть

$$\tilde{B} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Тогда $\tilde{X} \geq 0$ только, если неотрицательны элементы первого столбца матрицы $\tilde{A}^{-1}$. Рассматривая значения эле-

ментов второго, третьего и т. д. столбцов матрицы $\tilde{A}^{-1}$ при

$$\tilde{B} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ и т. д.,}$$

мы убедимся, что в данном случае $\tilde{A}^{-1} \geq 0$, что и требовалось доказать.

Из проведенного доказательства очевидно также, что невыполнение условия $\tilde{A}^{-1} \geq 0$ означает неограниченность целевой функции S-задачи на рассматриваемом множестве планов. Действительно, элементы $a_{ij}^{(p+1)}$, $i \in M$, $j \in M$ представляют собой элементы обратной матрицы $\tilde{A}^{-1}$ на $(p+1)$ -й итерации. Появление на данной итерации компонент $a_{ij}^{(p+1)} < 0$, $i \in M$, $j \in M$ может случиться, только если $a_{rr}^{(p)} < 0$. Но так как в S-задаче элемент $a_{rr}^{(p)}$ является единственным элементом разрешающей строки, который может быть положительным, то это означает неограниченность целевой функции.

Свойство 7 доказано полностью.

В заключение рассмотрим два частных случая S-матриц, часто встречающихся на практике: случай 1 — матрица A симметричная и положительно определенная; случай 2 — матрица A удовлетворяет условиям Адамара [19]

$$\left. \begin{aligned} a_{jj} &> \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}|, & j = 1, \dots, n, \\ \text{или} \\ a_{ii} &> \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, & i = 1, \dots, n. \end{aligned} \right\} \quad (1.7)$$

Ниже мы опишем особенности S-задач с подобными матрицами ограничений. Будет доказано, что рассматриваемые типы S-матриц обладают следующими свойствами: а) элементы $a_{rr}^{(p)} > 0$; б) матрица $A^{-1} \geq 0$. Поэтому в рассматриваемом случае при решении S-задачи не требуется проверять неограниченность целевой функции.

даться с помощью матричных степенных рядов:

$$(E-P)^{-1} = \sum_{k=1}^{\infty} P^k = E + P + P^2 + P^3 + \dots \quad (1.10)$$

Разложение (1.10) следует из общих свойств обратных операторов в псевдометрических пространствах [42, стр. 96].

Приведем доказательство формулы (1.10) для случая, когда оператором является матрица P [85]. Если разложение (1.10) верно, то

$$(E-P)(E-P)^{-1} = (E-P)(E+P+P^2+\dots) = E. \quad (1.11)$$

Заметим, что

$$(E-P)(E+P+P^2+\dots+P^k) = E - P^{k+1}.$$

Поэтому для доказательства выражения (1.11) достаточно убедиться, что при условиях (1.8), (1.9)

$$\lim_{k \rightarrow \infty} (E - P^{k+1}) = E$$

или

$$\lim_{k \rightarrow \infty} (-P^{k+1}) = 0. \quad (1.12)$$

Докажем (1.12). Элементы матрицы $P^2 = [p_{ij}^{(2)}]$ образуются следующим образом:

$$p_{ij}^{(2)} = \sum_{s=1}^n p_{is} p_{sj}.$$

Если обозначить $\sum_{i=1}^n p_{ij}$ через r_j , и ввести элемент r , для которого из (1.9)

$$\sum_{i=1}^n p_{ij} = r_j \leq r < 1,$$

то получим

$$\begin{aligned} \sum_{i=1}^n p_{ij}^{(2)} &= \sum_{i=1}^n \sum_{s=1}^n p_{is} p_{sj} = \sum_{s=1}^n \left(\sum_{i=1}^n p_{is} \right) p_{sj} = \\ &= \sum_{s=1}^n r_s p_{sj} \leq r \sum_{s=1}^n p_{sj} \leq r^2. \end{aligned} \quad (1.13)$$

Из условий (1.8)

$$p_{ij}^{(2)} \leq \sum_{i=1}^n p_{ij}^{(2)}, \quad (1.14)$$

так как $p_{is} \leq \sum_{i=1}^n p_{is}$ при $p_{is} \geq 0$, откуда

$$\sum_{i=1}^n p_{ij}^{(2)} = \sum_{s=1}^n \left(\sum_{i=1}^n p_{is} \right) p_{sj} \geq \sum_{s=1}^n p_{is} p_{sj} = p_{ij}^{(2)}.$$

На основании (1.13), (1.14) можно записать

$$p_{ij}^{(2)} \leq r^2.$$

Проделав аналогичные выкладки для $p^{(3)}, p^{(4)}, \dots, p^{(k)}$, получим

$$p_{ij}^{(k)} \leq r^k. \quad (1.15)$$

Таким образом, любой элемент $p_{ij}^{(k)}$ матрицы P^k стремится к нулю при $k \rightarrow \infty$ (так как в (1.15) $r < 1$). Условие (1.12) доказано. Следовательно, разложение (1.10) верно.

Матрица $(E - P)^{-1} \geq 0$, так как все матрицы ряда (1.10) неотрицательные. Итак, в данном случае

$$\tilde{A}^{-1} \geq 0, \quad (1.16)$$

что и требовалось доказать.

Из условия (1.16) следует положительность элементов $a_{jl}^{(p)}$. Действительно, пусть имеется матрица $\tilde{A}$ произвольного l -го порядка, удовлетворяющая условиям (1.16). Элемент b_{ll} обратной матрицы $\tilde{A}^{-1}$ может быть получен следующим образом:

$$b_{ll} = \frac{1}{a_{ll}^{(l-1)}}.$$

Если из (1.16) $b_{ll} \geq 0$, то $a_{ll}^{(l-1)} \geq 0$. Матрица $\tilde{A}$ неособенная, так как удовлетворяются условия Адамара (1.7). Поэтому $a_{ll}^{(l-1)} > 0$, что и требовалось доказать.

§ 2. Алгоритмы решения S-задач. Примеры

1. Модификации симплекс-метода. Специальные свойства S-задач позволяют упростить использование симплекс-метода для решения. На основании свойства 4 решение задачи можно разделить на два этапа: качественное

решение (определение оптимального базиса) и количественное решение (определение значений переменных в оптимальной точке). Нахождение оптимального базиса на основании свойства 1 заключается в определении, какая из двух переменных x_j, x_i^0 ($i=j$) в (1.1) — (1.3) является положительной в оптимальной точке. В исходной точке все $x_j=0$ все $x_i^0>0$. Для нахождения оптимального базиса достаточно определить, какие $x_j>0$ в оптимальной точке, причем на основании свойства 3 решение задачи ведется только в одну сторону, т. е. мы все время увеличиваем число положительных переменных x_j , пока не выявим все переменные, являющиеся положительными в оптимальной точке.

Итак, исходный опорный план $x_i^0=b_i$, исходные оценки внебазисных векторов [93] $\Delta_j=-c_j$. На основании свойства 3 векторы, имеющие коэффициенты $c_j>0$, обязательно войдут в оптимальный базис. Эти вектора обозначаются, как мы уже говорили, $\tilde{A}_j$. Их удобнее ввести в базис одновременно, так как мы знаем, где находятся разрешающие элементы (пункт б) свойства 2), и знаем, что из базиса они больше не будут выведены (свойство 3). Однако не будем вводить их в базис, пока не выявим все внебазисные векторы, оценки которых стали бы отрицательными, если бы мы ввели в базис векторы $\tilde{A}_j$. Для этого следует определить знаки элементов строки оценок $\tilde{\Delta}_j$ внебазисных векторов следующей итерации:

$$\tilde{\Delta} = -\tilde{C} + \Lambda \tilde{A}, \quad (2.1)$$

т. е. проверить знаки неравенств

$$\Lambda \tilde{A} \leq C, \quad (2.2)$$

где Λ находится из системы

$$\Lambda \tilde{A} = \tilde{C}. \quad (2.3)$$

Если знаки неравенств (2.2) « $\geq$ », значит, получен оптимум; если же имеются векторы $\tilde{A}_j$, имеющие отрицательные оценки, то, добавив их к векторам $\tilde{A}_j$, вновь определяем знаки неравенств (2.2) и т. д.

Таким образом, определяя (без преобразования самой матрицы) знаки элементов строки оценок внебазисных векторов, мы как бы «нащупываем» оптимальный базис (определяем все множество векторов $\tilde{A}_j$), а затем сразу получаем

решение задачи, вводя одновременно все векторы $\tilde{A}_j$ в базис. При этом процесс «нащупывания» оптимального базиса, насколько показали вычислительные эксперименты по решению задач оптимизации нефтедобычи, сходится очень быстро — за одну, максимум — три итерации.

Конкретно, алгоритм нахождения оптимального базиса задачи (1.1) — (1.3), заключающийся в определении индексов ненулевых *) переменных x_j в оптимальной точке, сводится к следующему. Решение начинается с точки, в которой все переменные x_j — нулевые. Выделяется множество M индексов переменных x_j с положительными коэффициентами c_j . Переменные с индексами $j \in M$ будут в оптимальной точке ненулевыми. Из элементов a_{ij} , c_j с индексами $i \in M$, $j \in M$ формируется матрица $\tilde{A}$, вектор $\tilde{C}$ и решается система линейных алгебраических уравнений (2.3). Затем проверяются оценки (2.1) (или неравенства (2.2)) для остальных переменных x_j . Если все оценки Δ_j оказываются неотрицательными, значит, найден оптимальный базис. Если же имеются переменные x_j с отрицательными оценками $\tilde{\Delta}_j$, то эти переменные также следует отнести к ненулевым. Для этого следует расширить множество M за счет индексов данных переменных. Матрица $\tilde{A}$ при этом окаймляется новыми строками и столбцами с индексами $i \in M$, $j \in M$. Добавляются также соответствующие элементы c_j , $j \in M$ к вектору $\tilde{C}$. Затем пересчитывается решение системы (2.3), полученное на предыдущей итерации, вновь проверяются оценки $\tilde{\Delta}_j$ оставшихся нулевых переменных x_j .

Заметим, что для формирования подматрицы $\tilde{A}$, чтобы не переносить строки и столбцы матрицы A в верхний левый угол матрицы (как это сделано на рис. 3.3), удобно при ручном счете применять вычеркивание строк и столбцов с индексами $i \in M$, $j \in M$. Это же касается и векторов $\tilde{B}$, $\tilde{C}$.

Для выявления случаев неограниченности целевой функции задачи удобнее всего в данном случае использовать критерий, приведенный в свойстве 76 предыдущего параграфа.

Блок-схема описанного алгоритма приведена на рис. 3.5.

*) В том случае, когда имеются компоненты $b_i = 0$ в правой части ограничений, более точно будет говорить «базисные» переменные вместо «ненулевые».

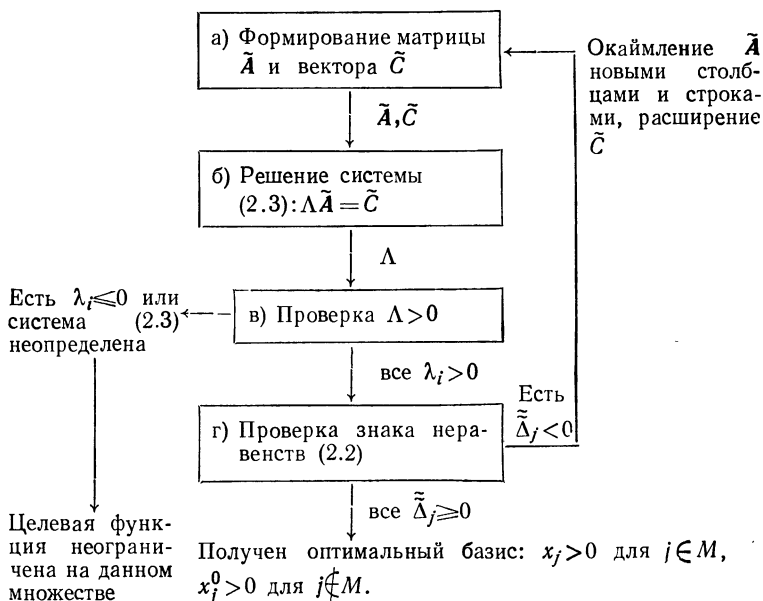


Рис. 3.5. Блок-схема алгоритма определения оптимального базиса задачи (1.1) — (1.3).

Назовем *итерацией* один полный шаг сверху вниз по блок-схеме, изображенной на рис. 3.5. Решение системы линейных алгебраических уравнений $\Lambda \tilde{A} = \tilde{C}$ на данной итерации следует начинать от значения Λ , полученного на предыдущей итерации, так как матрица $\tilde{A}^{(p)}$ данной итерации получается путем окаймления матрицы $\tilde{A}^{(p-1)}$ предыдущей итерации. Все предыдущие итерации являются, по сути дела, отдельными шагами относительно решения системы линейных уравнений (2.3), последней итерации, так как решение задачи ведется только в одну сторону (множество M и соответственно, матрица $\tilde{A}$ могут только расширяться). Поэтому общая трудоемкость нахождения оптимального базиса задачи определяется количеством вычислений, необходимых для решения системы линейных алгебраических уравнений (2.3) последней итерации. Добавляется, кроме этого, только операция проверки знака неравенств (2.2).

Проверку знака неравенств (2.2) и, соответственно, выявление переменных x_j , переводимых в ненулевые, можно осуществить не только после получения точного решения системы (2.3) на каждой итерации, но и непосредственно в процессе решения системы. Для этого достаточно использовать при решении системы (2.3) метод, обеспечивающий в процессе решения монотонное возрастание *) вектора Λ от нулевого значения.

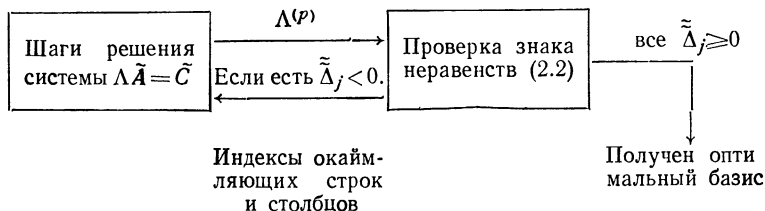
Действительно, если Λ монотонно возрастает, то вектор оценок $\tilde{\Delta}(\Lambda)$ будет в процессе решения системы (2.3) монотонно уменьшаться, так как $\tilde{A} \leq 0$, $\Lambda > 0$. Монотонное уменьшение $\tilde{\Delta}(\Lambda)$ при увеличении Λ обеспечивает то, что если $\tilde{\Delta}_j(\Lambda) < 0$ на данном шаге решения системы (2.3), то тем более $\tilde{\Delta}_j(\Lambda) < 0$ на остальных шагах. То есть, если при проверке неравенств (2.2) на данном шаге решения системы оказывается необходимым включение некоторых переменных в число ненулевых, то эти переменные тем более должны быть в числе ненулевых на остальных шагах решения системы (2.3) и, в частности, при значении Λ , соответствующем точному решению.

В том случае, если неравенства (2.2) проверяются параллельно с решением системы (2.3), не требуется проверять положительность компонент вектора Λ . Это обеспечивается выбором метода решения системы $\Lambda \tilde{A} = \tilde{C}$, позволяющим получить монотонное возрастание Λ по шагам решения системы, начиная от нулевого значения Λ . Например, если при решении системы $\Lambda \tilde{A} = \tilde{C}$ используется метод Гаусса, то компоненты $\lambda_i^{(p)}$ будут возрастать на каждом шаге при $a_{ii}^{(p)} > 0$.

Итак, если система $\Lambda \tilde{A} = \tilde{C}$ решается методом, обеспечивающим монотонное возрастание Λ , то проверка знака неравенств (2.2) может осуществляться параллельно с решением системы; матрица $\tilde{A}$ будет окаймляться новыми строками и столбцами непосредственно в процессе решения системы и процедура примет вид, изображенный на рис. 3.6.

*) Под «монотонным возрастанием» в данном случае понимается то, что на каждом шаге процедуры хотя бы одна компонента вектора Λ возрастает, а все остальные компоненты не убывают.

Алгоритм, позволяющий проводить проверку неравенств (2.2), не получая точного решения системы (2.3), имеет особое преимущество в том случае, когда для решения системы (2.3) используются итерационные методы. Это объясняется тем, что применение итерационных методов обычно позволяет за относительно небольшое число шагов получить неплохое приближение к решению системы. Полученное приближение используется для проверки неравенств (2.2),



[Рис. 3.6. Блок-схема алгоритма с параллельной проверкой неравенств (2.2).

т. е. для выявления новых переменных, переводимых в ненулевые. Затем матрица A окаймляется и продолжается решение. Таким образом достаточно быстро определяется оптимальный базис и только тогда уже с требуемой точностью находится решение системы.

2. Итерационные методы *). Известные итерационные методы решения задач линейного программирования (например, методы, связанные с использованием бесконечных итерационных процедур решения матричных игр) обладают крайне медленной скоростью сходимости. Для решения S-задач могут быть предложены итерационные процедуры, являющиеся обобщением на задачи линейного программирования значительно более отработанных итерационных методов решения систем линейных алгебраических уравнений.

В случае использования метода последовательных приближений итерационный процесс выделения оптимального базиса задачи (1.1) — (1.3) примет вид

$$\lambda_i^{(p+1)} = \beta_j + \sum_{\substack{i \in M \\ i \neq j}} \alpha_{ij} \lambda_i^{(p)}, \quad j = 1, \dots, n, \quad (2.4)$$

*) В линейном программировании часто употребляют термин *итеративные методы*.

где $\beta_j = c_j/a_{jj}$, $\alpha_{ij} = -a_{ij}/a_{jj}$, M — множество индексов положительных переменных $\lambda_i^{(p)}$. Как видно из (2.4), предлагаемое обобщение сводится к тому, что если при решении систем линейных алгебраических уравнений суммирование в итерационной формуле проводится по всем i , то для решения рассматриваемой задачи линейного программирования достаточно проводить суммирование только положительных переменных. При остановке процесса (2.4) будут получены двойственные оценки оптимального плана $\lambda_i = \lambda_i^{(p)} > 0$, $i \in M$; $\lambda_i = 0$, $i \notin M$; в оптимальный базис будут входить векторы с индексами $j \in M$.

Приведем обоснование процедуры и условия сходимости. Допустим, что S -задача решается алгоритмом, предложенным в предыдущем пункте. Пусть, далее, для решения системы (2.3) используется метод простой итерации. Процесс решения системы в этом случае записывается следующим образом:

$$\lambda_j^{(p+1)} = \beta_j + \sum_{\substack{i \in M \\ i \neq j}} \alpha_{ij} \lambda_i^{(p)}, \quad j \in M, \quad (2.5)$$

где

$$\alpha_{ij} = -\frac{a_{ij}}{a_{jj}} \geq 0, \quad i \in M, \quad j \in M \quad (a_{ij} \leq 0, \quad a_{jj} > 0),$$

$$\beta_j = \frac{c_j}{a_{jj}}, \quad j \in M, \lambda_i^0.$$

Проверка неравенств (2.2) будет вестись по аналогичным формулам

$$\tilde{\Delta}_j^{(p+1)} = \beta_j + \sum_{i \in M} \alpha_{ij} \lambda_i^{(p)} \geq 0, \quad j \notin M, \quad (2.6)$$

где

$$\alpha_{ij} = -\frac{a_{ij}}{a_{jj}} \geq 0, \quad i \notin M.$$

Переменные, имеющие оценки $\tilde{\Delta}_k^{(p+1)} > 0$, должны быть включены в число ненулевых — индексы k включаются в множество индексов M и в системе (2.5) появляются новые строки:

$$\lambda_k^{(p+1)} = \beta_k + \sum_{\substack{i \in M \\ i \neq k}} \alpha_{ik} \lambda_i^{(p)}.$$

Ранее эти строки входили в систему (2.6) и поэтому $\lambda_k^{(p+1)} = \tilde{\Delta}_k^{(p+1)} > 0$. При этом в имеющиеся строки системы (2.5) добавляются компоненты $\alpha_{ik}\lambda_k$, где $\lambda_k = \tilde{\Delta}_k > 0$. Таким образом, в условиях (2.5) всегда $\lambda_i^{(p)} \geq 0$, $\alpha_{ij} \geq 0$. Поэтому компоненты вектора Λ_i будут монотонно возрастать по шагам решения системы (2.3), и проверять условия (2.6) можно, не получая точного решения системы (как в алгоритме с параллельной проверкой неравенств (2.2)).

Процедура (2.5), (2.6) может быть записана одной формулой, если обозначить $\tilde{\Delta}_j^{(p)} = \lambda_j^{(p)}$ для $j \notin M$. Эта запись имеет вид (2.4). Запись достаточно очевидна, так как переменная $\lambda_k^{(p)}$ входит в правую часть условий (2.5), (2.6) только в том случае, если

$$\tilde{\Delta}_k^{(p)} = \lambda_k^{(p)} > 0.$$

Итерационный процесс (2.4) заканчивается, когда значения положительных переменных $\lambda_i^{(p)}$ ($i \in M$) меняются по итерациям в пределах заданной точности.

Связь процедуры (2.4) решения задачи линейного программирования с аналогичной процедурой решения систем линейных уравнений очевидна. Действительно, процесс метода итераций при решении некоторой n -мерной системы линейных уравнений $\Lambda A = C$ с квадратной матрицей A записывается

$$\lambda_j^{(p+1)} = \beta_j + \sum_{\substack{i=1 \\ i \neq j}}^n \alpha_{ij} \lambda_i^{(p)}, \quad j = 1, \dots, n. \quad (2.7)$$

Отличие процедуры (2.4) от (2.7) только в том, что при вычислении $(p+1)$ -го приближения переменной $\lambda_j^{(p+1)}$ учитываются только те из вычисленных ранее переменных $\lambda_i^{(p)}$, которые положительны.

Процедуры, аналогичные (2.4), могут быть построены при использовании ряда других итерационных методов. Пусть используется, например, метод Зейделя. Обычные формулы последовательных приближений для решения n -мерной системы линейных уравнений $\Lambda A = C$ по методу Зейделя имеют вид

$$\lambda_j^{(p+1)} = \beta_j + \sum_{i=1}^{j-1} \alpha_{ij} \lambda_i^{(p+1)} + \sum_{i=j+1}^n \alpha_{ij} \lambda_i^{(p)}, \quad j = 1, \dots, n.$$

Та же процедура для решения S -задачи линейного программирования запишется в форме

$$\lambda_j^{(p+1)} = \beta_j + \sum_{\substack{i \in M \\ i < j}} \alpha_{ij} \lambda_i^{(p+1)} + \sum_{\substack{i \in M \\ i > j}} \alpha_{ij} \lambda_i^{(p)}, \quad j = 1, \dots, n. \quad (2.8)$$

Таким образом, для решения задачи линейного программирования (1.1) — (1.3) итерационным методом достаточно запрограммировать простую итерационную процедуру типа (2.4) или (2.8). Скорость сходимости при решении задачи (1.1) — (1.3) в этом случае равняется скорости сходимости итерационного метода решения системы линейных уравнений $\Lambda \tilde{A} = \tilde{C}$.

Из обоснования предлагаемой итерационной процедуры очевидны и условия сходимости: процесс (2.4) (или (2.8)) сходится к решению соответствующей задачи линейного программирования, если сходится итерационный процесс решения любой системы линейных уравнений (2.5). Очевидно, это имеет место, если условия сходимости выполняются для максимальной возможной матрицы $\tilde{A}$ данной системы, т. е. для матрицы $\tilde{A}_m$, составленной из элементов исходной матрицы ограничений A без строк и столбцов, все элементы которых неположительные. Строки с неположительными элементами не рассматриваются, так как условия

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{при } a_{ij} \leq 0, x_j \geq 0, b_i \geq 0$$

не могут ограничивать переменные x_j . Столбцы с неположительными элементами не могут входить в матрицу $\tilde{A}$, так как если оказывается, что переменные x_j , соответствующие подобным столбцам, должны быть переведены в ненулевые, то это будет означать неограниченность целевой функции на рассматриваемом множестве ограничений. Данный случай неограниченности целевой функции должен учитываться при составлении алгоритма.

Для рассматриваемых в главе 1 объектов условия сходимости итерационных методов обычно выполняются. Это — объекты, описываемые уравнениями Лапласа, Пуассона, типа Лапласа, типа Пуассона, электрические цепи, балансовые задачи экономики и т. д. Для указанных объ-

ектов матрица $\tilde{A}_m$ совпадает с исходной матрицей ограничений. Наиболее употребительные условия сходимости имеют вид [82]

$$|a_{ii}| > \sum_{j=1}^n |a_{ij}|, \quad i = 1, \dots, n.$$

Для объектов, описываемых уравнениями типа Лапласа, Пуассона (после конечно-разностной аппроксимации) и для электрических цепей удобнее использовать другие условия сходимости:

— итерационный метод для системы $\Lambda \tilde{A} = \tilde{C}$ сходится, если матрица $\tilde{A}_m$ симметрична и положительно определена.

3. Количественное решение. Выше все время шла речь о выделении оптимального базиса задачи линейного программирования (1.1) — (1.3), т. е. о качественном решении. При известном качественном решении легко находится и количественное решение задачи (значения переменных x_j в оптимальной точке):

$$\tilde{A}_{\text{опт}} \tilde{X}_{\text{опт}} = \tilde{B}_{\text{опт}}. \quad (2.9)$$

Таким образом, для решения задачи линейного программирования (1.1) — (1.3) алгоритмом, приведенном в пункте 1, по сути дела, требуется решить две системы линейных алгебраических уравнений ((2.3) и (2.9)), матрицы которых при одинаковой записи являются транспонированными друг относительно друга. Система $\Lambda \tilde{A} = \tilde{C}$ (2.3) решается для нахождения качественного решения задачи. Система $\tilde{A}_{\text{опт}} \tilde{X}_{\text{опт}} = \tilde{B}_{\text{опт}}$ (2.9) решается для нахождения количественного решения. В процессе решения системы (2.3) происходит проверка неравенств (2.2) и окаймление матрицы $\tilde{A}$ соответствующими строками и столбцами до получения матрицы $\tilde{A}_{\text{опт}}$ оптимального базиса.

В случае применения итерационных методов процесс качественного решения S-задачи записывается простой итерационной формулой типа (2.4) или (2.8). В этом случае и при нахождении количественного решения удобно использовать соответствующий итерационный метод для решения системы (2.9).

4. Примеры.

Пример 1. Решим следующую S -задачу:

$$\begin{array}{ccccccccc}
 & \downarrow & & & \downarrow & & & & \\
 5x_1 - & 3x_2 + & x_3 - & 4x_4 - & 5x_5 \rightarrow \max, \\
 \rightarrow 10x_1 - & 0,5x_2 - & 5x_3 - & 4x_4 & & \leq 30, \leftarrow \\
 -2x_1 + & 10x_2 - & 4x_3 - & x_4 - & 2x_5 \leq 0, \\
 \rightarrow -2x_1 & & + 11x_3 - & 8x_4 - & 2x_5 \leq 50, \leftarrow \\
 -5x_1 - & x_2 - & x_3 + & 14x_4 - & x_5 \leq 10, \\
 & \uparrow & & & \uparrow & & & & \\
 & & x_1, x_2, x_3, x_4, x_5 \geq 0.
 \end{array}$$

Решение проводим в соответствии с блок-схемой рис. 3.5.

Первая итерация.

1а) Множество $M = \{1; 3\}$ индексов переменных с положительными коэффициентами c_j . Отметим стрелками строки и столбцы исходной таблицы с этими индексами. Тогда элементы матриц, находящиеся на пересечении отмеченных строк и столбцов, образуют подматрицу $\tilde{A}$, а отмеченные элементы строки C образуют вектор $\tilde{C}$:

$$\tilde{A} = (a_{ij}), \quad i \in M, \quad j \in M, \quad \tilde{C} = [c_j], \quad j \in M;$$

$$\tilde{A} = \begin{bmatrix} 10 & -5 \\ -2 & 11 \end{bmatrix}, \quad C = [5 \quad 1].$$

1б) Решаем систему $\tilde{A}^T \Lambda^T = \tilde{C}^T$ (T — знак транспонирования):

$$\begin{aligned}
 10\lambda_1 - 2\lambda_3 &= 5, & -5\lambda_1 + 11\lambda_3 &= 1, \\
 \lambda_1 &= 0,57, & \lambda_3 &= 0,35.
 \end{aligned}$$

1в) Все $\lambda_i > 0$.

1г) Проверяем неравенства $\tilde{A}^T \Lambda^T \leq \tilde{C}^T$, где подматрица $\tilde{A}$ составляется из элементов матрицы A , находящихся на пересечении невыделенных столбцов и выделенных строк; вектор $\tilde{C}$ составляется из невычеркнутых элементов строки C : $\tilde{A} = [a_{ij}], \quad i \in M, \quad j \notin M; \quad \tilde{C} = [c_j], \quad j \notin M;$

$$\begin{aligned}
 \tilde{A} &= \begin{bmatrix} -0,5 & -4 & -0 \\ -0 & -8 & -2 \end{bmatrix}, \\
 \tilde{C} &= [-3 \quad -4 \quad -5];
 \end{aligned}$$

неравенства имеют вид

$$\begin{aligned} -0,5\lambda_1 - 0 \cdot \lambda_3 &> -3, \\ -4\lambda_1 - 8\lambda_3 &< -4, \\ -0 \cdot \lambda_1 - 2\lambda_3 &> -5. \end{aligned}$$

Подстановка $\lambda_1=0,57$, $\lambda_3=0,35$ показывает, что переменную x_1 также следует перевести в ненулевые, так как выполняется со знаком $<$ соответствующее неравенство. Расширяем за счет нее множество $M=\{1; 3; 4\}$ и переходим к следующей итерации.

Вторая итерация

2а) Выделим дополнительно в исходной таблице строки и столбцы с индексами, добавленными в множество M на предыдущей итерации (выделим четвертую строку и столбец). Матрица $\tilde{A}$ окаймляется дополнительным столбцом и строкой и имеет вид

$$\tilde{A} = \begin{bmatrix} 10 & -5 & -4 \\ -2 & 11 & -8 \\ -5 & -1 & 14 \end{bmatrix},$$

а к вектору $\tilde{C}$ добавляется соответствующий элемент:

$$\tilde{C} = [5 \quad 1 \quad -4].$$

2б) Решаем систему

$$\begin{aligned} 10\lambda_1 - 2\lambda_3 - 5\lambda_4 &= 5, \\ -5\lambda_1 + 11\lambda_3 - 1 \cdot \lambda_4 &= 1, \\ -4\lambda_1 - 8\lambda_3 + 14\lambda_4 &= -4, \end{aligned}$$

начиная от точки решения системы пункта 1б). При использовании метода исключения для этого достаточно одного шага:

$$\lambda_1 = 0,64, \quad \lambda_3 = 0,39, \quad \lambda_4 = 0,12.$$

2в) Все $\lambda_i > 0$.

$$2г) \quad \tilde{\tilde{A}} = \begin{bmatrix} -0,5 & -0 \\ -0 & -2 \\ -1 & -1 \end{bmatrix}, \quad \tilde{\tilde{C}} = (-3 \quad -5).$$

Неравенства

$$-0,5\lambda_1 - 0 \cdot \lambda_3 - \lambda_4 > -3, \quad -0 \cdot \lambda_1 - 2\lambda_3 - \lambda_4 > -5.$$

Все неравенства при подстановке λ_i имеют знак $>$ ($\tilde{\Delta}_j \geq 0$).
Получен оптимальный базис.

Итак, качественное решение задачи — переменные x_1 , x_3 , x_4 ненулевые, ограничения 1-е, 3-е и 4-е обращаются в равенства.

Для получения количественного решения нужно найти решение системы (2.9), матрицей которой является подматрица $\tilde{A}$ последней итерации качественного решения, а вместо вектора $\tilde{C}$ в правой части стоит вектор $\tilde{B}$:

$$\begin{aligned} 10x_1 - 5x_3 - 4x_4 &= 30, \\ -2x_1 + 11x_3 - 8x_4 &= 50, \\ -5x_1 - 1 \cdot x_3 + 14x_4 &= 10, \\ x_1 &= 10, \quad x_3 = 10, \quad x_4 = 5. \end{aligned}$$

Ответ. Вектор

$$X_{\text{опт}} = \begin{bmatrix} 10 \\ 0 \\ 10 \\ 5 \\ 0 \end{bmatrix}.$$

Оптимальное значение целевой функции

$$L_{\text{опт}} = CX_{\text{опт}} = 39,9.$$

Пример 2. Решим рассмотренный выше пример, проводя проверку неравенств (2.2) параллельно с решением системы (2.3) (см. блок-схему рис. 3.6). Пусть для решения системы (2.3) используется метод итераций. Сходимость метода в данном случае обеспечена. Действительно, матрица $\tilde{A}$ максимальной размерности составляется из коэффициентов строк и столбцов исходной матрицы с индексами $i=1, 2, 3, 4$; $j=1, 2, 3, 4$. Пятый столбец не может быть включен в матрицу $\tilde{A}$, так как все его коэффициенты отрицательны, и если окажется, что переменную x_5 следует перевести в ненулевые, то это будет означать неограниченность целевой функции данной задачи. Коэффициенты главной диагонали матрицы $\tilde{A}_m$ по каждой строке больше суммы абсолютных

значений остальных элементов строки:

$$\tilde{A}_m = \begin{bmatrix} +10 & -0,5 & -5 & -4 \\ -2 & +10 & -4 & -1 \\ -2 & -0 & +11 & -8 \\ -5 & -1 & -1 & +14 \end{bmatrix},$$

т. е. условия сходимости метода итераций выполнены. Исходная система (2.3) записывается как и в примере 1:

$$\begin{aligned} M &= \{1, 3\}, \\ 10\lambda_1 - 2\lambda_3 &= 5, \\ -5\lambda_1 + 11\lambda_3 &= 1. \end{aligned}$$

Приведем эту систему к виду, удобному для использования метода итераций:

$$\lambda_1 = \frac{1}{10}(5 + 2\lambda_3), \quad \lambda_2 = \frac{1}{11}(1 + 5\lambda_1).$$

Пусть исходные значения переменных будут нулевые:

$$\lambda_1^{(0)} = 0, \quad \lambda_3^{(0)} = 0.$$

Первый шаг решения системы —

$$\lambda_1^{(1)} = \frac{1}{10} \cdot 5 = 0,5, \quad \lambda_3^{(1)} = \frac{1}{11} \cdot 1 = 0,09.$$

Второй шаг

$$\lambda_1^{(2)} = \frac{1}{10}(5 + 2 \cdot 0,09) = 0,52, \quad \lambda_3^{(2)} = \frac{1}{11}(1 + 5 \cdot 0,5) = 0,32.$$

Пусть проверка знака неравенств (2.2) будет производиться сначала через каждые два шага решения системы, а потом реже, например, через шесть шагов. Неравенства (2.2) при исходном множестве $M = \{1; 3\}$ записываются (как в примере 1):

$$\begin{aligned} -0,5\lambda_1 - 0 \cdot \lambda_3 &\leq -3, \\ -4\lambda_1 - 8\lambda_3 &\leq -4, \\ -0 \cdot \lambda_1 - 2\lambda_3 &\leq -5. \end{aligned}$$

Подставив в эти неравенства значения $\lambda_i = \lambda_i^{(p)}$, $i \in M$, получим:

$$\begin{aligned} -0,5 \cdot 0,52 - 0 &= -0,26 > -3, \\ -4 \cdot 0,52 - 8 \cdot 0,32 &= -4,64 < -4, \\ -0 &- 2 \cdot 0,32 = -0,64 > -5. \end{aligned}$$

Неравенство, соответствующее 4-му столбцу матрицы A , выполняется со знаком $<$. Значит, нужно расширить множество индексов M : $M = \{1, 3, 4\}$. Система (2.3) окаймляется элементами 4-й строки и столбца:

$$\begin{aligned} 10\lambda_1 - 2\lambda_3 - 5\lambda_4 &= 5, \\ -5\lambda_1 + 11\lambda_3 - 1\lambda_4 &= 1, \\ -4\lambda_1 - 8\lambda_3 + 14\lambda_4 &= -4 \end{aligned}$$

или в виде, удобном для итераций,

$$\begin{aligned} \lambda_1 &= \frac{1}{10}(5 + 2\lambda_3 + 5\lambda_4), \\ \lambda_3 &= \frac{1}{11}(1 + 5\lambda_1 + 1\lambda_4), \\ \lambda_4 &= \frac{1}{14}(-4 + 4\lambda_1 + 8\lambda_3). \end{aligned}$$

Продолжаем решение уже расширенной системы (2.3), приняв $\lambda_4^{(2)} = 0$. Тогда

$$\begin{aligned} \lambda_1^{(3)} &= \frac{1}{10}(5 + 0 + 2 \cdot 0,32 + 0) = 0,564, \\ \lambda_3^{(3)} &= \frac{1}{11}(1 + 5 \cdot 0,52 + 0 + 0) = 0,327, \\ \lambda_4^{(3)} &= \frac{1}{14}(-4 + 4 \cdot 0,52 + 8 \cdot 0,32 + 0) = 0,046. \end{aligned}$$

Четвертый шаг —

$$\begin{aligned} \lambda_1^{(4)} &= \frac{1}{10}(5 + 0 + 2 \cdot 0,327 + 5 \cdot 0,046) = 0,584, \\ \lambda_3^{(4)} &= \frac{1}{11}(1 + 5 \cdot 0,564 + 0 + 1 \cdot 0,046) = 0,351, \\ \lambda_4^{(4)} &= \frac{1}{14}(-4 + 4 \cdot 0,564 + 8 \cdot 0,327 + 0) = 0,062. \end{aligned}$$

Вновь проверим неравенства (2.2). Здесь $M = \{1, 3, 4\}$ и неравенства имеют вид

$$-0,5\lambda_1 - 0 - \lambda_4 \geq -3, \quad -0 - 2\lambda_3 - \lambda_4 \geq -5.$$

Подстановка $\lambda_i = \lambda_i^{(4)}$, $i \in M$ показывает, что все неравенства выполняются со знаком $>$. Значит, множество M не меняется. Продолжаем решение той же системы (2.3) (табл. 3.1). Через пять шагов вновь проверяем неравенства

Т а б л и ц а 3.1

Решение системы (2.3) итеративным методом

p	$\lambda_1^{(p)}$	$\lambda_2^{(p)}$	$\lambda_3^{(p)}$	$\lambda_4^{(p)}$
0	0	—	0	—
1	0,5	—	0,09	—
2	0,52	—	0,32	0
3	0,564	—	0,327	0,046
4	0,584	—	0,351	0,062
5	0,601	—	0,362	0,082
6	0,613	—	0,372	0,093
7	0,621	—	0,378	0,102
8	0,627	—	0,382	0,108
9	0,630	—	0,386	0,112
10	0,633	—	0,387	0,115
11	0,635	—	0,389	0,116
12	0,636	—	0,390	0,118

(2.2) при $M = \{1, 3, 4\}$, подставив $\lambda_i = \lambda_i^{(p)}$, $i \in M$. Все неравенства выполняются со знаком $>$. То же самое имеет место и при подстановке в неравенства точного решения системы $\lambda_i = \lambda_i^{(12)}$.

Итак, $M_{\text{опт}} = \{1, 3, 4\}$, т. е. качественное решение задачи заключается в следующем: переменные $x_1 > 0$, $x_3 > 0$, $x_4 > 0$, остальные переменные нулевые; ограничения 1, 3, 4 выполняются как равенства, остальные ограничения — неравенства.

Количественное решение, как и ранее, будет заключаться в решении системы $\tilde{A}_{\text{опт}} \tilde{X} = \tilde{B}_{\text{опт}}$. Так как при качественном решении мы использовали метод итераций, то и при количественном решении для решения системы имеет смысл использовать тот же метод.

Пример 3. Рассмотрим теперь решение того же примера итерационным методом (см. пункт 2 данного параграфа). Весь алгоритм выделения оптимального базиса задачи тогда записывается одной формулой типа (2.4) или (2.8) в зависимости от используемого итерационного метода. Будем, как и в предыдущем примере, использовать метод простой итерации — решение будет вестись по формуле (2.4).

Запишем систему (2.4) для данного примера:

$$\lambda_1^{(p+1)} = \frac{1}{10} (5 + 0 + 2\lambda_2^{(p)} + 2\lambda_3^{(p)} + 5\lambda_4^{(p)}),$$

$$\lambda_2^{(p+1)} = \frac{1}{10} (-3 + 0,5\lambda_1^{(p)} + 0 + 0 + 1\lambda_4^{(p)}),$$

$$\lambda_3^{(p+1)} = \frac{1}{11} (1 + 5\lambda_1^{(p)} + 4\lambda_2^{(p)} + 0 + 1\lambda_4^{(p)}),$$

$$\lambda_4^{(p+1)} = \frac{1}{14} (-4 + 4\lambda_1^{(p)} + 1\lambda_2^{(p)} + 8\lambda_3^{(p)} + 0),$$

$$\lambda_5^{(p+1)} = (-5 + 0 + 2\lambda_2^{(p)} + 2\lambda_3^{(p)} + 1\lambda_4^{(p)}).$$

Пятое уравнение здесь соответствует пятому столбцу матрицы ограничений, состоящему из неположительных элементов. Это уравнение записано аналогично остальным уравнениям системы (2.4), однако следует учитывать, что если в процессе решения появится значение $\lambda_5^{(p)} > 0$, то это будет означать неограниченность целевой функции данной задачи.

Пусть исходные значения $\lambda_i^{(p)} = 0$, $i = 1, \dots, 5$. Тогда $\lambda_1^{(1)} = 0,5$, $\lambda_2^{(1)} = -0,3$, $\lambda_3^{(1)} = 0,09$, $\lambda_4^{(1)} = -0,28$, $\lambda_5^{(1)} = -5$. Множество индексов положительных переменных $M^{(1)} = \{1, 3\}$. Продолжаем решение по формуле (2.4) при данном M :

$$\lambda_1^{(2)} = \frac{1}{10} (5 + 0 + 0 + 2 \cdot 0,09 + 0) = 0,528,$$

$$\lambda_2^{(2)} = \frac{1}{10} (-3 + 0,5 \cdot 0,5 + 0 + 0 + 0) = -0,275,$$

$$\lambda_3^{(2)} = \frac{1}{11} (1 + 5 \cdot 0,5 + 0 + 0 + 0) = +0,32,$$

$$\lambda_4^{(2)} = \frac{1}{14} (-4 + 4 \cdot 0,5 + 0 + 8 \cdot 0,09 + 0) = -0,09,$$

$$\lambda_5^{(2)} = (-5 + 0 + 0 + 2 \cdot 0,09 + 0) = -4,82.$$

Здесь по-прежнему $M^{(2)} = \{1, 3\}$.

Третий шаг —

$$\lambda_1^{(3)} = \frac{1}{10} (5 + 0 + 0 + 2 \cdot 0,32 + 0) = +0,564,$$

$$\lambda_2^{(3)} = \frac{1}{10} (-3 + 0,5 \cdot 0,52 + 0 + 0 + 0) = -0,274,$$

$$\lambda_3^{(3)} = \frac{1}{11} (1 + 5 \cdot 0,52 + 0 + 0 + 0) = +0,327,$$

$$\lambda_4^{(3)} = \frac{1}{14} (-4 + 4 \cdot 0,52 + 0 + 8 \cdot 0,32 + 0) = +0,046,$$

$$\lambda_5^{(3)} = (-5 + 0 + 0 + 2 \cdot 0,32 + 0) = -4,360.$$

Здесь уже множество M расширилось: $M^{(3)} = \{1, 3, 4\}$.

Четвертый шаг —

$$\lambda_1^{(4)} = \frac{1}{10} (5 + 0 + 0 + 2 \cdot 0,327 + 5 \cdot 0,046) = 0,584,$$

$$\lambda_2^{(4)} = \frac{1}{10} (-3 + 0,5 \cdot 0,564 + 0 + 0 + 1 \cdot 0,046) = -0,267,$$

$$\lambda_3^{(4)} = \frac{1}{11} (1 + 5 \cdot 0,564 + 0 + 0 + 1 \cdot 0,046) = +0,351,$$

$$\lambda_4^{(4)} = \frac{1}{14} (-4 + 4 \cdot 0,564 + 0 + 8 \cdot 0,327 + 0) = +0,062,$$

$$\lambda_5^{(4)} = (-5 + 0 + 0 + 2 \cdot 0,327 + 1 \cdot 0,046) = -4,300.$$

Остальные шаги процесса (2.4) приведены в таблице 3.2. Для наглядности положительные компоненты λ_i в таблице выделены жирным шрифтом.

Ответ. $M^{\text{опт}} = \{1, 3, 4\}$.

Т а б л и ц а 3.2

Качественное решение задачи итеративным методом

p	$\lambda_1^{(p)}$	$\lambda_2^{(p)}$	$\lambda_3^{(p)}$	$\lambda_4^{(p)}$	$\lambda_5^{(p)}$
0	0	0	0	0	0
1	0,5	-0,3	0,09	-0,28	-5
2	0,52	-0,275	0,32	-0,09	-4,82
3	0,564	-0,274	0,327	0,046	-4,360
4	0,584	-0,267	0,351	0,062	-4,300
5	0,601	-0,264	0,362	0,082	-4,236
6	0,613	-0,262	0,372	0,093	-4,194
7	0,621	-0,260	0,378	0,102	-4,163
8	0,627	-0,259	0,382	0,108	-4,142
9	0,630	-0,258	0,386	0,112	-4,128
10	0,633	-0,257	0,387	0,115	-4,116
11	0,635	-0,257	0,389	0,116	-4,111
12	0,636	-0,257	0,390	0,118	-4,104

§ 3. Различные приложения свойств S -задач. Некоторые замечания

1. Решение задачи по частям. Используя пункт (а) свойства 2, мы можем провести решение задачи по частям при произвольном неблочном заполнении матрицы ограничений ненулевыми элементами. Будем, как и в § 1, использовать форму записи симплекс-таблицы из [39], когда записываются только внебазисные векторы. Из свойства 2 пункт (а) следует, что если матрица ограничений задачи линейного программирования сведена к требуемому виду, то столбцы, которые были разрешающими, можно больше не вписывать в таблицу (вектор, однажды выведенный из базиса в данной задаче, больше не будет введен в базис — его можно не рассматривать). Вместо столбцов, которые были разрешающими, можно по блокам вводить в оперативную память машины столбцы A_j , имеющие отрицательные оценки векторов условий ($\Delta_j < 0$), выразив их через векторы базиса. Причем, на каждом этапе введения нового блока векторов в оперативную память машины следует использовать для выражения этих векторов через векторы базиса только те столбцы, которые были разрешающими после предыдущей замены блоков в памяти машины.

В соответствии с требованиями, определяемыми емкостью оперативной памяти машины, разобьем матрицу $A = [a_{ij}]$, $i = 1, \dots, m$, $j = 1, \dots, k_1, \dots, k_2, \dots, k_l = n$ на части (по столбцам)

$$A = [A_s], \quad s = 1, \dots, l,$$

где

$$A_s = [a_{ij}^s], \quad i = 1, \dots, m, \quad j = k_{s-1} + 1, \dots, k_s.$$

На каждом s -м этапе решения задачи (1.1) — (1.3), которую назовем «глобальной» задачей, используем множество векторов, коэффициенты выражения которых через векторы базиса образуют матрицу A_s . Назовем эти этапы решения «глобальной» задачи «локальными» задачами.

Приведенные соображения могут быть сформулированы в следующем виде:

— если матрица ограничений задачи линейного программирования является S -матрицей, то решение «глобальной»

задачи может быть получено через решение «локальных» задач и не более, чем за m итераций симплекс-метода.

При разбиении матрицы A на части следует учитывать величину оценок Δ_j . Векторы, имеющие $\Delta_j < 0$, обязательно будут введены в базис (на основании свойства 3). Поэтому подматрицы A_s с этими векторами нужно располагать в памяти машины таким образом, чтобы ими удобнее было оперировать в первую очередь, т. е. помещать, по возможности, в оперативную память. Не имеет смысла также отыскивать для введения в базис вектор с минимальной оценкой Δ_j — все векторы с отрицательными значениями Δ_j будут все равно введены в базис.

Процедура решения S -задачи по частям напоминает решение системы линейных алгебраических уравнений с разбиением матрицы на блоки, так как известны разрешающие элементы при преобразованиях матрицы (свойство 2 пункт (б)). Решение задачи по частям особенно эффективно в том случае, когда за счет «выбрасывания» из памяти машины векторов, бывших направляющими, удастся разместить в памяти ЦВМ задачи достаточно большой размерности.

Приведем пример. Пусть число ненулевых элементов матрицы ограничений таково, что исходные цифровые данные задачи полностью помещаются в оперативной памяти ЦВМ. Однако в процессе решения происходит увеличение числа ненулевых элементов матрицы, и оперативная память переполняется. Тут и может сыграть решающую роль «выбрасывание» из оперативной памяти векторов, которые были направляющими и (в соответствии со свойством 2а) больше не понадобятся для решения задачи.

2. Некоторые следствия из свойств S -задачи. На практике часто встречаются S -задачи, в которых все коэффициенты целевой функции положительные, и матрица A — квадратная. Из свойства 3 сразу следует: решение подобной задачи находится в точке решения системы $AX = B$. Если же вектор X , получаемый при решении данной системы, будет иметь отрицательные компоненты, то это означает неограниченность целевой функции на допустимом множестве точек S -задачи (см. свойство 7).

Приведем некоторые замечания, непосредственно вытекающие из свойства 4. Данное свойство дает возможность сразу выделить область устойчивости оптимального базиса при изменении вектора правых частей ограничений. Эта

область, очевидно, $B \geq 0$. Заметим, что выделение области устойчивости при изменении вектора B требуется во многих прикладных задачах оптимизации. В настоящее время этот вопрос специально учитывается при составлении некоторых программ решения задач линейного программирования. Кроме того, выделение области устойчивости очень важно в некоторых задачах стохастического программирования, а именно, в задачах со случайным вектором правых частей ограничений. Выделение областей устойчивости позволяет получить решение подобных задач для всего множества реализаций вектора случайной правой части. Имеются работы, специально посвященные устойчивости данных задач [1]. При сведении же задачи к S -задаче, как мы видим, вопрос решается тривиально.

Свойство 4 позволяет также разбить решение S -задачи на два этапа — качественное решение и количественное решение. На качественное решение вектор B не влияет, он используется только для получения количественных значений компонент оптимального плана. Кроме того, свойство 4 оказывается чрезвычайно полезным при решении задачи на моделирующих устройствах, в чем мы убедимся ниже, в главе VI. На основе данного свойства оказывается возможным решать подобные задачи оптимизации непосредственно на моделях многосвязных объектов, не используя дополнительные устройства, моделирующие ограничения.

3. Другое обоснование алгоритмов решения S -задачи. Алгоритмы решения S -задачи, предложенные в предыдущем параграфе, были получены на основе применения к решению S -задачи симплекс-метода с использованием специфических свойств данной задачи. Для лучшего понимания процедуры будет бесполезно рассмотреть получение данных алгоритмов из общих соображений, без опоры на симплекс-метод и свойства S -задач, связанных с применением симплекс-метода.

Пусть рассматривается задача линейного программирования (0.1) — (0.3). Построим алгоритм, позволяющий выделить сразу оптимальный базис данной задачи. Определить оптимальный базис — это значит назвать индексы n (в невырожденном случае) строк ограничений (0.2) — (0.3), обращающихся в оптимальной точке в равенства. Можно заметить, что в опорных планах данной задачи индексы j строк ограничений (0.3), обращающихся в неравенства, со-

впадают с индексами i строк ограничений (0.2), обращающих - ся в равенства. Действительно, с учетом условий (0.3) ограничить увеличение данной компоненты x_j может только условие (0.2) с индексом $i=j$, так как в остальные ограничения (0.2) неотрицательная переменная x_j входит с неположительным коэффициентом, а $b_i \geq 0$. Поэтому если $x_k > 0$, значит, обращается в равенство ограничение (0.2) с индексом $i=k$. Таким образом, для определения оптимального базиса в данной задаче достаточно определить, какие $x_j > 0$ в оптимальной точке.

Алгоритм определения оптимального базиса заключается в следующем. Для переменных, коэффициенты целевой функции которых положительны ($c_j > 0$), в оптимальной точке всегда $x_j > 0$. Для остальных неизвестных вопрос перевода переменной x_k в ненулевое значение решается анализом строки оценок. Пусть, например, используется вполне естественная оценка

$$\delta L = c_k \delta x_k + \sum_{\substack{j=1 \\ j \neq k}}^n c_j \delta x_j, \quad (3.1)$$

где δx_k — увеличение k -й переменной, δx_j — изменение значений остальных ненулевых переменных, вызванное увеличением k -й переменной, δL — изменение значения целевой функции. При этом решение проводится только в сторону увеличения x_j . Процедура выделения оптимального базиса оканчивается, когда не имеется переменных, увеличение которых (перевод в ненулевые) позволяет увеличить целевую функцию. Иными словами, процедура оканчивается, когда для всех $x_j = 0$ оценки (3.1) отрицательные.

Опишем алгоритм подробнее. Пусть индексы переменных x_j с положительными коэффициентами c_j составляют множество M . Если $M = \emptyset$, значит, исходная точка $X = 0$ является оптимальной. Переменные $x_j, j \in M$ обозначим через $\tilde{x}_j$. Эти переменные следует перевести в ненулевые. Увеличивать каждую переменную $\tilde{x}_j$ будем до тех пор, пока не обратится в равенство одно из ограничивающих неравенств. Очевидно, это будет j -е неравенство системы (0.2). Сформируем квадратную матрицу $\tilde{A}$, состоящую из элементов матрицы A , находящихся на пересечении строк и столбцов с индексами $i \in M, j \in M$. Значение L целевой функции после перевода

переменных в ненулевые

$$L = \tilde{C} \tilde{X} = \tilde{C} \tilde{A}^{-1} \tilde{B}, \quad (3.2)$$

где $\tilde{C}$, $\tilde{B}$ — векторы, составленные из элементов, соответственно c_j с индексами $j \in M$ и b_i с индексами $i \in M$. Выражение (3.2) для удобства запишем в виде

$$L = \Lambda \tilde{B},$$

где Λ определяется из решения системы

$$\Lambda \tilde{A} = \tilde{C}. \quad (3.3)$$

Для вычисления оценки (3.1), определяющей целесообразность перевода в ненулевые остальных переменных x_j , потребовалось бы (как следует из изложенного выше) решать соответствующие системы линейных алгебраических уравнений (3.3). Чтобы не делать этого, преобразуем оценку (3.1). Пусть мы находимся в точке, в которой все $\tilde{x}_j > 0$, а остальные переменные x_j ($j \notin M$) равны нулю. Предположим, что для вычисления оценки одна из переменных, имеющих нулевое значение $x_k = 0$, переводится в ненулевое состояние $x_k > 0$. Естественно, при этом изменяются значения ненулевых переменных. Оценка (3.1) примет вид

$$\delta L_k = c_k \delta x_k + \tilde{C} \delta \tilde{X}, \quad (3.4)$$

где $\delta \tilde{X}$ — изменение ненулевых переменных, вызванное увеличением k -й переменной. Учитывая, что при переводе x_k в ненулевое состояние обращается в равенство k -е условие (0.2), имеем

$$\delta x_k = \frac{b_k}{a_{kk}}, \quad (3.5)$$

$$\delta \tilde{X} = -\tilde{A}'_k \delta x_k, \quad (3.6)$$

где b'_k , a'_{kk} , $\tilde{A}'_k$ — элементы соответственно вектора B и матрицы A , выраженные через рассматриваемый базис:

$$\tilde{A}'_k = \tilde{A}^{-1} \tilde{A}_k, \quad (3.7)$$

где $\tilde{A}_k$ — вектор, составленный из элементов k -го столбца матрицы A ($k \notin M$) с индексами $i \in M$. С учетом (3.6), (3.7), оценка (3.4) примет вид

$$\delta L_k = c_k \delta x_k - \tilde{C} \tilde{A}^{-1} \tilde{A}_k \delta x_k.$$

Перейдем к эквивалентной оценке

$$\delta L_k^s = \frac{\delta L_k}{\delta x_k} = c_k - \tilde{C} \tilde{A}^{-1} \tilde{A}_k.$$

Здесь $\delta x_k > 0$, так как в (3.5) $b'_k \geq 0$ из-за структуры матрицы A , а значения $a'_{kk} > 0$, ибо в противном случае целевая функция данной задачи неограничена на рассматриваемом множестве ограничений (при $a'_{kk} > 0$ нет условий, в которые переменная x_k входит с положительным коэффициентом, а значения $b'_k \geq 0$, значит, ничто не ограничивает увеличение переменной x_k). Подставив Λ из (3.3), окончательно получим оценку

$$\delta L_k^s = c_k - \Lambda \tilde{A}_k \quad (3.8)$$

или, в матричном виде, для всех переменных x_j , $j \in M$

$$\delta L^s = \tilde{C} - \Lambda \tilde{A}. \quad (3.9)$$

Вычисление данной оценки равноценно проверке знака неравенств

$$\Lambda \tilde{A} \geq \tilde{C} \quad (\Lambda \tilde{A} < \tilde{C} \text{ при } L^s > 0). \quad (3.10)$$

Обозначения здесь те же, что и в § 1 (см. рис. 3.3). Выражения (3.9), (3.10) аналогичны выражениям (2.1), (2.2). Алгоритм, предложенный в пункте 1 § 2, отсюда очевиден. Но при таком подходе нужно отдельно обосновать, почему решение ведется только в одну сторону: переменные x_j переводятся только из нулевых в ненулевые, но не наоборот. Приведем это обоснование.

Прежде всего убедимся, что переменные, имеющие положительные коэффициенты c_j , переводятся в ненулевые потому, что для этих переменных оценки δL_j положительны. Действительно, учитывая условия неотрицательности переменных, увеличение некоторой j -й компоненты x_j , ограничивается j -м ограничением (0.2) при обращении его в равенство (j -е условие (0.2) имеет положительный элемент в j -м столбце). Если же добавить в левую часть j -го ограничения член $a_{jk}x_k \leq 0$ ($a_{jk} \leq 0$, $x_k > 0$), появляющийся при переводе k -й переменной x_k в число ненулевых, то переменная x_j может только увеличиться. Поэтому перевод k -й переменной в число ненулевых может вызвать только увеличение всех остальных ненулевых переменных, т. е., если $\delta x_k > 0$, то и $\delta x_j \geq 0$

для $j \neq k$. Отсюда — все переменные, имеющие положительные коэффициенты целевой функции, следует перевести в ненулевые, так как для них оценка (3.1) будет положительна. Будем поэтому в дальнейшем говорить о переводе в ненулевые только переменных, имеющих положительное значение оценки δL_j . Положительность коэффициентов c_j — это частный случай положительности δL_j .

Оценка δL_j зависит от того, сколько переменных имеют ненулевые значения, и меняется от итерации к итерации. Для обоснования алгоритма достаточно доказать, что если на данной итерации оценка δL_j положительна, то она останется положительной на всех последующих итерациях. Другими словами, нужно доказать, что оценка δL_j на итерациях не уменьшается.

Для удобства будем рассматривать оценку δL_j° (3.8), эквивалентную оценке δL_j (3.1). Пусть $\delta L_j^{\circ(1)}$ — значение оценки j -й переменной на данной итерации. Далее, пусть переводится в ненулевые некоторая k -я переменная, имеющая положительную оценку $\delta L_k^{\circ(1)}$. Оценкой j -й переменной на следующей итерации будет $\delta L_j^{\circ(2)}$. Требуется доказать, что

$$\delta L_j^{\circ(2)} \geq \delta L_j^{\circ(1)}. \quad (3.11)$$

Запишем выражение (3.8) для δL_j° :

$$\delta L_j^{\circ(1)} = c_j - \Lambda^{(1)} \tilde{A}_j^{(1)}, \quad \delta L_j^{\circ(2)} = c_j - \Lambda^{(2)} \tilde{A}_j^{(2)}.$$

Вектор-столбец $\tilde{A}_j^{(2)}$ формируется путем расширения вектора $\tilde{A}_j^{(1)}$ за счет элемента a_{kj} :

$$\tilde{A}_j^{(2)} = \begin{pmatrix} \tilde{A}_j^{(1)} \\ a_{kj} \end{pmatrix}. \quad (3.12)$$

Все компоненты векторов $\tilde{A}_j^{(1)}$ и $\tilde{A}_j^{(2)}$ неположительны. Векторы-строки $\Lambda^{(2)}$ и $\Lambda^{(1)}$ связаны следующим образом:

$$\Lambda^{(2)} = [(\Lambda^{(1)} + \delta \Lambda^{(1)}), \lambda_k]. \quad (3.13)$$

Учитывая выражения (3.12), (3.13), запишем

$$\delta L_j^{\circ(2)} - \delta L_j^{\circ(1)} = \Lambda^{(1)} \tilde{A}_j^{(1)} - \Lambda^{(2)} \tilde{A}_j^{(2)} = -\delta \Lambda^{(1)} \tilde{A}_j^{(1)} - \lambda_k a_{kj}.$$

Для доказательства условий (3.11) достаточно убедиться, что

$$\lambda_k \geq 0, \quad (3.14)$$

$$\delta \Lambda^{(1)} \geq 0. \quad (3.15)$$

Покажем, что условия (3.14), (3.15) выполняются, если целевая функция задачи ограничена на рассматриваемом множестве. Вектор $\Lambda^{(1)}$ получается из решения системы $\Lambda^{(1)} \tilde{A}_{(1)} = \tilde{C}_{(1)}$. Вектор $\Lambda^{(2)}$ получается из решения системы $\Lambda^{(2)} \tilde{A}_{(2)} = \tilde{C}_{(2)}$, где $\tilde{A}_{(2)}$ — матрица, получаемая из матрицы $\tilde{A}_{(1)}$ путем окаймления ее соответствующими элементами k -й строки и столбца; $\tilde{C}_{(2)}$ — вектор-строка, получаемая путем расширения вектора $\tilde{C}_{(1)}$ за счет элемента c_k . Если матрица $\tilde{A}_{(1)}$ имеет размерность, равную $l \times l$, то матрица $\tilde{A}_{(2)}$ имеет размерность $(l+1) \times (l+1)$. После перенесения элементов подматрицы $\tilde{A}_{(1)}$ в левый верхний угол матрицы A (как показано на рис. 3.3) и соответствующей перенумерации элементов, запишем

$$\left. \begin{aligned} \tilde{A}_{(2)} &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1l} & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2l} & a_{2k} \\ \dots & \dots & \dots & \dots & \dots \\ a_{l1} & a_{l2} & \dots & a_{ll} & a_{lk} \\ a_{k1} & a_{k2} & \dots & a_{kl} & a_{kk} \end{bmatrix} = \begin{bmatrix} \tilde{A}_{(1)} & u_k \\ v_k & a_{kk} \end{bmatrix}, \\ \tilde{C}_{(2)} &= [c_1 \ c_2 \ \dots \ c_l \ c_k] = [\tilde{C}_{(1)} \ c_k]. \end{aligned} \right\} \quad (3.16)$$

Пусть мы находимся в точке решения системы $\Lambda^{(1)} \tilde{A}_{(1)} = \tilde{C}_{(1)}$. Этой точке соответствует матрица

$$\left. \begin{aligned} A'_{(2)} &= \begin{bmatrix} \tilde{A}_{(1)}^{-1} & -\tilde{A}_{(1)}^{-1} u_k \\ v_k \tilde{A}_{(1)}^{-1} & a_{kk} - v_k \tilde{A}_{(1)}^{-1} u_k \end{bmatrix} = \begin{bmatrix} \tilde{A}'_{(1)} & u'_k \\ v'_k & a'_{kk} \end{bmatrix}, \\ \tilde{C}'_{(2)} &= [\tilde{C}_{(1)} \tilde{A}_{(1)}^{-1} \ c_k - \tilde{C}_{(1)} \tilde{A}_{(1)}^{-1} u_k] = [\tilde{C}'_{(1)} \ c'_k]. \end{aligned} \right\} \quad (3.17)$$

Вектор $\tilde{C}'_{(1)}$ здесь представляет собой решение $\Lambda^{(1)}$ системы $\Lambda^{(1)} \tilde{A}_{(1)} = \tilde{C}_{(1)}$. Пусть, далее, мы хотим перейти от решения $\Lambda^{(1)}$ «неокаймленной» системы к решению $\Lambda^{(2)}$ «окаймленной» системы. Тогда из (3.17) получим значения λ_k и $\delta \Lambda^{(1)}$ в

формуле (3.13):

$$\lambda_k = \frac{c'_k}{a_{kk}}, \quad (3.18)$$

$$\delta\Lambda^{(1)} = -v'_k \lambda_k = -v_k \tilde{A}_{(1)}^{-1} \lambda_k. \quad (3.19)$$

Для доказательства условий (3.14), (3.15) рассмотрим полученные выражения. В формуле (3.18) значение c'_k положительно, так как c'_k равняется $\delta L_k^{(1)}$ (по условию $\delta L_k^{(1)} > 0$). Это нетрудно заметить, если сравнить выражения c'_k из (3.17) и $\delta L_k^{(1)}$ из (3.8):

$$c'_k = c_k - \Lambda^{(1)} u_k, \quad \delta L_k^{(1)} = c_k - \Lambda^{(1)} \tilde{A}_k^{(1)},$$

где $u_k = \tilde{A}_k^{(1)}$. Вектор v_k в формуле (3.19) неположительный, так как в него не входят диагональные элементы S -матрицы. Таким образом, условия (3.14), (3.15) выполняются в тех случаях, когда $a'_{kk} \geq 0$, $\tilde{A}_{(1)}^{-1} \geq 0$. Но аналогично тому, как это сделано при доказательстве свойства 7 в § 1 данной главы, нетрудно показать, что $a'_{kk} \geq 0$, $\tilde{A}^{-1} \geq 0$, если целевая функция задачи ограничена на рассматриваемом множестве.

Итак, в случае ограниченности целевой функции задачи на рассматриваемом множестве выполняются условия (3.14), (3.15). Оценки δL_j^g по итерациям увеличиваются, и задачи можно решать в одну сторону — переводить x_j из нулевых в ненулевые, но не наоборот. Мы провели обоснование алгоритма для того случая, когда целевая функция ограничена на рассматриваемом множестве. Заметим, что на основе аналогичных выкладок нетрудно также провести обоснование способов выявления неограниченности целевой функции.

§ 4. Приближенное решение

При решении задачи линейного программирования большой размерности очень важно иметь возможность достаточно быстро получать оценочное, приближенное решение задачи. Специфика S -задач позволяет легко выделить базис, достаточно близкий к оптимальному. В некоторых задачах, например, при оптимизации нефтедобычи, этот «субоптимальный» базис обычно совпадает с оптимальным. В данном

параграфе будет описан процесс получения приближенного качественного решения задачи. Мы будем говорить только о качественном решении, так как, имея качественное решение, всегда можно получить и количественное, решив систему (2.9). Кроме того, во многих случаях нас интересует именно качественное решение, т. е. индексы векторов, входящих в оптимальный базис (или, как в данном случае, — в субоптимальный базис). Например, в задаче оптимизации нефтедобычи качественное решение задачи указывает, какие скважины необходимо отключить (или перевести на нижний предел по дебиту), а какие скважины оставить включенными. Численные же значения дебитов устанавливаются автоматически после перевода пласта на режим, соответствующий решению S -задачи линейного программирования.

Как было показано выше, в оптимальный базис S -задачи входят все векторы $\tilde{A}_j$, имеющие $c_j > 0$ и, кроме того, векторы, оценка Δ_j которых становится отрицательной при введении всех возможных векторов $\tilde{A}_j$ в базис. Заметим (из примеров, приведенных в главе I), что элементы псевдодиагонали рассматриваемых S -матриц, как правило, значительно превосходят остальные элементы матрицы. В то же время оценка Δ_k при введении в базис вектора $\tilde{A}_l$ изменяется:

$$\Delta'_k = \Delta_k + a_{lk}\lambda_l, \quad (4.1)$$

где $a_{lk} \leq 0$,

$$\lambda_l = \frac{-\Delta_l}{a_{ll}}. \quad (4.2)$$

В матричном виде данным выражениям соответствует запись (2.1), (2.3). Если в (4.1), (4.2) $a_{ll} \gg a_{lk}$, то Δ'_k незначительно меньше Δ_k . Поэтому базис, получаемый после введения в него векторов $\tilde{A}_j$ с положительными c_j , можно принимать за оценочное приближенное решение.

Таким образом, оценочное качественное решение задачи может быть дано без каких-либо вычислений и записывается следующим образом: $x_j > 0$ — для столбцов, имеющих $c_j > 0$, $x_j = 0$ — для остальных столбцов. Более близкое к оптимальному качественное решение получается, если приближенно определить индексы оценок Δ_j , которые становятся отрицательными при введении в базис векторов $\tilde{A}_j$, имеющих $c_j > 0$. Это решение будет следующим: $x_j > 0$ — для столбцов, имеющих $\Delta'_j < 0$, $x_j = 0$ —

для столбцов, имеющих $\Delta'_j \geq 0$, где

$$\Delta'_j = \begin{cases} -c_j, & \text{если } c_j > 0, \\ -c_j + \Lambda^n \tilde{A}_j, & \text{если } c_j \leq 0, \end{cases}$$

а $\lambda_j^n = \tilde{c}_j/a_{jj}$ — компоненты приближенного решения $\Lambda^n = [\lambda_i^n]$ системы (2.3). Область, в которой находится решение, близкое к оптимальному, может быть выделена, если добавить к полученной вершине другие, соответствующие введению в базис векторов с оценками Δ_j , близкими к нулю, и имеющих ненулевые коэффициенты a_{ij} .

Получение приближенного качественного решения задачи будет проиллюстрировано в § 6 данной главы на примере задачи оптимизации нефтедобычи. В данной задаче коэффициенты a_{jj} «самовлияния» скважины значительно превосходят коэффициенты a_{ij} «взаимовлияния» скважин ($i \neq j$). Приращения оценок Δ_k внебазисных векторов (см. (4.1)) обычно соизмеримы с погрешностью определения коэффициентов взаимовлияния. Поэтому приближенное качественное решение задачи обычно совпадает с точным качественным решением.

§ 5. Оптимальное управление режимом нефтяного пласта. Две постановки задачи

1. Две модели. Установившийся режим работы нефтяного пласта описывается уравнением типа Лапласа или типа Пуассона. После конечно-разностной аппроксимации это уравнение заменяется системой линейных алгебраических уравнений с S -матрицей (см. § 3 главы I). В качестве переменных в систему входят депрессии во всех узлах аппроксимирующей сетки. При решении задач управления пластом нас интересуют величины депрессий не во всех узлах сетки, а только в тех, в которых расположены скважины. Поэтому в ряде случаев имеет смысл перейти к другой системе линейных алгебраических уравнений (см. выражение (3.20) главы I), в которую в качестве переменных входят только депрессии в точках расположения скважин. При этом давления во всех остальных узлах сетки выражаются через давления в узлах-скважинах. Полученная система связывает депрессии и дебиты скважин. Для удобства

изложения приведем здесь эту систему:

$$\mathbf{A}\delta P = Q, \quad (5.1)$$

где $\mathbf{A}=[a_{ij}]$ — матрица коэффициентов взаимовлияния скважин по дебиту; $\delta P=[\delta p_j]$ — вектор депрессий в узлах расположения скважин; $Q=[q_i]$ — вектор дебитов скважин. Зависимость (5.1), очевидно, может быть представлена также в следующем виде:

$$\mathbf{A}^0 Q = \delta P, \quad (5.2)$$

где $\mathbf{A}^0=\mathbf{A}^{-1}$, $\mathbf{A}^0$ — матрица коэффициентов взаимовлияния скважин по давлению (или депрессии). Матрица $\mathbf{A}$ является S -матрицей, матрица $\mathbf{A}^0$ — неотрицательная. На практике могут быть получены как матрица $\mathbf{A}$, так и матрица $\mathbf{A}^0$. Если имеется матрица $\mathbf{A}$ коэффициентов взаимовлияния скважин по дебиту, то задача оптимизации нефтедобычи при линейной целевой функции и технологических ограничениях сводится к S -задаче линейного программирования (см. задачу (3.26) в первой главе). Если же имеется матрица $\mathbf{A}^0$ коэффициентов взаимовлияния по депрессии, то данная задача оптимизации нефтедобычи записывается в виде

$$C^0 Q \rightarrow \max \quad (5.3)$$

при

$$\mathbf{A}^0 Q \leq \delta P^{\text{пр}}, \quad (5.4)$$

$$Q \geq 0, \quad (5.5)$$

где C^0 — вектор коэффициентов целевой функции, например, коэффициентов нефтесодержания скважин, $\delta P^{\text{пр}}$ — предельные значения депрессий. Неравенства (5.4) получаются здесь после подстановки в ограничения $\delta P \leq \delta P^{\text{пр}}$ выражения δP из (5.2).

Задача (5.3) — (5.5) также может быть сведена к S -задаче. Но для этого нужно перейти от матрицы $\mathbf{A}^0$ к матрице $\mathbf{A}$ (получить матрицу $\mathbf{A}$ на модели нефтяного пласта или обратить матрицу $\mathbf{A}^0$). Однако рассматриваемые задачи оптимизации нефтедобычи можно решать с помощью предложенных в данной главе алгоритмов не только в том случае, когда задача формулируется на основе матрицы $\mathbf{A}$ и является S -матрицей, но и в случае использования матрицы $\mathbf{A}^0$, без специального перехода от задачи (5.3) — (5.5) к S -задаче.

В последующих двух параграфах будет рассмотрено решение данной задачи для обеих постановок: при использовании S -матрицы $\mathbf{A}$ коэффициентов взаимовлияния скважин по дебиту, и при использовании неотрицательной матрицы

A^0 коэффициентов взаимовлияния скважин по депрессии. Заметим, что как при решении S -задачи с матрицей A , так и при решении эквивалентной задачи с матрицей A^0 учитываются только основные ограничения задачи оптимизации нефтедобычи, определяемые технологией разработки пласта. Это — ограничения, накладываемые на значения депрессий, и условия, определяющие нижние границы дебитов скважин. Если в задаче, кроме данных, основных ограничений, имеются еще и некоторые другие, то, как уже указывалось в первой главе, задача может быть либо также сведена к S -задаче при дополнительных предположениях либо эти ограничения выделяются из S -задачи с помощью декомпозиции. В последнем случае S -задачу следует рассматривать как вспомогательную задачу в процедуре декомпозиции, а сам алгоритм декомпозиции — как метод, позволяющий расширить область применения специальных алгоритмов решения S -задач линейного программирования. В дальнейшем будут приведены некоторые случаи эффективного использования процедуры декомпозиции при решении S -задачи с дополнительным ограничением.

2. Зависимость матриц A и A^0 от числа рассматриваемых скважин. Для последующего изложения нам потребуется доказать следующее свойство матриц A и A^0 :

коэффициенты матрицы A^0 взаимовлияния скважин по депрессии не зависят от числа рассматриваемых скважин. Коэффициенты матрицы A взаимовлияния скважин по дебиту зависят от числа рассматриваемых скважин. При изменении числа рассматриваемых скважин в матрице A^0 вычеркивается или добавляется соответствующее число строк и столбцов, а в матрице A , кроме этого, происходит преобразование остальных коэффициентов по рекуррентным формулам жордановых исключений.

Поясним это свойство на примере. Пусть рассматривается нефтеносный участок, состоящий из четырех скважин, и матрицы коэффициентов взаимовлияния A^0 , A для этих четырех скважин следующие:

$$A^0 = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 1,02 & 0,12 & 0,14 & 0,08 \\ 0,12 & 2,31 & 0,22 & 0,11 \\ 0,14 & 0,22 & 4,43 & 0,13 \\ 0,08 & 0,11 & 0,13 & 1,55 \end{bmatrix} \end{matrix},$$

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} +0,9933 & -0,0468 & -0,0277 & -0,0456 \\ -0,0468 & +0,4385 & -0,0195 & -0,0271 \\ -0,0277 & -0,0195 & +0,2281 & -0,0163 \\ -0,0456 & -0,0271 & -0,0163 & +0,6508 \end{bmatrix} \end{matrix}.$$

Если одна из скважин, например, четвертая, вышла из строя либо оказалась отключенной при переводе пласта на оптимальный режим, то матрица A' коэффициентов взаимовлияния по депрессии для оставшихся трех скважин будет получена вычеркиванием четвертой строки и столбца:

$$A' = \begin{bmatrix} 1,02 & 0,12 & 0,14 \\ 0,12 & 2,31 & 0,22 \\ 0,14 & 0,22 & 4,43 \end{bmatrix}.$$

Матрица A' коэффициентов взаимовлияния по дебиту для оставшихся скважин будет получена после преобразования матрицы A по рекуррентным формулам жордановых исключений с разрешающим элементом a_{44} :

$$A' = \begin{bmatrix} +0,9901 & -0,0487 & -0,0289 \\ -0,0487 & +0,4374 & -0,0202 \\ -0,0289 & -0,0202 & +0,2276 \end{bmatrix}.$$

Здесь каждый коэффициент a'_{ij} вычисляется следующим образом:

$$a'_{ij} = a_{ij} - \frac{a_{ir}a_{rj}}{a_{rr}}, \quad i \neq r, \quad j \neq r,$$

где a_{rr} — разрешающий элемент, a_{ir} , a_{rj} — элементы разрешающей строки и столбца. Если исключаются из рассмотрения не одна, а несколько скважин, то аналогичные формулы записываются в виде

$$A'_{ij} = A_{ij} - A_{ir}A_{rr}^{-1}A_{rj},$$

где A_{rr} — подматрица, состоящая из элементов матрицы A , находящихся на пересечении разрешающих строк и столбцов (разрешающими в данном случае являются те строки и столбцы, которые соответствуют исключаемым из рассмотрения скважинам); A_{ir} — подматрица, состоящая из элементов, находящихся на пересечении разрешающих столбцов

и неразрешающих строк; A_{rj} — подматрица, состоящая из элементов, находящихся на пересечении разрешающих строк и неразрешающих столбцов; A_{ij} — подматрица, состоящая из элементов, находящихся на пересечении неразрешающих строк и столбцов матрицы A ($A_{ij} = A'$). Например, если в рассматриваемой задаче исключить третью и четвертую скважины, то матрица A' коэффициентов взаимовлияния по дебиту для оставшихся двух скважин примет вид

$$A' = \begin{bmatrix} +0,9864 & -0,0512 \\ -0,0512 & +0,4356 \end{bmatrix}.$$

Матрица A' получена по указанной формуле, где

$$\begin{aligned} A_{rr} &= \begin{bmatrix} +0,2281 & -0,0163 \\ -0,0163 & +0,6508 \end{bmatrix}, & A_{ir} &= \begin{bmatrix} -0,0277 & -0,0456 \\ -0,0195 & -0,0271 \end{bmatrix}, \\ A_{rj} &= \begin{bmatrix} -0,0277 & -0,0195 \\ -0,0456 & -0,0271 \end{bmatrix}, & A_{ij} &= \begin{bmatrix} +0,9933 & -0,0468 \\ -0,0468 & +0,4385 \end{bmatrix}. \end{aligned}$$

Матрица $A^{0'}$ коэффициентов взаимовлияния по депрессии для оставшихся двух скважин получается вычеркиванием третьей и четвертой строки и столбца матрицы A^0 :

$$A^{0'} = \begin{bmatrix} 1,02 & 0,12 \\ 0,12 & 2,31 \end{bmatrix}.$$

Приступим теперь к доказательству указанного свойства. Прежде всего напомним, откуда получают матрицы A и A^0 . Пусть после аппроксимации пласта сеткой, состоящей из N узлов, получена следующая система линейных алгебраических уравнений (см. выражения (3.16), (3.19) в главе I):

$$G \cdot \delta P = F, \quad (5.6)$$

где G — квадратная матрица размера $N \times N$; δP — N -мерный вектор депрессий в узлах сетки, F — N -мерный вектор значений функции источников или стоков в узлах сетки. Пусть n строк и столбцов системы (5.6), соответствующих узлам расположения скважин, перенесены в верхний левый угол матрицы G . Тогда запишем (аналогично (3.19) главы I)

$$G = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad (5.7)$$

где A_{11} — квадратная матрица размера $n \times n$, строки и столбцы которой соответствуют узлам расположения скважин.

Матрица A коэффициентов взаимовлияния скважин по дебиту получается на месте матрицы A_{11} после выражения депрессий во всех узлах сетки через депрессии в узлах расположения скважин:

$$A = A_{11} - A_{12}A_{22}^{-1}A_{21}. \quad (5.8)$$

Изменение числа n рассматриваемых скважин сказывается на разбиении (5.7) матрицы G на блоки. Меняются размеры блоков, а это в свою очередь изменяет не только размер матрицы A , но и значения коэффициентов (см. формулу (5.8)), причем коэффициенты матрицы A изменяются в этом случае в соответствии с рекуррентными формулами жордановых исключений.

Действительно, пусть получена матрица A размера $n \times n$. Чтобы получить матрицу A' коэффициентов взаимовлияния при исключении l скважин с индексами $i=j=r$, нужно выразить депрессии в r -х узлах сетки через депрессии в $n-l$ оставшихся узлах расположения скважин. Это осуществляется аналогично (5.8) следующим образом:

$$A' = A_{n-l, n-l} - A_{n-l, l}A_{ll}^{-1}A_{l, n-l}, \quad (5.9)$$

где индексация матриц указывает их размеры. Условие (5.9) и есть искомое преобразование.

Матрица A^0 получается так:

$$A^0 = A^{-1} = (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}$$

и является частью матрицы G^{-1} . Отсюда ясно, почему коэффициенты матрицы A^0 не зависят от числа n рассматриваемых скважин — коэффициенты обратной матрицы не зависят от того, каким образом разбита исходная матрица G на блоки в (5.7). При изменении числа рассматриваемых скважин меняются только размеры подматрицы $(A'_{11})^{-1} = A^0$, где $A'_{11} = A$ вычисляется по формуле (5.8). Что и требовалось доказать.

Заметим, что коэффициенты взаимовлияния скважин зависят от многих показателей — от свойств пласта и насыщающей жидкости, от расстояний между скважинами, расстояния до контура, и так далее. Однако мы остановились только на определении зависимости коэффициентов взаимовлияния от числа рассматриваемых скважин, так как именно этот показатель изменяется в процессе оптимизации.

Остальные величины, от которых зависят коэффициенты взаимовлияния, в данной статической постановке задачи оптимизации нефтедобычи остаются без изменения. И в случае изменения этих показателей разработки коэффициенты взаимовлияния пересчитываются вновь.

3. Определение коэффициентов взаимовлияния. На практике, матрицы A и A^0 находятся способом более простым, чем тот, который мы использовали выше при выводе уравнений. Если подать скачок $\delta'p_k$ депрессии по одной, k -й скважине, сохранив по остальным скважинам депрессии на прежнем уровне (режим заданных депрессий), то k -й столбец матрицы A из (5.1) примет вид

$$A_k = \begin{bmatrix} \delta q_1 / \delta' p_k \\ \delta q_2 / \delta' p_k \\ \vdots \\ \delta q_n / \delta' p_k \end{bmatrix},$$

где δq_i , $i=1, \dots, n$ — изменение дебита скважин, вызванное скачком $\delta'p_k$ депрессии k -й скважины. Если же подать по одной k -й скважине скачок дебита δq_k , сохранив дебиты остальных скважин на прежнем уровне (режим заданных дебитов), то k -й столбец матрицы A^0 из (5.2)

$$A_k^0 = \begin{bmatrix} \delta' p_1 / \delta q_k \\ \delta' p_2 / \delta q_k \\ \vdots \\ \delta' p_n / \delta q_k \end{bmatrix},$$

где $\delta'p_i$, $i=1, \dots, n$ — изменение депрессий скважин, вызванное скачком дебита на k -й скважине. Если скачок считать единичным, то в первом случае получим $a_{ik} = \delta q_i$, во втором случае $a_{ik}^0 = \delta p_i$. Указанный способ определения взаимных сопротивлений и проводимостей между узлами хорошо известен в электротехнике.

Для рассматриваемой нефтяной задачи процедура будет заключаться в следующем. Для получения матрицы A измеряются приращения дебитов скважин при подаче единичных возмущений депрессии поочередно на все n скважин. Для получения матрицы A^0 подаются скачки дебитов и измеряются приращения депрессий. В принципе, такая процедура может быть использована для определения матриц A

и A^0 непосредственно на нефтяном пласте. Но практически осуществить это на реальном месторождении не представляется возможным. Поэтому коэффициенты взаимовлияния скважин получают с помощью электрической модели нефтяного пласта. В этом случае дебиты интерпретируются как токи в соответствующих узлах, а депрессии — как напряжения.

Заметим, что матрицы A и A^0 можно получать также на ЦВМ с помощью решения n систем линейных алгебраических уравнений (5.6), возникающих при записи уравнений типа Пуассона в конечных разностях. Для получения каждого k -го столбца искомой матрицы решается отдельная система (5.6) при заданных значениях дебитов или депрессий в соответствующих узлах аппроксимирующей сетки. Дебиты или депрессии задаются в зависимости от режима (для матрицы A — режим заданных депрессий, для матрицы A^0 — режим заданных дебитов).

4. Замечание о решении задачи оптимизации без определения коэффициентов взаимовлияния. Важно отметить, что в некоторых случаях проще ставить задачу оптимизации, используя исходную матрицу G из (5.6), не переходя к матрице A (об этом говорилось в § 3 главы I). При использовании матрицы G этап нахождения коэффициентов взаимовлияния, естественно, отпадает. Задача оптимизации в этом случае также сводится к S -задаче. Предполагается только, что в $(N-n)$ узлах, в которых не имеется источников, скважины заранее считаются отключенными (в алгоритмах § 2 данной главы исходное множество индексов M состоит из индексов этих $(N-n)$ узлов). Итерационные алгоритмы § 2 в этом случае можно строить путем обобщения на задачи оптимизации любого итерационного метода решения систем линейных алгебраических уравнений, получаемых после дискретизации уравнения Пуассона. Обобщение будет заключаться в том (см. § 2), что вместо суммирования по всем j (что имеет место при решении систем), для решения задачи оптимизации суммирование будет производиться только положительных переменных ($j \in M$).

При таком подходе задача оптимизации будет решаться непосредственно после конечно-разностной аппроксимации пласта, без нахождения коэффициентов взаимовлияния, причем трудоемкость решения задачи оптимизации будет не

больше, чем трудоемкость расчета одного режима работы пласта. Практически трудоемкость будет даже меньше, так как суммирование производится только для $j \in M$.

§ 6. Решение задачи оптимизации нефтедобычи по матрице коэффициентов взаимовлияния скважин по дебиту

Пусть имеется матрица A коэффициентов взаимовлияния скважин по дебиту и задача сформулирована в виде S -задачи (см. § 3 главы I). Рассмотрим пример решения данной S -задачи с помощью алгоритмов, предложенных выше. Возьмем задачу оптимизации режима нефтяного месторождения с семью скважинами, цифровые данные которой приведены в таблице 3.3.

Поясним обозначения таблицы 3.3: C^0 — вектор коэффициентов нефтесодержания скважин (т — знак транспонирования), $\bar{C} = -C^0 A$ — вектор коэффициентов целевой функции после сведения исходной задачи к S -задаче, $B = A \delta P^{np}$ — вектор правой части ограничений S -задачи (физически — вектор дебитов скважин при предельных депрессиях). Заметим, что векторы C , B могут быть получены не только расчетным путем, с помощью умножения векторов C^0 и δP на матрицу A , но и сняты с модели нефтяного пласта. Для вектора B , очевидно, требуется замерить дебиты при предельных значениях депрессий на скважинах. Для получения вектора C достаточно замерить дебиты при задании депрессий на скважинах, пропорциональных коэффициентам нефтесодержания.

Итак, пусть имеется S -задача

$$CX \rightarrow \max, \quad (6.1)$$

$$AX \leq B, \quad (6.2)$$

$$X \geq 0, \quad (6.3)$$

где в качестве вектора переменных X принят вектор разности предельных и реальных значений депрессий (см. § 2 главы I). Задачу будем решать в два этапа: сначала найдем качественное решение, а потом — количественное. Напомним, что под качественным решением задачи понимается выделение оптимального базиса. Физически качественное решение в данном случае означает определение номеров скважин, которые следует отключить (или перевести на

Таблица 3.3

Данные задачи оптимизации месторождения с семью скважинами

С _{от}	А							В						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1,0	+2,3400	-0,0302	-0,0224	-0,0824	-0,0403	-0,0131	-0,0231	35,0						
0,25	-0,0302	+2,1600	-0,0070	-0,0407	-0,3350	-0,0070	-0,0403	15,0						
0,05	-0,0224	-0,0070	+1,1960	-0,0345	-0,0100	-0,0376	-0,0161	95,0						
0,9	-0,0824	-0,0407	-0,0345	+1,2340	-0,0903	-0,0376	-0,0087	90,0						
0,1	-0,0403	-0,3350	-0,0100	-0,0903	+3,5000	-0,0101	-0,0103	25,0						
0,15	-0,0131	-0,0070	-0,0376	-0,0376	-0,0101	+1,0680	-0,1430	90,0						
1,0	-0,0231	-0,0403	-0,0161	-0,0087	-0,0103	-0,1430	+1,6480	70,0						
С	-2,2281	-0 3980	+0,0181	-0,9929	-0,1324	+0,0344	-1,5837]							

нижний предел по дебиту) в оптимальной точке. Количественное решение дает значение дебитов скважин в оптимальной точке *).

Этап I. Качественное решение.

1. *Оценочное решение.* Следуя рекомендациям § 4 данной главы, оценочное качественное решение рассматриваемого примера может быть записано без проведения каких-либо вычислений:

$$\begin{aligned} x_3 > 0, \quad x_6 > 0, \quad \text{так как } c_3 > 0, c_6 > 0, \\ x_j = 0 \quad \text{для } j = 1, 2, 4, 5, 7. \end{aligned}$$

Физическая интерпретация. Полученное решение означает, что третья и шестая скважины при оптимизации должны быть отключены, а остальные скважины должны работать при предельных значениях депрессий. Докажем это.

Вектор Q дебитов скважин фигурирует в задаче (6.1) — (6.3) в виде вектора дополнительных переменных, обращающих условия (6.2) в равенства, т. е.

$$AX + EQ = B, \quad (6.4)$$

где E — единичная матрица. Напомним, как получается условие (6.4). Пусть имеются технологические ограничения $\delta P \leq \delta P^{\text{нр}}$, или, в виде равенства,

$$\delta P + X = \delta P^{\text{нр}}.$$

Подставив вместо значений депрессий их выражение через дебиты (5.2), получим

$$A^0 Q + X = \delta P^{\text{нр}}.$$

Умножим это выражение на матрицу $A = (A^0)^{-1}$ слева. Тогда

$$(AA^0) Q + AX = A\delta P^{\text{нр}}$$

или

$$EQ + AX = B, \quad (6.5)$$

где $B = A\delta P^{\text{нр}}$. Выражения (6.5) и (6.4) совпадают. Таким образом, исходным базисом задачи (6.1) — (6.3) является $Q = B, X = 0$, т. е. в исходной точке все скважины включены при предельных значениях депрессий. В процессе решения

*) Для удобства записи обозначения размерности приводятся только на конечном этапе решения задачи.

задачи будет происходить отключение скважин, или, если отключение недопустимо, то перевод на нижний предел по дебиту. При этом всегда будет выполняться следующее условие: если $x_k=0$, то $q_k>0$, если же $x_k>0$, то $q_k=0$ (см. свойство 1 из § 1 данной главы). Иными словами, если $x_k=0$, то скважина включена при предельных значениях депрессий по работающим скважинам, а если $x_k>0$, то скважина отключена, что и требовалось доказать.

Рассмотрим теперь, как физически интерпретировать тот факт, что данное решение задачи заключается в отключении скважин, которым соответствуют положительные коэффициенты c_j целевой функции (6.1). Напомним прежде всего смысл рассматриваемой задачи оптимизации нефтедобычи. Нефтяной пласт представляет единую гидродинамически связанную систему. Уменьшение дебита одних скважин вызывает увеличение дебита остальных. Поэтому может оказаться, что уменьшение дебита сильно обводненных скважин даст положительное приращение суммарной добычи чистой нефти по пласту в целом за счет получаемого при этом увеличения дебита малообводненных скважин. Увеличение суммарной добычи чистой нефти по пласту в этом случае будет сопровождаться уменьшением добычи жидкости. Задача и заключается в определении номеров скважин, уменьшение дебита которых дает положительное приращение δL_j целевой функции. Если $\delta L_j>0$, то уменьшать дебит j -й скважины нужно до нижнего предела по дебиту, или, в частности, до нуля (отключить скважину).

Вектор C в линейной форме (6.1) получается следующим образом: $C = -C^0 A$. Умножение вектора C^0 на матрицу A и представляет собой вычисление приращений δL_j суммарной добычи чистой нефти по пласту, вызванных уменьшением дебита j -й скважины ($j=1, 2, \dots, n$), т. е.

$$C = -C^0 A = \delta L.$$

Действительно, элемент a_{ij} матрицы A коэффициентов взаимовлияния скважин по дебиту показывает, насколько уменьшится (если $a_{ij}>0$), или увеличится (если $a_{ij}<0$) дебит i -й скважины при уменьшении на единицу интенсивности дебита j -й скважины. Умножение a_{ij} на коэффициент нефтесодержания c_i^0 дает уменьшение (или увеличение) дебита i -й скважины уже в отношении чистой нефти, а не жидкости. Значение $c_i^0 a_{ij}$ при $i=j$ показывает уменьшение (так как $a_{jj}>0$) добычи

чистой нефти на j -й скважине при уменьшении ее дебита. Значения $c_i^0 a_{ij}$ при $i \neq j$ показывают увеличение (так как $a_{ij} < 0$ при $i \neq j$) добычи чистой нефти по остальным скважинам при уменьшении дебита j -й скважины. Если потери чистой нефти на j -й скважине при уменьшении ее дебита оказываются меньше, чем суммарное приращение добычи чистой нефти по остальным скважинам, то дебит данной скважины имеет смысл уменьшить. В противном случае дебит скважины должен быть равен предельно возможному при имеющихся ограничениях на депрессию. Иными словами, если

$$\sum_{\substack{i=1 \\ i \neq j}}^n |c_i^0 a_{ij}| - c_j^0 a_{jj} > 0,$$

значит, имеет смысл уменьшить дебит j -й скважины до предела (или отключить); если

$$\sum_{\substack{i=1 \\ i \neq j}}^n |c_i^0 a_{ij}| - c_j^0 a_{jj} < 0,$$

значит, нужно включить j -ю скважину на режим предельного значения депрессии.

Таким образом, мы убедились, что

$$c_j = - \sum_{i=1}^{n+1} c_i^0 a_{ij} = \delta L_j^2 \quad (6.6)$$

определяет приращение суммарной добычи чистой нефти по всему пласту при уменьшении на единицу интенсивности дебита одной j -й скважины. Причем, если $\delta L_j > 0$, то при оптимизации следует уменьшить дебит j -й скважины до нижнего предела. Что и требовалось доказать.

Например, для рассматриваемой задачи с семью скважинами, если уменьшить дебит третьей скважины на единицу, то потеря нефти по 3-й скважине составит $c_3^0 a_{33} = 0,05 \cdot 1,196 = 0,0598$. Увеличение добычи чистой нефти по остальным скважинам при этом

$$\begin{aligned} \sum_{\substack{i=1 \\ i \neq j}}^n |c_i^0 a_{i3}| &= 1,0 \cdot 0,0224 + 0,25 \cdot 0,007 + 0,9 \cdot 0,0395 + \\ &+ 0,1 \cdot 0,01 + 0,15 \cdot 0,0376 + 1,0 \cdot 0,0161 = 0,0779. \end{aligned}$$

Прирост добычи чистой нефти больше, чем ее потеря:

$$c_3 = 0,0799 - 0,0598 = 0,0181 > 0,$$

поэтому третью скважину следует отключить.

2. *Приближенное решение.* Практически при решении реальных задач оптимизации нефтедобычи оценочное качественное решение обычно совпадает с точным качественным решением. Поэтому качественное решение задачи на этом можно было бы закончить. Однако для иллюстрации алгоритмов проведем решение полностью.

Приближенное качественное решение, более близкое к оптимальному, чем оценочное, получается в соответствии с рекомендациями § 4 следующим образом:

$$\lambda_3^n = \frac{0,0181}{1,196} = 0,0151; \quad \lambda_6^n = \frac{0,0344}{1,068} = 0,0322.$$

Приращения $\lambda^n \tilde{A}_j$ оценок Δ_j для $j \notin M$

i	$\lambda^n \tilde{A}_j$
1	$0,0224 \cdot 0,0151 + 0,0131 \cdot 0,0322 = 0,0007$
2	$0,0070 \cdot 0,0151 + 0,0070 \cdot 0,0322 = 0,0003$
4	$0,0345 \cdot 0,0151 + 0,0376 \cdot 0,0322 = 0,0006$
5	$0,0100 \cdot 0,0151 + 0,0101 \cdot 0,0322 = 0,0002$
7	$0,0161 \cdot 0,0151 + 0,1430 \cdot 0,0322 = 0,0048$

Оценки Δ'_j для $j \notin M$ получаются следующими:

$$\Delta'_1 = 2,2281 - 0,0007 = 2,2274,$$

$$\Delta'_2 = 0,3980 - 0,0003 = 0,3977,$$

$$\Delta'_4 = 0,9929 - 0,0006 = 0,9923,$$

$$\Delta'_5 = 0,1324 - 0,0002 = 0,1322,$$

$$\Delta'_7 = 1,5837 - 0,0048 = 1,5789.$$

Отрицательных оценок среди Δ'_j , $j \notin M$ не появилось. Значит, множество M индексов векторов, вводимых в базис, осталось тем же, данное решение совпадает с оценочным качественным решением.

3. *Точное качественное решение.* Получим теперь точное качественное решение по алгоритму, блок-схема которого представлена на рис. 3.5. В данном случае система $\Lambda \tilde{A} = \tilde{C}$ будет решаться уже не приближенно, как в предыдущем пункте, а точно.

Сформируем матрицу $\tilde{A}$ и вектор $\tilde{C}$. Множество M индексов векторов, вводимых в базис (т. е. номера скважин, которые необходимо отключить), $M = \{3, 6\}$. Вычеркнем строки и столбцы матрицы табл. 3.3 с этими индексами. Тогда элементы матрицы A , находящиеся на пересечении вычеркнутых строк и столбцов, образуют подматрицу $\tilde{A}$, а вычеркнутые элементы строки C образуют вектор $\tilde{C}$:

$$\tilde{A} = \begin{pmatrix} +1,196 & -0,0376 \\ -0,0376 & +1,068 \end{pmatrix},$$

$$\tilde{C} = (0,0181 \quad 0,0344).$$

Решаем систему $\Lambda \tilde{A} = \tilde{C}$:

$$\begin{aligned} +1,196\lambda_3 - 0,0376\lambda_6 &= 0,0181, \\ -0,0376\lambda_3 + 1,068\lambda_6 &= 0,0344. \end{aligned}$$

Получаем $\lambda_3 = 0,01616$, $\lambda_6 = 0,03269$. Заметим, что если при решении системы будет получен отрицательный корень, то это означает, что полученное математическое описание (матрица A) не соответствует реальному объекту. В этом случае матрица A должна быть уточнена. Случая неограниченности в данной задаче быть не может.

Проверяем знаки неравенств системы $\Lambda \tilde{A} \leq \tilde{C}$, где подматрица $\tilde{A}$ составляется из элементов матрицы A , находящихся на пересечении вычеркнутых строк и невычеркнутых столбцов, вектор $\tilde{C}$ составляется из невычеркнутых элементов строки C :

$$\tilde{A} = \begin{bmatrix} -0,0224 & -0,0070 & -0,0345 & -0,0100 & -0,0161 \\ -0,0131 & -0,0070 & -0,0376 & -0,0101 & -0,1430 \end{bmatrix},$$

$$\tilde{C} = (-2,2281 \quad -0,3980 \quad -0,9929 \quad -0,1324 \quad -1,5837).$$

Неравенства имеют такой вид:

$$\left. \begin{aligned} j=1 & -0,0224\lambda_3 - 0,1031\lambda_6 \leq -2,2281, \\ j=2 & -0,0070\lambda_3 - 0,0070\lambda_6 \leq -0,3980, \\ j=4 & -0,0345\lambda_3 - 0,0376\lambda_6 \leq -0,9929, \\ j=5 & -0,0100\lambda_3 - 0,0101\lambda_6 \leq -0,1324, \\ j=7 & -0,0161\lambda_3 - 0,1430\lambda_6 \leq -1,5837. \end{aligned} \right\} \quad (6.7)$$

Подстановка значений $\lambda_3=0,01616$ и $\lambda_6=0,03269$ показывает, что неравенства выполняются со знаком $<$. Значит, получен оптимальный базис. Точное качественное решение совпадает с приближенным, и заключается в следующем:

- скважины 3 и 6 отключены,
- скважины 1, 2, 4, 5, 7 включены на предельные депрессии.

Если бы имелись неравенства (6.7) со знаком $>$, то номера этих неравенств также включались бы в множество M ; при этом потребовалось бы вычеркнуть новые строки и столбцы в таблице 3.3, и вновь начать процедуру (см. блок-схему рис. 3.5) (матрица $\tilde{A}$ окаймляется новыми строками и столбцами, расширяется вектор $\tilde{C}$, решается новая система $\Lambda\tilde{A}=\tilde{C}$ и т. д.). Отыскание корней новой системы $\Lambda\tilde{A}=\tilde{C}$ удобно начинать от значений Λ , полученных на предыдущей итерации. Тогда трудоемкость решения всей задачи определяется количеством вычислений, необходимых для решения системы $\Lambda\tilde{A}=\tilde{C}$, полученной на последней итерации решения задачи.

Отметим, что приведенный алгоритм весьма легко реализуется на ЦВМ. Вся процедура строится на использовании двух стандартных подпрограмм: а) подпрограммы решения системы линейных алгебраических уравнений $\Lambda\tilde{A}=\tilde{C}$; б) подпрограммы умножения вектора Λ на матрицу $\tilde{A}$.

Физическая интерпретация. Рассмотрим, как интерпретируется процедура точного качественного решения в терминах задачи оптимизации нефтедобычи. Запишем задачу (6.1) — (6.3) в следующем виде:

$$C^0Q \rightarrow \max, \quad AX + EQ = B, \quad X \geq 0, \quad Q \geq 0. \quad (6.8)$$

Как указывалось в § 2, алгоритм, изображенный на рис. 3.5, является некоторой модификацией симплекс-метода

основанной на использовании специфики S -задачи. В симплекс-методе оценки Δ_j векторов условий относительно каждого базиса вычисляются либо через определение вектора двойственных оценок Λ либо нахождением коэффициентов a'_{ij} выражения внебазисных переменных через векторы базиса. Предположим, что мы идем по второму пути. Тогда, если в качестве исходного базиса задачи (6.8) взять базис, составленный из векторов матрицы E (тогда $Q=B$), оценки Δ_j внебазисных векторов относительно данного базиса

$$\Delta_j = \sum_{i=1}^n c_i^0 a_{ij} = -c_j, \quad (6.9)$$

где c_j — коэффициенты целевой функции S -задачи (6.1) — (6.3). В первом пункте данного параграфа мы убедились в том, что значение c_j есть не что иное, как приращение суммарной добычи чистой нефти по пласту в целом при подаче единичного возмущения дебита на j -й скважине (см. (6.6)). Если $c_j > 0$ (т. е. $\Delta_j < 0$), то дебит j -й скважины нужно уменьшить до предела, причем, на основании свойства 3 S -задач (см. § 1 данной главы), дебит этой скважины будет оставаться на нижнем пределе и в оптимальной точке.

Итак, пусть мы выделили множество M скважин, имеющих значения $c_j > 0$ ($\Delta_j < 0$) и подлежащих при оптимизации переводу на нижний предел по дебиту (будем говорить — подлежащих отключению). Это делается без каких-либо вычислений при оценочном качественном решении. Теперь, при точном качественном решении, нужно определить номера скважин, которые может потребоваться дополнительно отключить, если исследовать ситуацию, сложившуюся после отключения M упомянутых выше скважин. Оценки внебазисных векторов условий после отключения M скважин аналогично (6.9) записываются так:

$$\Delta'_j = \sum_{i \notin M} c_i^0 a'_{ij}, \quad (6.10)$$

где a'_{ij} — коэффициенты, получаемые после преобразования матрицы A по рекуррентным формулам жордановых исключений с разрешающими элементами a_{jj} , $j \in M$. В пункте 2 § 5 было показано, что при изменении числа рассматриваемых скважин коэффициенты матрицы A взаимовлияния скважин

по дебиту преобразуются по тем же рекуррентным формулам и с теми же разрешающими элементами. Иными словами, коэффициенты a'_{ij} в (6.10) есть не что иное, как элементы матрицы A' взаимовлияния оставшихся скважин по дебиту. Каждый элемент a'_{ij} показывает, насколько в данной ситуации изменится дебит i -й скважины при подаче единичного скачка дебита по j -й скважине.

Значение $c^0_j a'_{ij}$ дает то же изменение, но уже в отношении чистой нефти, а не жидкости (так как c^0_i — коэффициент нефтесодержания). Вопрос о том, имеет ли смысл отключить j -ю скважину, решается в зависимости от того, что больше: потеря нефти $c^0_j a'_{jj}$ от уменьшения дебита j -й скважины или приращение суммарной добычи нефти по пласту $\sum_{i \neq j} c^0_i a'_{ij}$,

вызванное увеличением дебита остальных скважин при подобном возмущении на j -й скважине. Если

$$c^0_j a'_{jj} - \sum_{i \neq j} c^0_i a'_{ij} < 0,$$

то скважину целесообразно отключить. Этот случай соответствует появлению отрицательной оценки (6.10). В противном случае оценка $\Delta'_j \geq 0$ и скважину не следует отключать. В рассматриваемом примере на 7 скважин имеет место последний случай — здесь все $\Delta'_j \geq 0$. Иными словами, в данном примере при точном качественном решении не появляется скважин, подлежащих дополнительному отключению относительно оценочного решения.

Совпадение точного качественного решения с приближенным характерно для задач оптимизации нефтедобычи, так как в данном случае коэффициенты «самовлияния» скважин a_{jj} значительно превосходят коэффициенты «взаимовлияния» a_{ij} ($i \neq j$) и изменения оценок Δ_j при отключении скважин обычно соизмеримы с флуктуацией коэффициентов взаимовлияния скважин.

4. *Решение итерационным методом.* В предыдущем пункте было получено точное качественное решение рассматриваемого примера с помощью алгоритма рис. 3.5, который является некоторой модификацией конечных методов линейного программирования применительно к специфике S -задачи.

Проиллюстрируем теперь на рассматриваемом примере решение задачи оптимизации нефтедобычи с помощью

итерационной процедуры, предложенной в § 2 данной главы.

Будем вести решение, например, по формуле (2.4). Запишем ее так:

$$\lambda_j^{(p+1)} = \frac{1}{a_{jj}} (c_j - \sum_{\substack{i \in M \\ i \neq j}} a_{ij} \lambda_i^{(p)}),$$

где $a_{ij} \leq 0$, $i \neq j$, M — множество индексов положительных переменных $\lambda_i^{(p)}$.

1-я итерация

$$\begin{aligned} \lambda_i^{(0)} &= 0, & i &= 1, 2, 3, 4, 5, 6, 7, \\ \lambda_1^{(1)} &= \frac{-2,2281}{2,3400} = -0,9522, & \lambda_5^{(1)} &= \frac{-0,1324}{3,5000} = -0,0378, \\ \lambda_2^{(1)} &= \frac{-0,3980}{2,1600} = -0,1843, & \lambda_6^{(1)} &= \frac{0,0344}{1,0680} = +0,0322, \\ \lambda_3^{(1)} &= \frac{0,0181}{1,1960} = +0,0151, & \lambda_7^{(1)} &= \frac{-1,5837}{1,6480} = -0,9619, \\ \lambda_4^{(1)} &= \frac{-0,9929}{1,2340} = -0,8040, \end{aligned}$$

Множество M после 1-й итерации: $M = \{3, 6\}$.

2-я итерация

$$\begin{aligned} \lambda_1^{(2)} &= \frac{1}{2,3400} [-2,2281 + (0,0224\lambda_3^{(1)} + 0,0131\lambda_6^{(1)})], \\ \lambda_2^{(2)} &= \frac{1}{2,1600} [-0,3980 + (0,0070\lambda_3^{(1)} + 0,0070\lambda_6^{(1)})], \\ \lambda_3^{(2)} &= \frac{1}{1,1960} [0,0181 + 0,0376\lambda_6^{(1)}], \\ \lambda_4^{(2)} &= \frac{1}{1,2340} [-0,9929 + (0,0345\lambda_3^{(1)} + 0,0376\lambda_6^{(1)})], \\ \lambda_5^{(2)} &= \frac{1}{3,5000} [-0,1324 + (0,01\lambda_3^{(1)} + 0,0101\lambda_6^{(1)})], \\ \lambda_6^{(2)} &= \frac{1}{1,0680} [0,0344 + 0,0376\lambda_3^{(1)}], \\ \lambda_7^{(2)} &= \frac{1}{1,6480} [-1,5837 + (0,0161\lambda_3^{(1)} + 0,143\lambda_6^{(1)})]. \end{aligned}$$

$$\begin{aligned}
 \lambda_1^{(2)} &= -0,9518, & \lambda_5^{(2)} &= -0,0376, \\
 \lambda_2^{(2)} &= -0,1841, & \lambda_6^{(2)} &= -0,0326, \\
 \lambda_3^{(2)} &= 0,0161, & \lambda_7^{(2)} &= -0,964. \\
 \lambda_4^{(2)} &= -0,8032,
 \end{aligned}$$

После второй итерации множество M осталось тем же: $M = \{3, 6\}$. Уже на третьей итерации процедура сходится. Значения переменных $\lambda_j^{(p)}$ по итерациям приведены в таблице 3.4. Множество M по итерациям не меняется.

Ответ. $M_{\text{опт}} = \{3, 6\}$, т. е. отключены должны быть 3-я и 6-я скважины.

Таблица 3.4

Качественное решение задачи итерационным методом

Номер итерации (p) Переменные	1	2	3
$\lambda_1^{(p)}$	-0,9522	-0,9518	-0,9518
$\lambda_2^{(p)}$	-0,1843	-0,1841	-0,1841
$\lambda_3^{(p)}$	+0,0151	+0,0161	+0,0161
$\lambda_4^{(p)}$	-0,8040	-0,8032	-0,8032
$\lambda_5^{(p)}$	-0,0378	-0,0376	-0,0376
$\lambda_6^{(p)}$	+0,0322	+0,0326	+0,0326
$\lambda_7^{(p)}$	-0,9619	-0,9640	-0,9640

Решение примера лишний раз показывает, что нельзя просто отключать сильно обводненные скважины без проведения расчетов. В данном случае, например, отключается 6-я скважина с обводненностью 0,85 %, в то же время остается включенной 5-я скважина, имеющая большую обводненность.

5. *Заключение.* Сделаем некоторые заключения относительно качественного решения. Мы убедились, что качественное решение рассматриваемой задачи оптимизации нефтедобычи показывает, какие именно скважины нефтяного пласта должны быть отключены (или переведены на нижний режим по дебиту). Остальные скважины пласта должны

работать при предельном режиме по депрессиям. Поэтому практически после качественного решения задачи имеется достаточно данных для проведения оптимизации.

Для получения точного качественного решения задачи (6.1) — (6.3) достаточно решить систему линейных алгебраических уравнений (2.3), параллельно проверяя выполнение неравенств (2.2). Размерность системы (2.3) в данном случае равна числу отключаемых при оптимизации скважин, а не общему числу скважин месторождения.

Далее, мы отмечали, что точное качественное решение задачи оптимизации нефтедобычи является очень близким, а обычно даже просто совпадающим с оценочным решением. Более того, приращения оценок Δ_j при переходе от приближенного качественного решения к точному практически соизмеримы с погрешностью определения коэффициентов взаимовлияния скважин. Это важно в следующем отношении. Физически, если имеется отличие точного качественного решения задачи от приближенного решения, то оно заключается в том, что при точном качественном решении оказывается больше отключаемых при оптимизации скважин. Идти на такой ответственный шаг, каким является отключение скважин, можно только в том случае, если точно известно, что это целесообразно в смысле увеличения целевой функции. Именно оценка Δ_j показывает приращение суммарной добычи нефти по пласту при подаче единичного возмущения дебита j -й скважины. И практически отключать следует только те скважины, для которых оценки $\Delta_j < \epsilon$, где $\epsilon < 0$ — некоторая величина, определяемая погрешностью задания исходных данных и погрешностью вычислений.

Таким образом, практически при качественном решении задач оптимизации нефтедобычи, как правило, оказывается достаточно оценочного решения. А оценочное решение задачи (6.1) — (6.3) может быть записано без произведения каких-либо вычислений. Получать оценочное решение может и человек без специальной подготовки.

Итак, появляется возможность практически применять методы оптимизации. Осуществляться это должно следующим образом. На электрической модели, или некоторым другим способом, определяется матрица коэффициентов взаимовлияния по дебиту всех скважин нефтяного пласта. Эти коэффициенты меняются медленно и могут считаться постоянными достаточно длительный период времени. Зна-

чительно быстрее изменяются коэффициенты нефтесодержания продукции скважин. Поэтому задача оптимизации должна пересчитываться с интервалом примерно раз в один-два месяца в зависимости от того, насколько быстро изменились коэффициенты нефтесодержания, и от размеров площади.

Матрица коэффициентов взаимовлияния скважин передается непосредственно на нефтяное месторождение. Оптимальное управление режимом нефтяного пласта будет заключаться в том, что при изменении коэффициентов нефтесодержания скважин (раз в 1—2 месяца) определяются элементы вектора $C=[c_j]$ S -задачи (6.1) — (6.3), и скважины, для которых коэффициенты $c_j > 0$ переводятся на нижний режим по дебиту (или отключаются, если это допускается технологией разработки пласта). Остальные скважины при этом работают с предельными значениями депрессий. Для получения вектора C необходимо умножить вектор-строку S^0 коэффициентов нефтесодержания скважин на матрицу A взаимовлияния скважин. Это вполне осуществимо и при ограниченных возможностях использования счетной техники, так как даже для больших нефтяных месторождений каждая строка матрицы A имеет не более 20—30 ненулевых элементов (коэффициенты взаимовлияния отдаленных скважин практически равны нулю). Причем, умножать на строку S^0 следует не все столбцы матрицы A , а только те, которые соответствуют сильно обводненным скважинам. Малообводненные скважины при оптимизации, очевидно, отключаться не будут, поэтому заранее можно сказать, что коэффициенты c_j для них неотрицательны.

Этап II. Количественное решение. На этапе качественного решения задачи было определено множество M номеров, отключаемых при оптимизации скважин, и назначен режим по депрессиям для остальных скважин. При количественном решении задачи определяются дебиты неотключенных скважин в оптимальной точке. В исходной точке задачи (6.1) — (6.3) дебиты скважин равны элементам правой части условий (6.2):

$$q_i = b_i, \quad i = 1, 2, \dots, n.$$

В оптимальной точке дебиты неотключенных скважин также равны соответствующим элементам правой части, преобразованным после отключения скважин:

$$q_i = b'_i \quad \text{для } i \notin M. \quad (6.11)$$

Следуя введенным ранее обозначениям (см. рис. 3.3), имеем $\tilde{B}=[b_i]$, $i \in M$, $\tilde{\tilde{B}}=[\tilde{b}_i]$, $i \notin M$. Тогда условие (6.11) запишется в виде

$$Q_{\text{онт}} = \tilde{\tilde{B}}' = \tilde{\tilde{B}} - \tilde{A}^T \tilde{X}, \quad (6.12)$$

где $\tilde{X}$ определяется из решения системы

$$\tilde{A} \tilde{X} = \tilde{B}. \quad (6.13)$$

В выражениях (6.12), (6.13) матрицы $\tilde{A}$, $\tilde{\tilde{A}}$ и векторы $\tilde{B}$, $\tilde{\tilde{B}}$ соответствуют оптимальному базису — они получаются при том множестве M , которое имеется на последней итерации качественного решения. В формуле (6.12) вместо матрицы $\tilde{\tilde{A}}_B$ (из рис. 3.3) записывается матрица $\tilde{A}^T$, транспонированная к $\tilde{A}$, так как в задачах оптимизации нефтедобычи исходная матрица A симметрична.

Заметим, что при количественном решении S -задач оптимизации нефтедобычи требуется нахождение не самих переменных x_j S -задачи (это делается по формуле (2.9) из § 2 данной главы), а определение дополнительных переменных q_j , обращающих условия (6.2) в равенства. Поэтому в данных задачах, кроме решения системы (2.9) (или, в данном случае, (6.13)), производятся еще вычисления по формуле (6.12).

Для рассматриваемого примера оптимизации месторождения с семью скважинами система (6.13) имеет вид

$$\begin{aligned} +1,196x_3 - 0,0376x_6 &= 95, \\ -0,0376x_3 + 1,068x_6 &= 90, \\ x_3 &= 82,17, \quad x_6 = 87,16. \end{aligned}$$

Дебиты неотключенных скважин в исходной точке при работе на предельных депрессиях:

$$\tilde{\tilde{B}} = \begin{bmatrix} 35 \\ 15 \\ 90 \\ 25 \\ 70 \end{bmatrix}.$$

Увеличение этих дебитов $\tilde{A}^T \tilde{X}$ после перевода пласта на

оптимальный режим:

$$0,0224 \cdot 82,17 + 0,0131 \cdot 87,16 = 2,98,$$

$$0,007 \cdot 82,17 + 0,007 \cdot 87,16 = 1,19,$$

$$0,0345 \cdot 82,17 + 0,0376 \cdot 87,16 = 6,11,$$

$$0,01 \cdot 82,17 + 0,0101 \cdot 87,16 = 1,69,$$

$$0,0161 \cdot 82,17 + 0,143 \cdot 87,16 = 13,78.$$

Оптимальные значения дебитов неотключенных скважин по (6.12):

$$q_1 = 35 + 2,98 = 37,98 \text{ м}^3/\text{сут},$$

$$q_2 = 15 + 1,19 = 16,19 \text{ м}^3/\text{сут},$$

$$q_4 = 90 + 6,11 = 96,11 \text{ м}^3/\text{сут},$$

$$q_5 = 25 + 1,69 = 26,69 \text{ м}^3/\text{сут},$$

$$q_7 = 70 + 13,78 = 83,78 \text{ м}^3/\text{сут}.$$

Суммарный прирост δL_Σ добычи чистой нефти по пласту, получаемый в результате оптимизации, можно вычислить так:

$$\delta L_\Sigma = \bar{C}_{\text{опт}} \bar{X}_{\text{опт}}.$$

Для рассматриваемого примера

$$\delta L_\Sigma = 0,0181 \cdot 82,17 + 0,0344 \cdot 87,16 = 4,49 \text{ м}^3/\text{сут}.$$

Суммарная добыча чистой нефти по пласту в оптимальной точке

$$L_{\text{опт}} = L_{\text{исх}} + \delta L_\Sigma,$$

где $L_{\text{исх}} = C^0 Q = C^0 B$. Исходной точкой здесь считается режим, когда все скважины пласта работают при предельных значениях депрессий. Для рассматриваемого примера:

$$L_{\text{исх}} = 1,0 \cdot 35 + 0,25 \cdot 15 + 0,05 \cdot 95 + 0,9 \cdot 90 + \\ + 0,1 \cdot 25 + 0,15 \cdot 90 + 1,0 \cdot 70 = 210,5 \text{ м}^3/\text{сут},$$

$$L_{\text{опт}} = 210,5 + 4,49 = 214,99 \text{ м}^3/\text{сут},$$

или, если перейти к измерению чистой нефти в тоннах в сутки, как это принято, то

$$L_{\text{исх}} = 162,1 \text{ т/сут}, \delta L = 3,5 \text{ т/сут}, L_{\text{опт}} = 165,6 \text{ т/сут}.$$

При проведении оптимизации имеет смысл вычислить также следующие весьма интересные показатели: суммарный дебит

жидкости по пласту в исходной и оптимальной точке

$$q_{\Sigma \text{ исх}} = \sum_{j=1}^n q_{j \text{ исх}}, \quad q_{\Sigma \text{ опт}} = \sum_{j=1}^n q_{j \text{ опт}}, \quad \delta q_{\Sigma} = q_{\Sigma \text{ исх}} - q_{\Sigma \text{ опт}};$$

процентное увеличение добычи чистой нефти по пласту и уменьшение добычи жидкости

$$\delta L_{\Sigma} (\%) = \frac{\delta L_{\Sigma} \cdot 100\%}{L_{\text{исх}}}, \quad \delta q_{\Sigma} (\%) = \frac{\delta q_{\Sigma} \cdot 100\%}{q_{\Sigma \text{ исх}}}.$$

Для рассматриваемого примера

$$q_{\Sigma \text{ исх}} = \sum_{i=1}^7 b_i = 420 \text{ м}^3/\text{сут}, \quad q_{\Sigma \text{ опт}} = \sum_{i \in M} b_i = 260,76 \text{ м}^3/\text{сут},$$

$$\delta q = 159,24 \text{ м}^3/\text{сут},$$

$$\delta L_{\Sigma} (\%) = 2,15\%, \quad \delta q_{\Sigma} (\%) = 37,91\%.$$

Таким образом, мы видим, что эффект оптимизации весьма значителен даже для рассмотренного небольшого примера месторождения с семью скважинами. На 37% уменьшается выкачка жидкости из пласта и при этом добыча нефти даже возрастает (на 2,15%). Экономический эффект от внедрения подобных мероприятий огромен *).

§ 7. Решение задачи оптимизации нефтедобычи по матрице коэффициентов взаимовлияния скважин по депрессиям

Пусть имеется матрица A^0 коэффициентов взаимовлияния скважин по депрессиям и задача сформулирована в виде (5.3) — (5.5). Задача (5.3) — (5.5) не является S -задачей, но известно, что обратная матрица $A = (A^0)^{-1}$ — это S -матрица (так как A — матрица коэффициентов взаимовлияния скважин по дебиту). В данном параграфе будет показано, что задача (5.3) — (5.5) может эффективно решаться с помощью алгоритмов, аналогичных алгоритмам решения S -задачи, без специального получения матрицы A и сведения

*) Подчеркнем еще раз, что в данном случае речь идет об эффекте, получаемом при реализации оптимальных режимов скважин на каждый отдельный момент времени. Эта *тактическая* задача должна решаться в рамках тех требований, которые определяются общим *стратегическим* планом разработки нефтяного месторождения на весь период эксплуатации.

данной задачи к S -задаче. Изложение иллюстрируется примером решения задачи оптимизации нефтяного месторождения с семью скважинами. Будет принята следующая последовательность: сначала приводится алгоритм решения, потом числовой пример, а затем дается обоснование процедуры.

1. Алгоритм.

Решение будем проводить, как и ранее, в два этапа: качественное решение, а потом количественное.

Этап I. Качественное решение

а) *Приближенное решение.* Приближенное качественное решение, которое мы называли «оценочным», заключается в данном случае в следующем. Решается система линейных алгебраических уравнений

$$A^0 C = -C^0. \quad (7.1)$$

Элементы $c_j > 0$ соответствуют отключаемым (или переводимым на нижний предел по дебиту) скважинам ($q_j = 0$). Элементы $c_j \leq 0$ соответствуют неотключаемым скважинам ($q_j > 0$). Как и ранее, неотключаемые скважины работают при предельных значениях депрессий, т. е., если $q_j > 0$, то $x_j = 0$. Вместо системы (7.1) можно решать аналогичную систему

$$A^0 \Delta = C^0, \quad (7.2)$$

где $\Delta = -C$. Тогда приближенное качественное решение, очевидно, будет записываться так:

$$\begin{aligned} q_j &= 0, \text{ если } \Delta_j < 0, \\ q_j &> 0, \text{ если } \Delta_j \geq 0. \end{aligned}$$

б) *Точное решение.* Вычеркиваем из системы (7.2) строки и столбцы, соответствующие отключаемым в пункте а) скважинам, и решаем оставшуюся систему

$$\sum_{j \notin M} a_{ij}^0 \Delta_j = c_i^0, \quad i \notin M, \quad (7.3)$$

где, как и ранее, M — множество номеров отключаемых скважин. Если при решении системы (7.3) будут получены отрицательные компоненты Δ_j , то номера этих переменных включаются в множество M , вычеркиваются дополнительно строки и столбцы системы (7.3) с индексами $i \in M$, $j \in M$, и вновь решается оставшаяся система. Процедура оканчивается, когда при решении «оставшейся» системы (7.3) не оказывается отрицательных компонент Δ_j .

Качественное решение записывается следующим образом:

$$\begin{aligned} q_j &= 0 \text{ при } j \in M_{\text{опт}}, \\ q_j &> 0 \text{ при } j \notin M_{\text{опт}}. \end{aligned}$$

Здесь $M_{\text{опт}}$ — множество номеров отключаемых скважин, которое имеется на последней итерации качественного решения.

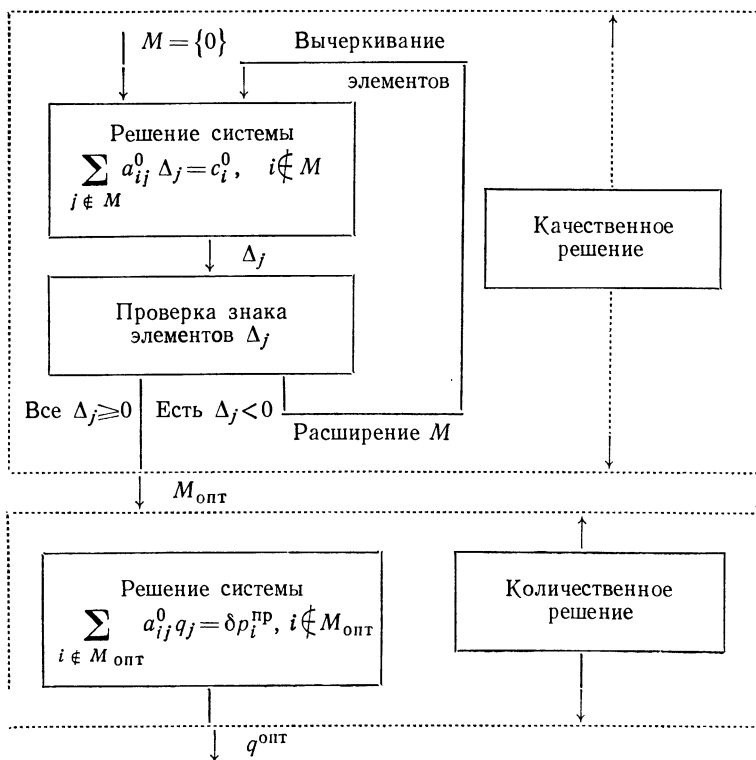


Рис. 3.7. Блок-схема решения задачи оптимизации нефтедобычи по матрице A^0 .

Этап II. Количественное решение

Значения дебитов скважин в оптимальной точке получаются из решения системы

$$\sum_{j \notin M_{\text{опт}}} a_{ij}^0 q_j = \delta p_i^{\text{np}}, \quad i \notin M_{\text{опт}}. \quad (7.4)$$

Блок-схема алгоритма приведена на рис. 3.7. Кроме значений дебитов скважин, имеет смысл определить также некоторые другие показатели (как и при решении задачи по матрице A). Это будут:

добыча чистой нефти по пласту в исходной и в оптимальной точке

$$L_{\text{исх}} = \sum_{j=1}^n c_j^0 q_j^{\text{исх}}, \quad L_{\text{опт}} = \sum_{j=1}^n c_j^0 q_j^{\text{опт}};$$

добыча жидкости по пласту в исходной и оптимальной точке

$$q_{\Sigma \text{ исх}} = \sum_{j=1}^n q_j^{\text{исх}}, \quad q_{\Sigma \text{ опт}} = \sum_{j=1}^n q_j^{\text{опт}};$$

приращения добычи чистой нефти и жидкости по пласту в натуральном и процентном отношении.

2. Числовой пример.

Рассмотрим пример решения задачи оптимизации режима пласта, состоящего из семи скважин. Этот пример решался в предыдущем параграфе с использованием матрицы A . Здесь решение будет проводиться по матрице A^0 .

Исходные данные задачи (5.3) — (5.5) приведены в таблице 3.5 (матрица A^0 , векторы C^0 и $\delta P^{\text{пр}}$).

Т а б л и ц а 3.5

Данные задачи оптимизации месторождения с семью скважинами, сформулированной через матрицу A^0

	A^0							$\delta P^{\text{пр}}$
1	0,4288	0,0078	0,0093	0,0299	0,0065	0,0077	0,0072	20,06
2	0,0078	0,4708	0,0042	0,0198	0,0457	0,0061	0,0126	12,09
3	0,0093	0,0042	0,8382	0,0255	0,0037	0,0321	0,0114	86,10
4	0,0299	0,0198	0,0255	0,8165	0,0235	0,0315	0,0083	81,26
5	0,0065	0,0457	0,0037	0,0235	0,2908	0,0046	0,0036	11,31
6	0,0077	0,0061	0,0321	0,0315	0,0046	0,9499	0,0832	97,67
7	0,0072	0,0126	0,0114	0,0083	0,0036	0,0832	0,6146	52,87
C^0	1	2	3	4	5	6	7	
	1,0	0,25	0,05	0,9	0,1	0,15	1,0	

Этап I. Качественное решение.

а) Приближенное решение. Находим решение системы (7.2) 7-го порядка:

$$\begin{aligned} \Delta_1 &= 2,2281, & \Delta_5 &= 0,1324, \\ \Delta_2 &= 0,3978, & \Delta_6 &= -0,0344, \\ \Delta_3 &= -0,0181, & \Delta_7 &= 1,5839. \\ \Delta_4 &= 0,993, \end{aligned}$$

Имеем:

— скважины 3 и 6 должны быть отключены ($\Delta_3 < 0$, $\Delta_6 < 0$);

— остальные скважины должны работать при предельных значениях депрессий.

б) *Точное решение.* Вычеркиваем третью и шестую строку и столбец матрицы A^0 (так как $M = \{3, 6\}$), и решаем оставшуюся систему (7.3) пятого порядка (см. таблицу 3.5):

$$\begin{aligned} 0,4288\Delta'_1 + 0,0078\Delta'_2 + 0,0299\Delta'_4 + 0,0065\Delta'_5 + & + 0,0072\Delta'_7 = 1,0, \\ 0,0078\Delta'_1 + 0,4708\Delta'_2 + 0,0198\Delta'_4 + 0,0457\Delta'_5 + & + 0,0126\Delta'_7 = 0,25, \\ 0,0299\Delta'_1 + 0,0198\Delta'_2 + 0,8165\Delta'_4 + 0,0235\Delta'_5 + & + 0,0083\Delta'_7 = 0,9, \\ 0,0065\Delta'_1 + 0,0457\Delta'_2 + 0,0235\Delta'_4 + 0,2908\Delta'_5 + & + 0,0036\Delta'_7 = 0,1, \\ 0,0072\Delta'_1 + 0,0126\Delta'_2 + 0,0083\Delta'_4 + 0,0036\Delta'_5 + & + 0,6146\Delta'_7 = 1,0. \end{aligned}$$

Получаем

$$\begin{aligned} \Delta'_1 &= 2,2272, & \Delta'_5 &= 0,1320, \\ \Delta'_2 &= 0,3974, & \Delta'_7 &= 1,5787, \\ \Delta'_4 &= 0,9912, \end{aligned}$$

Итак, $M_{\text{опт}} = \{3, 6\}$. Точное качественное решение совпадает с приближенным.

Этап II. Количественное решение. Решаем систему (7.4) для данной задачи:

$$\begin{aligned} 0,4288q_1 + 0,0078q_2 + 0,0299q_4 + 0,0065q_5 + 0,0072q_7 &= 20,06, \\ 0,0078q_1 + 0,4708q_2 + 0,0198q_4 + 0,0457q_5 + 0,0126q_7 &= 12,09, \\ 0,0299q_1 + 0,0198q_2 + 0,8165q_4 + 0,0235q_5 + 0,0083q_7 &= 81,26, \\ 0,0065q_1 + 0,0457q_2 + 0,0235q_4 + 0,2908q_5 + 0,0036q_7 &= 11,31, \\ 0,0072q_1 + 0,0126q_2 + 0,0083q_4 + 0,0036q_5 + 0,6146q_7 &= 52,87. \end{aligned}$$

Итак, оптимальные значения дебитов

$$\begin{aligned} q_1 &= 37,97 \text{ м}^3/\text{сут}, & q_2 &= 16,17 \text{ м}^3/\text{сут}, & q_4 &= 96,12 \text{ м}^3/\text{сут}, \\ q_5 &= 26,7 \text{ м}^3/\text{сут}, & q_7 &= 83,79 \text{ м}^3/\text{сут}, & q_3 &= 0, & q_6 &= 0. \end{aligned}$$

3. Обсуждение

Описанная процедура решения задачи (5.1) — (5.3) получена по аналогии с алгоритмом рис. 3.5 решения

S -задачи (6.1)—(6.3). Поэтому в данном случае остаются в силе все замечания § 6 о том, что на практике точное качественное решение данной задачи оптимизации нефтесодержания скважин совпадает с приближенным решением.

Для нахождения приближенного решения в случае использования матрицы A требовалось получить вектор C S -задачи (6.1) — (6.3). Это осуществлялось умножением вектора C^0 коэффициентов нефтесодержания скважин на матрицу A . При использовании матрицы A^0 вектор C находится с помощью решения системы (7.1), размерность которой равна общему числу скважин месторождения.

Для получения точного качественного решения в случае использования матрицы A решается система линейных алгебраических уравнений (2.3), размерность которой равна числу *отключаемых* при оптимизации скважин. Если же используется матрица A^0 , то решается система (7.3), размерность которой равна числу *неотключенных* скважин. Поэтому в последнем случае количество вычислений, необходимых для решения, оказывается больше, чем в случае использования матрицы A . Однако это количество вычислений значительно меньше, чем при решении задачи (5.1) — (5.3) общими методами линейного программирования. Например, решение данной задачи оптимизации для нефтяного пласта с 43 скважинами потребовало 104 итерации симплекс-метода. При решении той же задачи описанным выше алгоритмом оказалось достаточно одной итерации, заключающейся в решении системы линейных уравнений (7.3).

4. Обоснование алгоритма.

Рассмотрим связь задачи (5.3) — (5.5) с S -задачей (6.1) — (6.3). Переменные q_j задачи (5.3) — (5.5) являются дополнительными переменными, обращающими условия (6.2) в равенства (см. вывод уравнения (6.5)), а переменные X задачи (6.1) — (6.3)—это дополнительные переменные относительно условий (5.4). Отличие формулировок этих двух задач обуславливается только различием исходного единичного базиса. Для задачи (5.3) — (5.5) исходная точка: $X = \delta P^{\text{пр}}$, $Q = 0$ (все скважины отключены). Для задачи (6.1) — (6.3) исходная точка: $Q = B$, $X = 0$ (все скважины включены на предельные депрессии). Таким образом, задачи (5.3) — (5.5) и (6.1) — (6.3) — это одна и та же задача в различной записи. Поэтому, если при решении задачи (5.3) — (5.5) идти по тем же вершинам, что и при

решении задачи (6.1) — (6.3), то все полезные свойства процедуры решения S -задач будут сохранены.

Исходному опорному плану S -задачи (6.1) — (6.3) соответствует вектор оценок

$$\Delta = C^0 A.$$

Для задачи (5.3) — (5.5), когда вместо матрицы A используется матрица A^0 , оценки, соответствующие тому же опорному плану, получаются из решения системы

$$\Delta A^0 = C^0,$$

где $A^0 = A^{-1}$. Затем при решении S -задачи осуществляется переход в вершину, соответствующую отключению скважин с отрицательными оценками $\Delta_j, j \in M$. Тогда оценки оставшихся внебазисных векторов условий, соответствующие этому опорному плану, можно записать так (по первой форме признака оптимальности [93]):

$$\Delta'_2 = C'_2 \cdot A'_2, \quad (7.5)$$

где A'_2 — подматрица A_2 матрицы A (рис. 3.8), преобразованная по формулам жордановых исключений при введении векторов $\bar{A}_j$ в базис. В случае, если вместо матрицы A

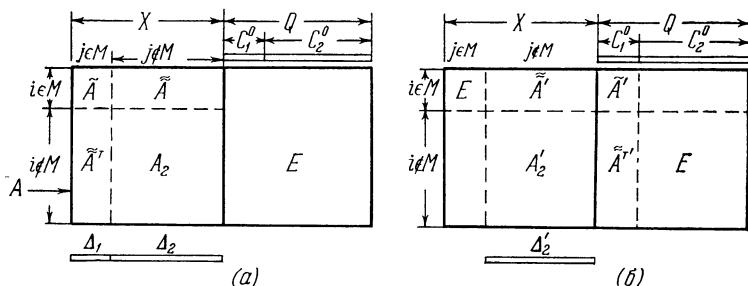


Рис. 3.8. Матрица условий S -задачи (6.1)—(6.3): а) исходная; б) преобразованная.

используется матрица A^0 , оценки (7.5), соответствующие этому опорному плану, получаются из решения системы

$$\Delta'_2 \cdot A'_2 = C'_2, \quad (7.6)$$

где A'_2 — подматрица, полученная вычеркиванием из матрицы

A^0 строк и столбцов с индексами $i \in M, j \in M$ (рис. 3.9). Действительно, выражение (7.5) можно записать в виде

$$\Delta'_2 (A'_2)^{-1} = C_2^0.$$

Нетрудно доказать, что $(A'_2)^{-1} = A_2^0$. Предположим, что требуется получить обратную матрицу $A^{-1} = A^0$ с помощью блочных жордановых исключений. Матрица A разбита на четыре блока в соответствии с рис. 3.8, а на первой блочной итерации в качестве разрешающего элемента берется подматрица $\tilde{A}$. После этого матрица приобретает вид, изображенный на рис. 3.8, б. Здесь

$$A'_2 = A_2 - \tilde{A} \tilde{A}^{-1} \tilde{A}.$$

Если теперь взять в качестве разрешающего блок A'_2 , то на месте единичной матрицы рис. 3.8, а будет получена обратная матрица $A^{-1} = A^0$ (см. рис. 3.9), причем подматрица A_2^0 рис. 3.9 получается в данном случае после преобразования разрешающего блока последней итерации, т.е. $A_2^0 = (A'_2)^{-1}$.

Доказательство выражения (7.6) очевидно также из следующих физических соображений. Матрица A_2 является матрицей коэффициентов взаимовлияния неотключенных скважин

по дебиту (см. § 5, пункт 2). Тогда обратная матрица $(A'_2)^{-1}$ — это матрица A_2^0 коэффициентов взаимовлияния оставшихся скважин по депрессиям. Выше (§ 5, пункт 2) было доказано, что матрица коэффициентов взаимовлияния по депрессиям после отключения некоторых скважин получается из исходной матрицы A^0 вычеркиванием строк и столбцов, соответствующих отключенным скважинам. Таким образом, система (7.3), или, в матричной записи, выражение (7.6) являются аналогом условия (7.5) решения S -задачи, что и требовалось доказать.

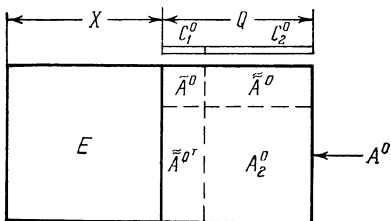


Рис. 3.9. Матрица условий задачи (5.3)—(5.5).

§ 8. Симметричные задачи линейного программирования

Имеется большое число объектов, описываемых системой линейных алгебраических уравнений с симметричной матрицей. Оптимизация подобных объектов часто сводится к решению следующей задачи, которую мы назовем симметричной задачей линейного программирования:

$$CX \rightarrow \max, \quad (8.1)$$

$$\text{при } AX \leq B, \quad (8.2)$$

$$X \geq 0, \quad (8.3)$$

где A — симметричная матрица. Отметим, что все утверждения данного параграфа без труда переносятся на случай любого знака ограничивающих неравенств (8.2) $\leq$, $\geq$, $=$ и при постановке задачи как на $\max$, так и на $\min$ линейной формы.

Использование свойства симметрии матрицы позволило бы вдвое сократить требуемый объем памяти машины и объем вычислений. Однако трудность заключается в том, что в общем случае при решении задачи обычными конечными методами линейного программирования симметричность матрицы, находящейся в оперативной памяти машины, может нарушиться после первой же итерации, а специальные методы линейного программирования для решения задач с симметричной матрицей ограничивающих неравенств (подобно широко известным методам решения систем линейных алгебраических уравнений с симметричной матрицей) нам не известны.

Ниже даны условия сохранения свойства симметрии матрицы при преобразованиях по рекуррентным формулам модифицированных жордановых исключений (м. ж. и.) в процессе решения задачи симплекс-методом или его модификациями. Далее показано, что эти условия будут всегда выполняться, если исходная задача (8.1) — (8.3) сведена к S -задаче с симметричной матрицей ограничений.

Назовем квадратную матрицу $A' = [a'_{ij}]$ размера $n \times n$ симметричной по абсолютным значениям, если $|a'_{s\omega}| = |a'_{\omega s}|$, $s = 1, \dots, n$; $\omega = 1, \dots, n$; $s \neq \omega$, где i, j — индексы соответственно строки и столбца; s, ω (ω, s) — номера, присваиваемые соответственно строке и столбцу произвольного элемента матрицы.

Условия сохранения свойства симметрии достаточно очевидны и заключаются в следующем. Пусть дана симметричная матрица $A = [a_{ij}]$. Тогда матрицы $A' = [a'_{ij}]$, получаемые при преобразованиях матрицы A по рекуррентным формулам м. ж. и., будут симметричны по абсолютным значениям, если разрешающие элементы при преобразованиях всегда будут находиться на главной диагонали.

Доказательство. Условия сохранения свойства симметрии, матрицы по абсолютным значениям после первого преобразования ее по рекуррентным формулам м. ж. и. ($s \neq \omega$)

$$a_{ks}a_{\omega l} = a_{sl}a_{k\omega}, \quad \text{если } \{s, \omega\} \cap \{k, l\} = \emptyset, \quad (8.4)$$

$$a_{ks}a_{\omega l} = a_{\omega s}(a_{kl} \pm 1), \quad \text{если } \{s, \omega\} \cap \{k, l\} \neq \emptyset \quad (8.5)$$

$$a_{ks}a_{\omega l} = a_{s\omega}a_{\omega s} - 1, \quad \text{если } \{s, \omega\} \setminus \{k, l\} = \emptyset, \quad k \neq l. \quad (8.6)$$

Для $\{s, \omega\} \cap \{k, l\} \neq \emptyset$ и $k = l$ элементы $a'_{s\omega}$ и $a'_{\omega s}$ будут отличаться только знаком: $|a'_{s\omega}| = |a'_{\omega s}|$. Здесь k, l — индексы соответственно разрешающей строки и столбца; $\emptyset$ — пустое множество.

Знак перед единицей в правой части условий (8.5) и индексация членов левой части условий (8.5) и (8.6) зависят от вида пересечения множеств $\{s, \omega\}$ и $\{k, l\}$. Условия (8.4) — (8.6) выводятся из предположения, что матрица преобразования по рекуррентным соотношениям м. ж. и. осталась симметричной. При $k = l$ условия (8.5), (8.6) отпадают, а условия (8.4) выполняются, так как при $k = l$, $a_{ks} = a_{sl}$ и $a_{\omega l} = a_{k\omega}$.

После первого преобразования матрицы по рекуррентным формулам м. ж. и. при $k = l$ получаем, что для $\{s, \omega\} \cap \{k, l\} \neq \emptyset$, $a'_{s\omega}$ и $a'_{\omega s}$ отличаются знаком; для $\{s, \omega\} \cap \{k, l\} = \emptyset$, $a'_{s\omega} = a'_{\omega s}$. Если при последующих преобразованиях матрицы по рекуррентным соотношениям м. ж. и. будет соблюдаться условие $k = l$, то матрица сохранит симметричность по абсолютным значениям, так как будут выполняться условия

$$|a_{ks}^{(p-1)}a_{\omega l}^{(p-1)}| = |a_{sl}^{(p-1)}a_{k\omega}^{(p-1)}| \quad \text{для } \{s, \omega\} \cap \{k, l\} \neq \emptyset,$$

причем знаки перед произведениями $a_{ks}^{(p-1)} a_{\omega l}^{(p-1)}$ и $a_{sl}^{(p-1)} \times a_{k\omega}^{(p-1)}$ будут различны, если различны знаки $a_{s\omega}^{(p-1)}$ и $a_{\omega s}^{(p-1)}$, и будут одинаковы, если одинаковы знаки $a_{s\omega}^{(p-1)}$ и $a_{\omega s}^{(p-1)}$.

Эти условия обеспечивают выполнение $|a_{s\omega}^{(p)}| = |a_{\omega s}^{(p)}|$ ($s=1, \dots, n; \omega=1, \dots, n; s \neq \omega$), где $a_{ij}^{(p-1)}$ и $a_{ij}^{(p)}$ — значения элементов матрицы, составленной из коэффициентов выражения внебазисных векторов через векторы базиса соответственно перед p -м преобразованием матрицы и после преобразования, что и требовалось доказать.

Из доказанного свойства следует, что матрицы $A' = [a'_{ij}]$, получаемые на итерациях решения симметричной задачи линейного программирования симплекс-методом, симметричны по абсолютным значениям, если задача (8.1) — (8.3) сведена к S -задаче.

Действительно, если задача (8.1) — (8.3) является S -задачей, то разрешающие элементы при решении симплекс-методом всегда будут находиться на главной диагонали квадратной матрицы A' (см. свойство 2 S -задач в § 1 данной главы).

Таким образом, мы убедились, что для использования свойства симметрии достаточно свести задачу (8.1) — (8.3) к S -задаче. Однако имеется большое число объектов, описываемых системой линейных алгебраических уравнений с симметричной матрицей, которая является S -матрицей уже в исходной записи. В качестве примеров подобных объектов можно назвать электрические цепи, многосвязные объекты с симметричными связями, объекты, описываемые уравнениями типа Лапласа, типа Пуассона и другие объекты, рассмотренные в первой главе.

Задачи оптимизации данных систем часто формулируются в виде S -задачи с симметричной матрицей ограничений. Алгоритмы решения S -задач, предлагаемые в § 2 данной главы, позволяют использовать свойство симметрии матрицы. Это очевидно, так как матрица в этих процедурах не преобразуется в процессе решения задачи. Заметим, однако, что иногда имеет смысл проводить решение S -задачи по частям (см. § 3, пункт 1 данной главы), а в этом случае отдельные блоки матрицы будут преобразовываться. Утверждения данного параграфа позволяют заключить, что и в этом случае матрица остается симметричной по абсолютным значениям.

§ 9. Некоторые обобщения

В данном параграфе будут рассмотрены некоторые особенности решения S -задачи в случае произвольного знака ограничивающих неравенств и в случае наличия некоторых дополнительных условий, не сводящихся к специальному виду условий S -задачи.

1. Решение S -задачи в произвольной записи. Рассмотрение алгоритмов решения S -задач было проведено нами применительно к задаче, записываемой в виде (0.1) — (0.3) или в виде (1.1) — (1.3). Эти задачи ставятся на максимум целевой функции, а знаки ограничивающих неравенств в системе (0.2) имеют вид « $\leq$ » (соответственно переменные $x_j^?$ в (1.2) являются дополнительными переменными, обращающими условия (0.2) в равенства). Если же ограничивающие неравенства имеют другие знаки, то переменные $x_j^?$ в (1.2) являются искусственными. Покажем, что все полезные свойства S -задач остаются в силе независимо от того, на максимум или на минимум целевой функции ставится задача, и независимо от знака неравенств (0.2).

Тот факт, что свойства S -задачи не изменяются в случае отыскания минимума целевой функции, является очевидным. Действительно, достаточно умножить целевую функцию на (-1) и мы получим задачу максимизации. При этом изменяются только знаки коэффициентов функции цели. Однако при постановке S -задачи (см. (0.1) — (0.3)) не накладывается никаких ограничений на знаки коэффициентов целевой функции, т. е. свойства S -задачи от этого не изменятся.

Рассмотрим теперь случай, когда среди ограничений (0.2) имеются условия-равенства. Обозначим номера таких условий индексом p . При наличии условий-равенств можно заранее, еще до решения задачи, сказать, что переменные x_j с индексами $j=p$ будут являться ненулевыми в оптимальной точке (более общо, учитывая вырожденный случай, следует говорить, что переменные x_p будут базисными переменными в оптимальной точке). То есть алгоритмы выделения оптимального базиса S -задачи, предложенные выше, несколько не усложнятся — просто можно будет сразу выделить некоторое число условий (0.2), которые должны выполняться как равенства в оптимальной точке, и назвать некоторое число переменных, которые должны быть базисными в оптимальной точке независимо от значений коэффициентов

целевой функции и матрицы ограничений. Например, если все условия (0.2) являются равенствами, то решение задачи будет находиться в точке решения системы

$$\tilde{A}_m \tilde{X} = \tilde{B},$$

где $\tilde{A}_m$ — матрица $\tilde{A}$ максимальной размерности. Если при этом имеются строки исходной матрицы ограничений, не вошедшие в $\tilde{A}_m$, то задача будет несовместной. Если же имеются столбцы исходной матрицы, не вошедшие в $\tilde{A}_m$, и оценки Δ_j этих столбцов отрицательные, то целевая функция данной задачи будет неограничена на рассматриваемом множестве ограничений.

Приведем доказательство процедуры решения S -задачи с условиями-равенствами. Запишем p -е условие:

$$a_{pp}x_p + \sum_{\substack{j=1 \\ j \neq p}}^n a_{pj}x_j = b_p, \quad (9.1)$$

где $a_{pp} \geq 0$, $a_{pj} \leq 0$, $b_p \geq 0$. Учитывая неотрицательность переменных (см. (0.3)), очевидно, что ненулевое равенство (9.1) может выполняться, только если переменные $x_p > 0$ (в невырожденном случае), что и требовалось доказать.

Процедура решения S -задачи останется той же и в том случае, если считать, что исходный опорный план данной задачи мы получаем методом искусственного базиса. Действительно, пусть мы имеем среди условий (0.2) равенства. В случае неравенств ($\leq$) мы сразу имели исходный единичный базис (исходный опорный план), в рассматриваемом же случае исходный опорный план нужно еще получить. Записываем искусственный базис. В данном случае можно сразу назвать векторы, введение которых в базис заменит в нем искусственные. Это — векторы A_j с индексами, равными индексам соответствующих искусственных: $j=p$ (так как разрешающие элементы могут находиться только на главной диагонали). Введем упомянутые векторы A_p в базис, получив тем самым исходный опорный план (если условия задачи совместны, то все разрешающие элементы положительные). После этого, в соответствии с методом искусственного базиса, столбцы, выведенные из базиса, можно больше не рассматривать, т. е. далее можно проводить решение задачи,

используя оставшуюся часть S -матрицы, как обычно при условиях $\leq$.

Рассмотрим теперь случай, когда среди условий (0.2) имеются неравенства со знаком $\geq$. Введением дополнительных переменных x_j^d мы можем перейти от неравенств $\geq$ к равенствам, не нарушив при этом специальный вид матрицы ограничений S -задачи, так как подматрица дополнительных переменных содержит только неположительные элементы. Тем самым S -задача с ограничениями « $\geq$ » сводится к рассмотренному выше случаю, когда среди ограничений (0.2) имеются условия равенства.

2. Решение S -задач с дополнительными ограничениями. На практике часто возникает такая ситуация, что задачи линейного программирования отличаются от задач специального вида некоторым числом дополнительных строк или столбцов. В то же время, если в специальную задачу добавить хотя бы одно произвольное условие, то полученная задача перестает относиться к специальному виду, и для ее решения становится неприменим эффективный специальный метод. Поэтому большое значение имеют методы, позволяющие обобщить алгоритмы решения специальных задач на случай наличия дополнительных условий. Такие методы используются, например, при решении транспортных задач с дополнительными ограничениями, эти же методы применимы и при решении S -задач, осложненных дополнительными ограничениями.

Ниже мы опишем применение для данного случая одного из известных методов решения специальных задач с дополнительными ограничениями, а именно — применение метода декомпозиции.

Пусть имеется задача

$$CX \rightarrow \max, \quad (9.2)$$

$$A_0X \leq B_0, \quad (9.3)$$

$$AX \leq B, \quad (9.4)$$

$$X \geq 0, \quad (9.5)$$

где условия (9.4), (9.5) являются ограничениями S -задачи, а условия (9.3) — произвольные. Принцип разложения (декомпозиции) дает возможность решить задачу (9.2) — (9.5) путем решения ряда S -задач с ограничениями (9.4) — (9.5). Опишем алгоритм декомпозиции Данцига — Вульфа в

обозначениях [26]. Выпуклое многогранное множество, определяемое неравенствами (9.4) — (9.5), обозначим через G . Любую точку $X \in G$ можно представить линейной выпуклой комбинацией вершин многогранного множества G и направляющих векторов его неограниченных ребер:

$$X = \sum_{v=1}^N z_v X_v, \quad (9.6)$$

где

$$\sum_{v=1}^N z_v \varepsilon_v = 1, \quad z_v \geq 0, \quad v = 1, \dots, N,$$

X_v — вершины и направляющие векторы неограниченных ребер G ,

z_v — коэффициенты представления точки X через X_v ,

$$\varepsilon = \begin{cases} 1, & \text{если } X_v \text{ — вершина многогранного множества } G; \\ 0, & \text{если } X_v \text{ — направляющий вектор неограниченного ребра } G. \end{cases}$$

Подставив X , выраженное таким образом, в условиях (9.2), (9.3) получаем задачу относительно переменных z_v (Данциг и Вульф называют эту задачу «координирующей» или «главной» [94, 32]):

$$\sum_{v=1}^N \sigma_v z_v \rightarrow \max \quad (9.7)$$

при

$$\sum_{v=1}^N P_v z_v \leq B_0, \quad (9.8)$$

$$\sum_{v=1}^N \varepsilon_v z_v = 1, \quad (9.9)$$

$$z_v \geq 0, \quad v = 1, \dots, N, \quad (9.10)$$

где

$$P_v = A_0 X_v, \quad \sigma_v = C X_v. \quad (9.11)$$

«Главная» задача (9.7) — (9.10) имеет по отношению к исходной задаче (9.2) — (9.5) меньшее число ограничений, равное $(m_0 + 1)$ (m_0 — число условий в системе (9.3)), но значительно

большее число переменных. Однако нам не нужно запоминать все векторы $\bar{P}_v = \begin{bmatrix} P_v \\ \varepsilon_v \end{bmatrix}$ матрицы условий «главной» задачи. Если решать «главную» задачу методом обратной матрицы, то на каждой итерации необходимо иметь только один вектор, а именно — вектор, имеющий наименьшую отрицательную оценку Δ_v (Δ_v — оценки векторов $\begin{bmatrix} P_v \\ \varepsilon_v \end{bmatrix}$ «главной» задачи). Иными словами, на каждой итерации решения «главной» задачи необходимо определить вектор $\begin{bmatrix} P_v \\ \varepsilon_v \end{bmatrix}$, имеющий

$$\min_v \Delta_v = \min_v (\Lambda P_v + \lambda_0 \varepsilon_v - \sigma_v), \quad (9.12)$$

где Λ — вектор оценок условий (9.8) на данной итерации решения «главной» задачи, λ_0 — оценка условия (9.9) на данной итерации. Запись в (9.12) $\Delta_v = \Lambda P_v + \lambda_0 \varepsilon_v - \sigma_v$ следует из той формы признака оптимальности, которая используется в методе обратной матрицы. Подставим в (9.12) выражения P_v и σ_v из (9.11). Тогда получим

$$\min_v \Delta_v = \max_v (C - \Lambda A_0) X_v - \lambda_0 \varepsilon_v. \quad (9.13)$$

Значения X_v определяются условиями (9.4), (9.5). Поэтому для отыскания максимума (9.13) нужно решить следующую задачу относительно переменных x_j :

$$C_\lambda X \rightarrow \max \text{ при } AX \leq B, \quad X \geq 0, \quad (9.14)$$

где $C_\lambda = C - \Lambda A_0$. Задачу (9.14) называют «вспомогательной». При получении оптимума «главной» задачи $\min_v \Delta_v = 0$.

Решение «главной» задачи определяет решение исходной задачи (9.2) — (9.5) через выражение (9.6). Конечность алгоритма очевидна, так как число вершин «главной» задачи конечно.

Итак, процедура решения задачи общего вида (9.2) — (9.5) сведена к решению ряда задач (9.14) специального вида. В данном случае задача (9.14) является S -задачей. Дополнительные условия (9.3) отнесены в целевую функцию задачи

(9.14), аналогично методу множителей Лагранжа. Принцип декомпозиции Данцига — Вульфа, по сути, дает конечный алгоритм для отыскания значения вектора множителей Лагранжа Λ . Это становится понятным, если рассматривать решение исходной задачи (9.2) — (9.5), записанной в канонической форме, как нахождение седловой точки функции Лагранжа, $L(\Lambda, X)$:

$$\min_{\Lambda \geq 0} \max_{X \in G} L(\Lambda, X) = \min_{\Lambda \geq 0} \max_{X \in G} [CX + \Lambda(B_0 - A_0X - X^A)], \quad (9.15)$$

где X^A — вектор дополнительных переменных, обращающих условия (9.3) в равенства. Тогда для каждого фиксированного значения Λ получим целевую функцию задачи (9.14):

$$f(\Lambda) = \max_{X \in G} C_\lambda X + \Lambda B_0, \quad (9.16)$$

где $C_\lambda = C - \Lambda A_0$. Аналогия между задачами (9.16) и (9.14) очевидна.

Процедура декомпозиции, несмотря на изящество, обладает весьма существенным недостатком, связанным с медленной сходимостью процесса. Но после опубликования широко известной работы Данцига и Вульфа [94] в советской и иностранной литературе появилось очень большое число статей, посвященных декомпозиции. И в этих статьях показаны пути ускорения сходимости алгоритмов декомпозиции. Возникло целое направление, названное «блочным программированием» [26].

Основные достижения блочного программирования представляются в следующем. В общем случае процедура декомпозиции заключается в отыскании минимума функции $f(\Lambda)$ [23, 68, 90]. Интуитивно это понятно из условий (9.15), (9.16). Задача минимизации $f(\Lambda)$ специфична тем, что, во-первых, $f(\Lambda)$ задана неявно (для отыскания ее значения при данном Λ нужно решить задачу (9.16)) и, во-вторых, эта функция не гладкая (кусочно-линейная). Заметим, что вторую трудность можно обойти, используя для минимизации метод обобщенных градиентов [90].

В терминах возможных направлений [38] любой метод блочного программирования можно интерпретировать [26] как различные реализации процесса решения задачи минимизации $f(\Lambda)$, отличающиеся способом T выбора возмож-

ного направления и способом E выбора длины шага в данном направлении.

На основе данной схемы формализации методов блочного программирования в [23, 26] получены блочные аналоги основных конечных методов линейного программирования. При этом утверждается, что медленная сходимость процедур декомпозиции является следствием того, что способы T и E в любом методе блочного программирования в течение всего процесса вычислений остаются постоянными. Предлагается строить алгоритмы, основанные на гибких схемах выбора способов T и E . При этом, наиболее эффективным представляется в начале процесса отдать предпочтение итерационному способу (чтобы быстро получить грубое приближение), а затем, по мере замедления сходимости итерационного процесса, перейти к одному из конечных методов выбора возможного направления.

3. Случай наличия только одного дополнительного ограничения. Рассмотрим теперь отдельно более простой, но часто встречающийся случай, когда в S -задаче имеется всего одно дополнительное условие. С такими задачами приходится сталкиваться в приложениях — примером может служить задача оптимизации нефтедобычи, поставленная на минимум затрат при заданном плане добычи нефти, или задача минимизации суммарного отбора жидкости по пласту при условии выполнения заданного плана добычи. В этих случаях S -задача будет осложнена наличием одного дополнительного условия-равенства, определяющего плановое задание по добыче нефти. Кроме того, дополнительное условие может появиться в S -задачах в тех случаях, когда S -матрица ограничений получается не сразу, а после проведения некоторых алгебраических преобразований с исходной матрицей.

Например, пусть рассматривается задача, которая не обладает S -матрицей. Однако матрица ограничений этой задачи такова, что каждая строка и столбец содержат один элемент, превосходящий остальные элементы данной строки и столбца (матрица с диагональным преобразованием). Тогда путем вычитания из каждой строки матрицы некоторого дополнительного условия-равенства может оказаться возможным сведение данной задачи к S -задаче. Но при этом в S -задаче появится одно дополнительное условие-равенство.

Итак, пусть рассматривается задача (9.2) — (9.5), в которой система (9.3) состоит только из одного условия,

$$\sum_{j=1}^n a_j^0 x_j \leq b^0, \quad (9.17)$$

или имеется одно дополнительное условие-равенство

$$\sum_{j=1}^n a_j^0 x_j = b^0. \quad (9.18)$$

Тогда вектор Λ , определяющий целевую функцию «вспомогательной» задачи (9.14), является одномерным. Поэтому отыскать оптимальное значение Λ (и тем самым отыскать решение задачи) можно, считая Λ параметром в целевой функции задачи (9.14). Отсюда очевидны два пути решения рассматриваемой задачи: 1) использование декомпозиции; 2) применение методов параметрического программирования.

Заметим, что указанные методы не являются единственно возможными при решении специальной задачи с одним дополнительным условием (см., например, [41, 66]). Однако, следуя идее изложения, мы не задаемся целью рассмотреть все возможные методы решения различных обобщений S -задачи. Наша задача ограничивается указанием некоторых применимых методов и рассмотрением особенностей их использования в данном случае.

Начнем с метода декомпозиции. Описанный выше алгоритм остается в данном случае тем же самым, могут быть внесены только некоторые упрощения, связанные с наличием лишь одного дополнительного условия. Заметим также, что в том случае, когда имеется условие-равенство (9.18), нужно будет прежде всего получить исходный опорный план «главной» задачи. В идейном отношении этот вопрос не принципиальный — он решается обобщением любого метода отыскания исходного опорного плана на данный случай. Однако практическое осуществление этой операции связано с определенными трудностями (некоторое усложнение программы, увеличение числа итераций и т. д.). Покажем, что в случае наличия только одного дополнительного условия (9.18), исходный опорный план «главной» задачи определяется сразу.

Ограничения (9.8), (9.9) «главной» задачи для данного случая состоят всего из двух строк: одной строкой будет

$$\sum_{v=1}^N P_v z_v = b^0, \quad (9.19)$$

где

$$P_v = A_0 X_v = \sum_{j=1}^n a_j^0 x_{jv};$$

другая строка — это условие (9.9). Пусть имеется базис «главной» задачи, состоящий из векторов $\bar{P}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ и $\bar{P}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Вектор $\bar{P}_1$, имеющий единицу в строке (9.19), является искусственным. Вектор $\bar{P}_2$ образован нулевым планом вспомогательной задачи: $\bar{P}_2 = \begin{bmatrix} P_0 \\ 1 \end{bmatrix}$, где $P_0 = A_0 X_0$, $X_0 = 0$. Идея упрощения процедуры отыскания исходного опорного плана «главной» задачи заключается в том, чтобы назвать такой план «вспомогательной» задачи, что соответствующий ему вектор $\bar{P}_v$ при введении в базис главной задачи наверняка заменит искусственный вектор $\bar{P}_1$.

Пусть имеется вектор, вводимый в базис «главной» задачи: $\bar{P}_v = \begin{bmatrix} P_v \\ \varepsilon_v \end{bmatrix}$, где $P_v = \sum_{j=1}^n a_j^0 x_{jv}$. Чтобы определить вектор, который при этом выводится из базиса, находится, как известно, элемент

$$\theta_m = \min \{ \theta_1, \theta_2 \},$$

где

$$\theta_1 = \frac{b^0}{P_v} = \frac{b^0}{\sum_{j=1}^n a_j^0 x_j}$$

соответствует условию (9.19) (на 1-й итерации коэффициенты выражения вектора $\bar{P}_v$ через векторы базиса равны соответствующим элементам $\bar{P}_v$); $\theta_2 = 1/\varepsilon_v$ соответствует условию (9.9). Вектор $\bar{P}_v$ заменит в базисе вектор $\bar{P}_1$, если $\theta_m = \theta_1$.

Таким образом, требуется определить вектор X_v , при котором $\theta_1 < \theta_2$, т. е.

$$\frac{b^0}{\sum_{j=1}^n a_j^0 x_{j_v}} < \frac{1}{\varepsilon_v}.$$

Очевидно, что этот вектор X_v должен быть либо направляющим вектором неограниченного ребра выпуклого многогранного множества G , образованного условиями S -задачи линейного программирования (тогда $1/\varepsilon_v = \infty$), либо вершиной многогранного множества G , для которой

$$\sum_{j=1}^n a_j^0 x_{j_v} > b^0. \quad (9.20)$$

Отыскать вектор X_v , удовлетворяющий условиям (9.20), можно элементарно. Например, пусть рассматривается задача оптимизации нефтедобычи и дополнительное условие (9.18) является плановым заданием на добычу чистой нефти. При записи S -задачи переменными x_j будут являться разности предельных и реальных значений депрессий по скважинам (см. § 3 первой главы). Тогда для удовлетворения условия (9.20) (в том случае, если это возможно) достаточно выбрать $x_{j_v} = 0$ для $a_j^0 < 0$ и задать максимально возможные значения x_{j_v} для $a_j^0 > 0$.

Рассмотрим теперь путь решения данной задачи с использованием параметрического программирования [22, 26]. Запишем целевую функцию S -задачи (9.14) для данного случая:

$$\sum_{j=1}^n (c_j - \lambda a_j^0) x_j \rightarrow \max. \quad (9.21)$$

Процедура решения исходной S -задачи (9.2), (9.4), (9.5) с одним дополнительным условием (9.17) будет заключаться в следующем. Находится решение $X^{(1)} = [x_j^{(1)}]$ S -задачи (9.2), (9.4), (9.5). В данном случае в задаче (9.21), (9.4), (9.5) значение $\lambda^{(1)} = 0$. Проверяется выполнение при $x_j = x_j^{(1)}$ дополнительного условия (9.17). Если это условие выполняется, значит, $X^{(1)}$ и есть искомое решение задачи (9.2), (9.4), (9.5), (9.17). Если же дополнительное условие не выполняется, то в соответствии с правилами параметрического

программирования [26] определяется последовательность значений $\lambda^{(2)}, \lambda^{(3)}, \dots, \lambda^{(l)}$, при которых меняется оптимальный базис S -задачи (9.21), (9.4), (9.5) (причем $\lambda^{(1)} < \lambda^{(2)} < \lambda^{(3)} < \dots < \lambda^{(l)}$) и рассчитываются соответствующие оптимальные планы $X^{(2)}, X^{(3)}, \dots, X^{(l)}$. Переход от одного решения $X^{(k-1)}$ к следующему $X^{(k)}$ осуществляется как обычно в параметрическом программировании, но при этом учитываются полезные свойства S -задачи (направляющие элементы находятся на главной диагонали и т. д.). Для каждого решения $X^{(k)}$ ($k=1, 2, 3, \dots, l$) также проверяется выполнение условия (9.17). Процедура заканчивается, если при некотором плане $X^{(l)}$ условие (9.17) будет выполнено, либо если окажется, что вся ось $\lambda \geq 0$ просмотрена (т. е. найдены решения $X^{(k)}$ на всех отрезках $[\lambda^{(k)}, \lambda^{(k+1)}]$ и на луче $[\lambda^{(k)}, \infty)$, а условие (9.17) нигде не выполнялось. В первом случае искомое решение задачи (9.2), (9.4), (9.5), (9.17) находится следующим образом:

$$X^* = \alpha X^{(l-1)} + (1 - \alpha) X^{(l)},$$

где

$$\alpha = \frac{b^0 - L^{(l)}}{L^{(l-1)} - L^{(l)}},$$

$L^{(l-1)}, L^{(l)}$ — значения левой части условий (9.17) при $X^{(l-1)}, X^{(l)}$. Второй случай означает, что условия (9.4), (9.5), (9.17) исходной задачи несовместны. Описанная процедура изменится незначительно и тогда, когда вместо условия (9.17) в исходной задаче будет условие (9.18).

§ 10. Анализ устойчивости

Для успешного моделирования физических процессов, подлежащих оптимизации, часто очень важно иметь сведения о требуемой точности модели. Задача ставится так: требуется определить область значений погрешностей исходных данных задачи линейного программирования (элементов матрицы A и столбца B), в пределах которой решение задачи качественно не изменится (т. е. оптимальный базис прямой и двойственной задачи будет состоять из одних и тех же векторов).

В известной литературе по анализу устойчивости решений задач линейного программирования к изменению исход-

ных данных [26, 32, 54] рассматриваются вопросы определения пределов варьирования одного элемента или одной строки (столбца), при которых оптимальный базис не меняется (все остальные элементы, входящие в исходные данные задачи, при этом считаются зафиксированными). Нас же интересует случай, когда меняются все элементы матрицы A . Ниже показан путь решения данной задачи.

Заметим, что решение вопроса об устойчивости в такой постановке значительно сложнее, чем решение самой задачи оптимизации, однако это позволяет в ряде случаев получить более близкую к реальности модель, что оправдывает все, так как точность модели определяет успех внедрения оптимизации. Кроме того, требования к точности данных часто необходимы при постановке задач исследования объекта как источника информации. Специальные свойства S -задачи позволяют упростить анализ устойчивости. Ниже мы рассмотрим сначала процедуру исследования устойчивости для общей задачи линейного программирования, а затем покажем упрощения, которые имеют место при анализе S -задачи.

Пусть имеется общая задача линейного программирования

$$CX \rightarrow \max \quad (10.1)$$

при

$$AX \leq B, \quad (10.2)$$

$$X \geq 0. \quad (10.3)$$

Как известно из теории линейного программирования, решить задачу (10.1) — (10.3) — значит найти n -мерный вектор $X^* = [x_j^*]$ и m -мерный вектор $Y^* = [y_j^*]$, удовлетворяющие следующей системе:

$$AX \leq B, \quad (10.4)$$

$$X \geq 0, \quad (10.5)$$

$$YA \geq C, \quad (10.6)$$

$$Y \geq 0, \quad (10.7)$$

$$CX = BY. \quad (10.8)$$

Здесь X^* — оптимум задачи (10.1) — (10.3), Y^* — оптимум задачи линейного программирования, двойственной к задаче (10.1) — (10.3). Условия (10.8) являются следствием усло-

вий $\sum_{i=1}^m a_{ij}y_i = c_j$ при $x_j \neq 0$ [93] и могут быть учтены как усло-

вия (10.6), обращающиеся в равенства (при выполнении второй теоремы двойственности). Пусть в результате вариации исходных условий решение X^3 , Y^3 изменилось и приняло значение X_1^3 , Y_1^3 . Устойчивость решения X_1^3 , Y_1^3 при вариации исходных данных означает, что те условия системы (10.4) — (10.7), которые выполнялись как равенства при X^3 , Y^3 , остались равенствами и при X_1^3 , Y_1^3 .

Отсюда нетрудно записать условия устойчивости оптимального базиса задачи линейного программирования при вариации исходных данных. Не теряя общности изложения, будем рассматривать случай изменения элементов матрицы A (векторы B и C всегда могут быть отнесены к матрице A). Обозначим: $\delta A = \{\delta a_{ij}\}$ — матрица изменения A ; $\delta Y = [\delta y_j]$, $\delta X = [\delta x_i]$ — изменение векторов X^3 , Y^3 , вызванное изменением A ; Γ — множество индексов $n+m$ условий (10.4) — (10.7), выполняющихся в оптимальной точке X^3 , Y^3 как равенства.

Запишем:

$$\sum_{j=1}^n a_{i_1 j} \delta x_j + \sum_{j=1}^n x_j^3 \delta a_{i_1 j} \leq b_{i_1}^z, \quad (10.9)$$

$$\delta x_{j_1} \geq b_{j_1}^z, \quad (10.10)$$

$$\sum_{i=1}^m a_{i j_2} \delta y_i + \sum_{i=1}^m y_i^3 \delta a_{i j_2} \geq b_{j_2}^z, \quad (10.11)$$

$$\delta y_{i_2} \geq b_{i_2}^z, \quad (10.12)$$

где

$$b_{i_1}^z = \begin{cases} 0, & \text{если } i_1 \in \Gamma, \\ b_{i_1} - \sum_{j=1}^n a_{i_1 j} x_j^3, & \text{если } i_1 \notin \Gamma, \end{cases}$$

$$b_{j_1}^z = \begin{cases} 0, & \text{если } j_1 \in \Gamma, \\ -x_{j_1}^3, & \text{если } j_1 \notin \Gamma, \end{cases}$$

$$b_{j_2}^z = \begin{cases} 0, & \text{если } j_2 \in \Gamma, \\ c_{j_2} - \sum_{i=1}^m a_{i j_2} y_i^3, & \text{если } j_2 \notin \Gamma, \end{cases}$$

$$b_{i_2}^z = \begin{cases} 0, & \text{если } i_2 \in \Gamma, \\ -y_{i_2}^3, & \text{если } i_2 \notin \Gamma. \end{cases}$$

Условия устойчивости (10.9) — (10.12) получены при отбрасывании членов высшего порядка малости $\delta a_{ij}\delta x_j$ и $\delta a_{ij}\delta y_i$, и поэтому являются линейными. Заметим, что если при исследовании устойчивости заданы допустимые значения δx_j , δy_i , или если заданы значения δa_{ij} , то условия устойчивости будут линейными и без отбрасывания членов $\delta a_{ij}\delta x_j$, $\delta a_{ij}\delta y_i$. Заметим также, что равенства в условиях (10.9) — (10.12) сохранились для того же множества индексов Γ , что и в оптимальном базисе исходной «невозмущенной» задачи.

Условия устойчивости (10.9) — (10.12) можно использовать следующим образом:

а) если заданы допустимые δx_j (например, из требуемой точности определения оптимального значения целевой функции задачи (10.1) — (10.3)), то можно получить значения погрешностей δa_{ij} , решив задачу, в которой условия устойчивости фигурируют в качестве ограничений и выбран некоторый критерий (например, максимально допустимая средняя погрешность определения a_{ij});

б) если заданы δa_{ij} — погрешности определения коэффициентов a_{ij} , то можно получить «разброс» оптимальных решений δx_j , δy_i , решив задачу с целевой функцией

$$-\sum_{i=1}^m b_i \delta y_i + \sum_{j=1}^n c_j \delta x_j \rightarrow \max$$

при ограничениях (10.9) — (10.12). Так как в этом случае блоки с переменными δx_j и δy_i оказываются несвязанными, то можно решать отдельно задачу с целевой функцией

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

при ограничениях (10.9) — (10.10), и задачу

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

при ограничениях (10.11) — (10.12);

в) если не заданы ни δx_j , ни δa_{ij} , то можно решить задачу одновременного определения допустимых δx_j и δa_{ij} по некоторому критерию при ограничениях (10.9) — (10.12).

Описанная методика исследования устойчивости может быть применена, например, в задачах оптимизации нефтедо-

бычи для следующих целей: для определения требуемой точности моделирования; для выделения области решения, соответствующей данному базису; в задачах определения оптимальных организационно-технических мероприятий; в некоторых задачах динамической оптимизации и в задачах нахождения коэффициентов взаимовлияния скважин по промысловым данным.

Рассмотрим теперь, что вносят в анализ устойчивости специальные свойства S -задач. Пусть задача (10.1) — (10.3) является S -задачей; матрица A является S -матрицей, вектор $B \geq 0$. Пусть, далее, параметры задачи A , B , C изменяются таким образом, что специальный вид задачи сохраняется. Это обычно соответствует конкретному физическому содержанию задачи, так как структура матрицы обуславливается физикой процесса и изменение параметров в некоторых пределах не может изменить эту структуру. Свойства S -задачи говорят о том, что оптимальный базис ее не зависит от значений компонент вектора B , и при $B \geq 0$ условие неотрицательности опорного плана $X \geq 0$ всегда выполняется. Это означает, что при анализе устойчивости оптимального базиса S -задачи условия (10.9), (10.10) можно не рассматривать. Таким образом, матрица ограничений при анализе устойчивости S -задачи значительно сокращается. Условия (10.9) — (10.10) следует рассматривать только тогда, когда нас интересуют количественные значения параметров δx_j .

§ 11. Обсуждение

В этом параграфе мы попытаемся оценить, какие преимущества в вычислительном отношении дает специальная структура S -задачи по сравнению с общей задачей линейного программирования.

Принципиальные трудности решения задач линейного программирования большой размерности объясняются следующими основными причинами: 1) ограниченные возможности памяти существующих ЦВМ; 2) накопление ошибок округления; 3) ограничение времени решения. Кроме того, при решении задачи встают вопросы, связанные с плохой обусловленностью, линейной зависимостью и пр.

Сведения о проблеме решения задач линейного программирования большой размерности можно почерпнуть,

например, из обзорных статей [25, 27, 63]. Здесь же мы ограничимся только интересующими нас вопросами.

Известны некоторые специальные виды задач линейного программирования, которые эффективно решаются в условиях большой размерности. К таким задачам следует прежде всего отнести транспортную задачу, ее модификации и обобщения. Часто для решения задачи пытаются использовать специальную структуру заполнения матрицы ограничений ненулевыми элементами. Например, этот путь оказывается весьма полезным в том случае, когда большая часть исходной матрицы обладает блочно-диагональной структурой. Тогда, с помощью декомпозиции [94, 23, 68, 90] или некоторой модификацией симплекс-метода (см. например, [37]) удастся в процессе решения задачи сохранить блочно-диагональное заполнение матрицы ненулевыми элементами. Это позволяет уменьшить требования к памяти ЦВМ. Те же методы применяются для решения задач специального вида, осложненных дополнительными ограничениями.

Но как быть в том случае, если рассматриваемая задача не может быть отнесена ни к одному из известных специальных классов и имеет произвольную структуру заполнения матрицы ненулевыми элементами? Именно такой вопрос встает при решении задач линейного программирования, к которым сводится статическая оптимизация многосвязных систем и других задач, рассмотренных в первой главе. Использовать для решения подобных задач многократное разложение практически не представляется возможным из-за плохой сходимости процедуры декомпозиции. В случае применения итеративного агрегирования также немедленно возникает проблема скорости сходимости процесса.

Выход из положения в данном случае может быть найден в выделении нового специального свойства рассматриваемых задач, а именно — в использовании *специальной структуры знаков* элементов матрицы ограничений S -задачи. Рассмотрим это подробнее. При решении общей задачи линейного программирования *конечными* методами приходится либо хранить в памяти машины всю матрицу ограничений, которая будет преобразовываться на каждой итерации, либо хранить обратную матрицу базиса, и преобразовывать на каждой итерации эту матрицу. При преобразованиях растет число ненулевых элементов матрицы, которые нужно записывать в память машины, и, кроме того, происходит на-

копление ошибок округления. Это может сделать принципиально невозможным получение решения.

Можно построить конечные методы линейного программирования таким образом, чтобы при решении задачи не происходило преобразования исходной матрицы. Но тогда на каждом шаге процесса нужно будет решать по две системы линейных алгебраических уравнений: одну — для определения двойственных оценок и нахождения отсюда вектора, вводимого в базис на данной итерации; другую — для выражения этого вектора через векторы базиса, и отсюда определения вектора, выводимого из базиса на данной итерации. При таком построении алгоритма может оказаться недопустимо большим время решения задачи.

Другой путь решения задач линейного программирования большой размерности без преобразования исходной матрицы ограничений заключается в использовании *итерационных* (или «итеративных») методов. В последнее время значительно вырос интерес к группе итеративных методов, основанных на бесконечных процессах решения матричных игр [26, 17, 44, 31]. Но наряду с преимуществами подобные методы обладают серьезным недостатком, заключающимся в крайне медленной сходимости итеративных процедур. Работы по ускорению сходимости итеративных методов ведутся, имеются некоторые обнадеживающие результаты, однако в настоящее время можно говорить о достаточно эффективном использовании итеративных методов только по отношению к некоторым частным задачам.

Мы говорили о проблемах, возникающих при решении общей задачи линейного программирования большой размерности. Посмотрим теперь, что получается при решении S -задачи. Как и ранее, будем отдельно рассматривать использование конечных и итеративных методов.

1. Конечные методы. Прежде всего рассмотрим, что дает специальная структура S -задач при использовании «обычных» алгоритмов, т. е. алгоритмов, *предполагающих преобразование исходной матрицы*.

а) *Относительно памяти машины*. В процессе решения S -задачи векторы, которые выводятся из базиса, не могут быть вновь введены в базис; их можно не запоминать. За счет этого в процессе решения задачи часть памяти машины освобождается. Это может компенсировать рост необходимого количества ячеек памяти, связанный с увеличением

числа ненулевых элементов при преобразованиях матрицы в процессе решения. Кроме того, матрица S -задачи может использоваться по частям даже при произвольном, неблочном заполнении ненулевыми элементами.

б) *Относительно накопления погрешностей вычислений и времени решения.* Процесс решения S -задачи идет только в одну сторону: не может быть такого положения, что один и тот же вектор на одной итерации вводится в базис, а на другой итерации выводится. Причем, все векторы, имеющие хотя бы на одной итерации отрицательные оценки Δ_j , обязательно должны входить в оптимальный базис. Известны также векторы, которые при этом будут выведены из исходного базиса. Поэтому все векторы, имеющие отрицательные оценки, можно легко ввести в базис одновременно, за одну итерацию. Если же при этом появятся еще векторы с отрицательными оценками, то эти векторы также следует все одновременно ввести в базис. Таким образом, число итераций при решении S -задачи резко сокращается: как показывает практика, с учетом точности вычислений обычно требуется от одной до трех итераций. Это соответственно вызывает резкое сокращение времени решения и уменьшение накопления ошибок вычислений (благодаря уменьшению числа преобразований матрицы). Кроме того, в случае симметрии исходной матрицы ограничений требования к памяти и время решения сокращаются еще в два раза, так как при преобразованиях в процессе решения S -задачи свойство симметрии матрицы сохраняется.

Уменьшение числа итераций и времени решения S -задачи относительно общей задачи линейного программирования легко проиллюстрировать на примере задачи оптимизации нефтедобычи. Эта задача удобна для примера в данном случае потому, что может быть сформулирована как в виде общей задачи линейного программирования (5.3) — (5.5), так и в виде S -задачи (6.1) — (6.3). Для нефтяного пласта с 43 скважинами задача оптимизации решалась за 104 итерации симплекс-метода в постановке (5.3) — (5.5) (10 минут машинного времени на БЭСМ-3М). В постановке (6.1) — (6.3) S -задача для того же пласта решалась специальным алгоритмом за две итерации (20—25 секунд машинного времени на БЭСМ-4 с учетом времени ввода и вывода данных).

Построение процедуры решения S -задачи конечными методами, предполагающими преобразование исходной матри-

цы, было описано в § 3 данной главы (см. пункт 1). На примере этой процедуры видны преимущества S -задач.

Однако наибольшие преимущества могут быть получены при решении S -задачи методами, *не использующими преобразования матрицы* (см. § 2). В этом случае, векторы, имеющие отрицательные оценки, не будут вводиться в базис. Будет определяться только изменение оценок остальных векторов, которое было бы получено, если бы мы ввели эти вектора в базис. Таким образом, по одной строке оценок (без преобразования самой матрицы) находятся все векторы, которые должны входить в оптимальный базис — выделяется сразу оптимальный базис.

Вся процедура выделения оптимального базиса заключается в решении *одной* системы линейных алгебраических уравнений, которая окаймляется в процессе решения дополнительными строками и столбцами при появлении новых векторов с отрицательными оценками (см. § 2, пункт 1). Для получения количественных значений переменных должна быть решена еще одна система линейных алгебраических уравнений, матрица которой является транспонированной по отношению к матрице системы, используемой при определении оптимального базиса.

Таким образом, при решении S -задачи конечными методами, без преобразования исходной матрицы, нужно решить *не по две* системы линейных алгебраических уравнений *на каждом шаге* процесса (как в общей задаче линейного программирования), *а всего две* системы: одну — для определения оптимального базиса (качественное решение); другую — для определения количественных значений переменных в оптимальной точке (количественное решение *).

Можно сказать в итоге, что решение рассматриваемой задачи линейного программирования сведено к решению системы линейных алгебраических уравнений. Отметим при этом два факта. Во-первых, в настоящее время проблема решения систем линейных алгебраических уравнений

*) Отметим, что при данном сравнении процедур решения S -задачи и общей задачи линейного программирования мы в обоих случаях учитываем только число систем линейных алгебраических уравнений, которые нужно решить, откидывая одинаковую для обоих случаев операцию проверки оценок Δ_j векторов условий (для чего необходимо кроме того перемножить вектор A , полученный после решения системы, на матрицу A).

большой размерности достаточно разработана. Имеются программы для решения систем до 1200 порядка, исследованы вопросы распределения памяти машины при решении больших систем, вопросы оценки числа итераций и времени решения, проблемы влияния ошибок округления, решения плохо обусловленных систем и др. [16]. Во-вторых, система линейных алгебраических уравнений, к которой сводится решение Р-задач, имеет, как правило, значительно меньшую размерность, чем исходная задача линейного программирования. Размерность системы равняется числу векторов, которые должны быть введены в базис при переходе от исходной точки к оптимальной. При разумном выборе исходного базиса в данных задачах число вводимых векторов значительно меньше общего числа столбцов матрицы ограничений S -задачи.

Рассмотрим, например, задачу оптимизаций нефтедобычи (6.1) — (6.3). Исходная точка — все скважины включены. Введение вектора $\tilde{A}_j$ в базис в данном случае соответствует выключению j -й скважины (либо переводению скважины на нижний предел по производительности, если выключение не допускается). Так как число отключаемых скважин в реальных задачах значительно меньше общего числа работающих скважин месторождения, то размерность системы, к которой сводится решение S -задачи, значительно меньше размерности исходной задачи.

Например, решалась задача оптимизации месторождения с фондом в 100 скважин, и в процессе оптимизации оказалось необходимым отключить шесть скважин: решение задачи линейного программирования размером $m \times n = 100 \times 100$ свелось к решению системы 6-го порядка. А это можно осуществить даже без машины.

Перейдем теперь к другой группе методов решения S -задачи.

2. Итерационные методы. Мы уже отмечали, что основным препятствием на пути применения итерационных методов решения общей задачи линейного программирования является крайне медленная сходимость известных итерационных процедур. При решении S -задачи такого препятствия не существует, так как в данном случае алгоритм может быть построен на основе любого итерационного метода решения систем линейных алгебраических уравнений (см. § 2, пункт 2). При этом условия сходимости итерационных

методов и оценка скорости сходимости при решении S -задачи линейного программирования остаются такими же, как и при решении систем линейных алгебраических уравнений. Таким образом, при решении S -задачи не существует той разницы в отношении к конечным и итерационным методам, которая наблюдается в линейном программировании, в отличие от других разделов вычислительной математики. В данном случае итерационные методы занимают такое же важное место, как и конечные.

К числу других преимуществ S -задач следует отнести простоту алгоритмизации процедуры решения. Для построения конечных алгоритмов используются две стандартные подпрограммы: решение системы линейных алгебраических уравнений и умножение матриц. Итерационные методы строятся на основе простых формул, аналогичных обычным итерационным формулам решения систем линейных алгебраических уравнений.

3. Приближенное решение S -задачи. В настоящее время возможность достаточно просто получить приближенное решение задачи большой размерности оценивается зачастую выше, чем все остальные алгоритмические преимущества. В этом отношении S -задачи находятся в значительно более выигрышном положении, чем общие задачи линейного программирования. Оценочное качественное решение S -задачи (оценочный оптимальный базис) может быть получено без проведения каких-либо вычислений. Имеются весьма важные задачи, в которых такого оценочного решения оказывается вполне достаточно. Примером могут служить задачи оптимизации нефтедобычи. Если же этого решения недостаточно, то, проведя весьма небольшое количество вычислений, мы можем получить базис, более близкий к оптимальному.

На основе алгоритмов, рассмотренных в данной главе, были разработаны программы, позволяющие решать S -задачи до размерности $m \times n = 1000 \times 1000$ на вычислительных машинах типа БЭСМ-4. Программы предназначались в основном для решения задач оптимального управления режимом нефтяных месторождений при наличии технологических ограничений (число скважин — до 1000, число ограничений также не более 1000). Одновременно может быть отключено (или переведено на нижний режим по дебиту) в данных программах не более 90 скважин. Структура

матрицы может быть произвольная. Для сравнения заметим, что обычной программой симплекс-метода на подобных машинах можно решать задачи оптимизации нефтяных месторождений с фондом до 50—60 скважин.

С помощью разработанной программы было решено несколько задач, в том числе задача оптимизации режима эксплуатации пласта ДП Константиновского нефтяного месторождения Башкирии. На основе данных расчетов впервые в истории нефтедобычи были проведены эксперименты по оптимальному управлению режимами скважин пласта ДП Константиновской площади. Эксперименты подтвердили все теоретические посылки и расчеты. Реальные задачи оптимизации нефтедобычи решались с помощью предложенного алгоритма, также на машинах с ограниченной памятью и быстродействием типа «Наири». Качественное решение неоднократно находилось *вручную*.

Все сказанное выше можно кратко резюмировать следующим образом. Нами был рассмотрен достаточно широкий класс задач большой размерности, которые не могут быть отнесены ни к одному из известных специальных классов задач линейного программирования. Было выделено новое специальное свойство, присущее рассматриваемым задачам, которое заключается в специальной структуре знаков элементов матрицы. Эта специфика *S*-задач позволяет при решении получить следующие основные преимущества по отношению к общей задаче линейного программирования. В случае использования для решения конечных методов, предполагающих преобразование исходной матрицы, — уменьшаются требования к памяти и резко сокращается число итераций. Как следствие последнего, уменьшается время решения и накопление ошибок вычислений.

Однако значительно большие преимущества могут быть получены при использовании для решения *S*-задачи методов, не предполагающих преобразование исходной матрицы. Конечная процедура решения *S*-задачи без преобразования исходной матрицы будет сводиться, по сути дела, к решению всего двух систем линейных алгебраических уравнений. Если же строить конечную процедуру, не предполагающую преобразования исходной матрицы, для решения общей задачи линейного программирования, то придется решать по две системы на каждом шаге алгоритма типа симплекс-метода.

Итерационная процедура решения S -задачи линейного программирования может быть построена на основе любого известного итерационного метода решения систем линейных алгебраических уравнений при тех же условиях сходимости и тех же оценках скорости сходимости. Если же строится итерационная процедура для решения общей задачи линейного программирования, то данные условия и оценки оказываются значительно хуже; например, приходится обращаться к итерационным методам решения матричных игр, которые пока обладают весьма медленной сходимостью.

Очень важным преимуществом S -задач является также исключительная простота получения приближенного качественного решения. А так как для некоторых задач подобного решения оказывается вполне достаточно, то появляется возможность приближенно решать данные задачи вручную даже в случае большой размерности.

§ 12. Некоторые соображения по динамической оптимизации многосвязных систем

1. Вводные замечания. В первой главе мы рассмотрели задачу оптимизации разработки нефтяного месторождения в упругом режиме. Мы подчеркнули, что нахождение оптимальных режимов работы скважины и пласта сводится к решению задачи математического программирования в некотором абстрактном пространстве. Нетрудно понять, что для класса многосвязных объектов, функционирование которых определяется обобщенным показателем качества, задача динамической оптимизации сводится к решению задачи математического программирования в функциональном пространстве.

Проблеме математического программирования в абстрактном пространстве в литературе уделялось значительное внимание, и получены весьма интересные результаты (см., например, [33, 28, 24, 71]). В процедурном аспекте основное внимание уделялось градиентным методам. В настоящем параграфе мы приведем некоторые результаты, полученные в работах [62, 84], где основное внимание обращено на возможность распространения идей симплекс-метода на решение задач программирования в абстрактном пространстве.

Ниже рассматриваются линейные объекты, для которых справедлив принцип суперпозиции и для которых показате-

лем оптимальности является линейный функционал регулируемых величин.

2. Формулировка задачи оптимизации в функциональном пространстве. Как показано в [56], в линейной системе многосвязного регулирования (СМР) процессы могут быть описаны уравнением

$$AY = BY_{\text{эТ}} + CF, \quad (12.1)$$

где Y — вектор выходных координат системы, $Y_{\text{эТ}}$ — вект эталонных значений координат, или вектор заданий, F — вектор нагрузки, A, B, C — некоторые матричные дифференциальные операторы обыкновенного вида, причем

$$\deg A > \max(\deg B, \deg C).$$

Кроме того, считаем, что заданы начальные условия

$$Y|_{t=0} = Y_0, \quad \dot{Y}|_{t=0} = \dot{Y}_0. \quad (12.2)$$

Используя преобразование Лапласа, решения системы (12.1) в изображениях можно записать в виде

$$Y(p) = A^{-1}(p) [B(p) Y_{\text{эТ}}(p) + C(p) F(p)] + \Phi(p),$$

где $\Phi(p)$ — вектор, появляющийся в результате учета начальных условий, p — комплексный аргумент. Переходя к оригиналам и учитывая наше замечание относительно степеней операторов, получаем решение в виде

$$Y(t) = \int_0^t K(t-\theta) Y_{\text{эТ}}(\theta) d\theta + P(t), \quad (12.3)$$

где $K(t-\theta)$ — матричное ядро, $P(t)$ — вектор, получающийся в результате учета F и Φ . Для рассматриваемого класса СМР обобщенный показатель качества обычно есть дистрибутивная функция своих компонент. Физический смысл экстремизации состоит в требовании максимизации производительности или накопленного продукта, или минимизации затрат и т. д. Учитывая выражение (12.3), можем написать

$$l = l(Y_{\text{эТ}}), \quad (12.4)$$

где l — дистрибутивный функционал. Что касается ограничений, то мы их будем задавать следующими соотношениями:

$$LY_{\text{эТ}} \leq M(t), \quad GY \leq N(t), \quad (12.5)$$

где L, G — вещественные матрицы, $M(t), N(t)$ — заданные векторы. Используя тривиальные преобразования, можно задачу сформулировать следующим образом: минимизировать

$$\eta = \eta(X) \quad (12.6)$$

при ограничениях

$$\int_0^t H(t-\theta) X(\theta) d\theta + UX(t) = Z(t), \quad X(t) \geq 0, \quad (12.7)$$

где матрица-ядро $H(t-\theta)$, вектор-функция $Z(t)$ и функционал $\eta(X)$ получаются из соответствующих компонент формул (12.3) — (12.5), U — вещественная матрица.

Задача (12.6), (12.7) суть задача линейного программирования в некотором функциональном пространстве. Отметим, что если рассматривать лишь установившиеся состояния, то задача (12.6), (12.7) становится обычной задачей линейного программирования в конечномерном евклидовом пространстве.

Задачу (12.6) — (12.7) можно формулировать и из более абстрактных соображений. Пусть мы имеем некоторую систему, в которой выходной параметр Y и входной X связаны соотношением:

$$Y = BX, \quad (12.8)$$

где B — дистрибутивный оператор (например, интегральное преобразование Фредгольма). Множество возможных входных сигналов обозначим через Ω , а множество возможных выходных сигналов через Ψ (очевидно, Ω и Ψ связаны, но это нас пока не интересует). В качестве показателя качества задан дистрибутивный функционал $l(BX, X)$, причем такой записью мы подчеркиваем, что функционал зависит не только от входных, но и от выходных координат. Задача состоит в отыскании числа

$$l^* = \inf l(X), \quad X \in \Omega, \quad Y \in \Psi, \quad Y = BX, \quad (12.9)$$

причем из соображений, подобных приведенным выше, интересно рассмотреть задачу (12.9) в линейном случае.

3. $\frac{3}{4}$ Программирование в функциональном пространстве Гильберта и процедура решения. Проведем некоторые абстрактные рассуждения. Пусть X — функциональное пространство Гильберта $L_2(\mu_q)$, т. е.

множество всех суммируемых по Лебегу функций, определенных на измеримом относительно вполне аддитивной меры q множестве μ , точки которого мы будем обозначать через n . Аналогично, $L_2(v_q)$ — пространство Гильберта, образуемое всеми суммируемыми функциями, определенными на измеримом относительно вполне аддитивной меры g множестве v , точки которого будем обозначать через v . Элемент x будем называть *неотрицательным*, если

$$q \{m \in \mu : x(m) < 0\} = 0.$$

Последнее соотношение будем записывать в виде $x \geq 0$. Положим, что A — линейный оператор из X в Y , элемент $b \in Y$ и фиксирован. Определим в X множество K по формуле

$$K = \{x \in X : x \geq 0, Ax = b, b \in Y\}. \quad (12.10)$$

Задача формулируется так: найти число

$$l^* = \inf l(x), \quad (12.11)$$

где l — линейный функционал. Задача (12.11) есть задача линейного программирования в пространстве Гильберта. Качественные результаты для такой задачи известны из работы [28], где классические качественные теоремы Куна и Таккера обобщаются на случай линейного топологического пространства. Но сама процедура решения не найдена. Попытаемся учесть особенности пространства Гильберта и подойти к задаче с точки зрения симплекс-метода Дж. Данцига. Многие из тех посылок, на которых базируется симплекс-метод, удастся перенести на наш случай. Так, нетрудно видеть, что K — замкнутое выпуклое множество. Если оно ограничено, то линейный функционал достигает на нем своих точных границ, причем найдется такая крайняя точка из K , где значение функционала будет равно его точной границе. Признаки крайней точки даются в следующей теореме.

Теорема 1. Для того чтобы элемент $x_0 \in K$ был крайней точкой множества K , необходимо и достаточно, чтобы уравнение

$$AP_{\alpha(x_0)} x = 0_y, \quad (12.12)$$

где 0_y — нулевой элемент пространства Y , $P_{\alpha(x_0)}$ — оператор проектирования из X в X , определяемый по формуле

$$P_{\alpha(x_0)} x = \begin{cases} 0 & \text{при } m \in \{m \in \mu : x_0(m) \leq 0\}, \\ x(m) & \text{при } m \in \{m \in \mu : x_0(m) > 0\}, \end{cases} \quad (12.13)$$

не имело ни одного такого решения, для которого можно найти такое число $\eta(x) > 0$, что

$$\left| \frac{x_0}{x} \right| \geq \eta(x) \quad (12.14)$$

почти везде на множестве $\alpha(x_0) \cap \alpha(x)$, причем $q(\alpha(x_0) \cap \alpha(x)) > 0$ и $\alpha(x) = \{m \in \mu : x(m) \neq 0\}$.

Заметим, что линейный функционал в пространстве Гильберта всегда можно представить в виде (x, C) , где C — фиксированный элемент. Признаки того, что решение найдено, и процедура решения следуют из следующей теоремы.

Теорема 2. Если $x^* \in K$ и уравнение

$$P_{\alpha(x^*)} A^* y = P_{\alpha(x^*)} C,$$

где A^* — сопряженный с A оператор и $y \in Y$, разрешимо, то условие

$$C - A^* y \geq 0$$

есть достаточное для того, чтобы $x^* \in K_{\text{inf}}$, где

$$K_{\text{inf}} = \{x \in K : l(x) \leq l(x_1), x_1 \in K\}.$$

Если, кроме того, уравнение

$$AP_{\alpha} x = b,$$

где $\alpha \supset \alpha(x^*)$ и $q(\alpha) > q(\alpha(x^*))$, можно разрешить, что $q(\alpha(x) \cap \alpha(x^*)) > 0$, то упомянутое выше условие суть и необходимое.

Делая некоторые предположения, на основе последнего утверждения можно построить абстрактную процедуру решения задачи, которая во многом оказывается аналогичной симплекс-методу. Конкретную процедуру решения задачи мы рассмотрим в следующем случае. Пусть задан интегральный оператор

$$y(v) = \int_{\mu} K(v, m) x(m) dq(m) \quad (v \in v). \quad (12.15)$$

Относительно ядра $K(v, m)$ сделаем следующие предположения:

а) $K(v, m)$ суммируемо на декартовом произведении $\mu \times v$;

б) для всех крайних точек x_0 ядро полно на множестве $\alpha(x_0) \times v$;

в) для любой крайней точки x_0 можно найти такое множество $\alpha \supseteq \alpha(x_0)$, при котором наш оператор имеет обратный из Y в $P_\alpha X$;

г) промежуточное ядро, определяемое из уравнения

$$K(v, m) = \int_{\alpha(x_0)} K(v, \theta) x_0(\theta, m) dq(\theta), \quad v \in v, \quad m \in \mu - \alpha(x_0), \quad (12.16)$$

где x_0 — крайняя точка, есть суммируемая функция на $\alpha(x_0) \times [\mu - \alpha(x_0)]$, причем мера на обоих сомножителях есть q (заметим, что в силу предположений а) б) уравнение (12.16) разрешимо);

д) выполнены условия, указанные во второй части теоремы 2, для любой крайней точки.

Заметим, что мы ничего не говорим о том, зависят ли одни предположения от других и не вытекают ли они одно из другого. Вполне возможно, что это так, но это должно явиться предметом специального исследования. Отметим, что в случае конечномерного евклидова пространства (частный случай пространства Гильберта) все предположения, кроме условия в), выполнены автоматически. Условие в) суть условие невырожденности задачи. В практических случаях сделанные предположения обычно выполняются.

Определим функционал по формуле

$$l(x) = \int_{\mu} c(m) x(m) dq(m), \quad (12.17)$$

где $c(m)$ — суммируемая функция, определяющая функционал. При сделанных предположениях процедура решения задач имеет вид:

1) Находим какую-либо крайнюю точку. Здесь можно применить очевидный аналог метода искусственного базиса.

2) По формуле (12.16) определяем первое промежуточное ядро $\kappa_0(\theta, m)$.

3) Вычисляем A^*y_0 — c по формуле

$$A^*y_0 - c = \begin{cases} 0 & \text{при } m \in \alpha(x_0), \\ -c(m) + \int_{\alpha(x_0)} c(\theta) \kappa_0(\theta, m) dq(\theta) & \text{при } m \in \mu - \alpha(x_0) \end{cases} \quad (12.18)$$

и находим множество

$$\alpha_0^{\oplus} = \{m \in \mu: A^*y_0 - c > 0\}.$$

4) На множестве $\alpha_0^{\oplus} \cup \alpha(x_0)$ решаем задачу поиска новой крайней точки $x_1 \in K$ такой, что $q[\alpha(x_1) \cap \alpha(x_0)] > 0$ по формулам

$$x_0(m) = \begin{cases} \int_{\alpha(x_1) - \alpha(x_1) \cap \alpha(x_0)} \kappa_0(m, \theta) x_1(\theta) dq(\theta) \\ \quad \text{при } m \in \alpha(x_0) - \alpha(x_0) \cap \alpha(x_1), \\ \int_{\alpha(x_1) - \alpha(x_1) \cap \alpha(x_0)} \kappa_0(m, \theta) x_1(\theta) dq(\theta) + x_1 m \\ \quad \text{при } m \in \alpha(x_0) \cap \alpha(x_1), \end{cases} \quad (12.19)$$

5) Определяем новое промежуточное ядро по формулам: при $m \in \mu - \alpha(x_1) \cup \alpha(x_0)$

$$\kappa_0(\theta, m) = \begin{cases} \int_{\alpha(x_1) - \alpha(x_1) \cap \alpha(x_0)} \kappa_0(\theta, s) \kappa_1(s, m) dq(s) \\ \quad \text{при } \theta \in \alpha(x_0) - \alpha(x_1) \cap \alpha(x_0), \\ \kappa_0(\theta, m) + \int_{\alpha(x_1) - \alpha(x_1) \cap \alpha(x_0)} \times \\ \quad \times \kappa_0(\theta, s) \kappa_1(s, m) dq(s) \\ \quad \text{при } \theta \in \alpha(x_1) \cap \alpha(x_0), \end{cases} \quad (12.20)$$

и при каком-либо фиксированном $m \in \mu - \alpha(x_1) \cap \alpha(x_0)$

$$\kappa_1(\theta, m) = \begin{cases} \int_{\alpha(x_0) - \alpha(x_1) \cap \alpha(x_0)} \kappa_1(\theta, s) \kappa_0(s, m) dq(s) \\ \quad \text{при } \theta \in \alpha(x_1) - \alpha(x_0) \cap \alpha(x_1), \\ \kappa_0(\theta, m) + \int_{\alpha(x_0) - \alpha(x_1) \cap \alpha(x_0)} \times \\ \quad \times \kappa_1(\theta, s) \kappa_0(s, m) dq(s) \\ \quad \text{при } \theta \in \alpha(x_0) \cap \alpha(x_1). \end{cases} \quad (12.21)$$

6) Для x_1 выполняем пункты 3) — 5) приведенной процедуры и т. д. Процесс закончен тогда и только тогда, когда либо $\alpha_n^{\oplus}$ пусто, либо $q(\alpha_n^{\oplus}) = 0$.

Приведенная процедура полностью аналогична симплекс-методу со всеми его преимуществами. А именно, нетрудно видеть, что все пункты процедуры выполняются через промежуточное ядро $\kappa_0(\theta, m)$, каждое последующее из которых рекуррентно определяется через предыдущее, т. е. для того чтобы сделать следующий шаг, мы должны знать лишь результаты предыдущего шага. Далее, для вычислений в соответствующих пунктах первых из уравнений систем (12.19) — (12.20) надо решить лишь локальные уравнения Фредгольма первого рода (локальные в смысле множеств, на которых ищется решение). Остальные действия сводятся к интегрированию и алгебраическим операциям. Задача поиска новой крайней точки может решаться лишь качественно по процедуре, аналогичной соответствующей процедуре в симплекс-методе.

В заключение отметим, что для практического использования изложенных результатов, особенно в условиях большой размерности, необходимо еще основательно потрудиться.

Мы привели эти результаты, чтобы привлечь внимание исследователей к этому важнейшему вопросу.

В данной главе будут рассмотрены свойства и специальные алгоритмы решения задач, названных нами S^k -задачами. Речь идет о задачах, матрица ограничений которых состоит из нескольких S -матриц. S^k -задачи возникают при оптимизации нелинейных многосвязных систем, в марковском программировании и т. д. Примеры S^k -задач приводились во второй главе. Здесь же мы рассмотрим свойства и алгоритмы решения S^k -задач вообще и конкретных S^k -задач, возникающих в приложениях.

§ 1. Общие свойства S^k -задач

S^k -задача, т. е. задача линейного программирования, матрица ограничений которой состоит из нескольких S -матриц, может записываться по-разному. В случае, когда S -матрицы располагаются по вертикали (одна под другой), S^k -задача записывается в виде

$$CX \rightarrow \max \text{ при } \left\{ \begin{array}{l} A^{(k)'} X \leq B^k, \quad k = 1, \dots, K, \\ X \geq 0, \end{array} \right. \quad (1.1)$$

где $A^{(k)'} — S$ -матрица. Специфика S^k -задачи (1.1) не нарушается, если задача будет ставиться на минимум целевой функции при ограничивающих неравенствах типа « $\geq$ », или если в задаче не будет условий неотрицательности переменных x_j .

В случае, когда S -матрицы располагаются по горизонтали (одна за другой), S^k -задача записывается так:

$$\left. \begin{array}{l} \text{при} \\ - \sum_{k=1}^K B^k Y^k \rightarrow \max \\ \sum_{k=1}^K A^k Y^k \geq C, \quad Y^k \geq 0, \end{array} \right\} \quad (1.2)$$

где A^k — S -матрицы, $C \geq 0$. S^k -задача (1.2) может ставиться как на максимум, целевой функции, так и на минимум; знаки ограничивающих неравенств также могут быть любые в обоих случаях: « $\geq$ », « $\leq$ », « $=$ ». Кроме специфики знаковой структуры матрицы дополнительно предполагается только неотрицательность вектора C правых частей ограничений. Наличие условий неотрицательности переменных y_i^k в S^k -задаче (1.2) обязательно.

Если мы имеем S^k -задачу, в которой S -матрицы располагаются по вертикали, то всегда можем перейти к двойственной S^k -задаче, в которой S -матрицы расположены по горизонтали (см., например, (1.1) — (1.2)). Удобнее оперировать с задачей (1.2), так как при расположении S -матриц по горизонтали число строк обычно значительно меньше числа столбцов. Будем поэтому вести рассмотрение свойств S^k -задач на основе задачи (1.2). Причем будем считать в этой задаче все ограничения равенствами, так как условия-неравенства типа « $\leq$ » или « $\geq$ » сводятся к равенствам введением матрицы дополнительных переменных, которая является частным случаем S -матрицы. Сведение задачи (1.2) с ограничениями-неравенствами к задаче линейного программирования, записанной в канонической форме, приведет просто к увеличению на единицу общего числа S -матриц S^k -задачи (1.2).

Каждая матрица A^k размера $m \times n$ состоит из m столбцов A_i^k ($i=1, 2, \dots, m$) n -го порядка

$$\sum_{k=1}^K A^k Y^k = \sum_{k=1}^K \sum_{i=1}^m A_i^k y_i^k.$$

Сгруппируем теперь вместе все столбцы с одним и тем же индексом i . Тогда задача (1.2) при ограничениях-равенствах примет удобный для последующего изложения вид

$$-\sum_{i=1}^m \sum_{k=1}^{k_i} b_i^k y_i^k \rightarrow \max \quad (1.3)$$

при

$$\sum_{k=1}^{k_j} a_{jj}^k y_j^k - \sum_{\substack{i=1 \\ i \neq j}}^m \sum_{k=1}^{k_i} a_{ij}^k y_i^k = c_j, \quad j = 1, 2, \dots, n, \quad (1.4)$$

где

$$a_{ij}^k \geq 0, \quad i \neq j, \quad c_j \geq 0, \quad y_i^{(k)} \geq 0, \\ i = 1, 2, \dots, m, \quad k = 1, 2, \dots, k_i. \quad (1.5)$$

Произведенная перегруппировка пояснена на простом трехмерном примере, приведенном на рис. 4.1 и 4.2.

	$k=1$				$k=2$			$k=3$		
	A_1^1	A_2^1	A_3^1	A_4^1	A_1^2	A_2^2	A_3^2	A_1^3	A_2^3	A_3^3
$j=1$	+	—	—	—	+	—	—	+	—	—
$j=2$	—	+	—	—	—	+	—	—	+	—
$j=3$	—	—	+	—	—	—	—	—	—	+

Рис. 4.1. Знаковая структура матрицы задачи (1.2).

Рассмотрим свойства задачи (1.3) — (1.5).

Свойство 1.1. Любой опорный план задачи (1.3) — (1.5) содержит не более, чем по одной положительной переменной y_j^k для каждого j .

	$i=1$			$i=2$			$i=3$			$i=4$
	A_1^1	A_1^2	A_1^3	A_2^1	A_2^2	A_2^3	A_3^1	A_3^2	A_3^3	A_4^1
$j=1$	+ + +			—	—	—	—	—	—	—
$j=2$	—	—	—	+ + +			—	—	—	—
$j=3$	—	—	—	—	—	—	+ — +			—

Рис. 4.2. Знаковая структура матрицы задачи (1.3) — (1.5).

Доказательство. Возьмем произвольный опорный план задачи (1.3) — (1.5), при котором имеется r переменных $y_i^k > 0$. Обозначим через J множество индексов $\{i, k\}$ переменных y_i^k , являющихся положительными: $\{i, k\} \in J$,

если $y_i^k > 0$. Запишем систему (1.4), откинув нулевые переменные y_i^k , $i, k \notin J$:

$$\sum_{\substack{k=1 \\ \{j,k\} \in J}}^{k_j} a_{ij}^k y_j^k - \sum_{\substack{i=1 \\ i \neq j}}^m \sum_{\substack{k=1 \\ \{i,k\} \in J}}^{k_i} a_{ij}^k y_i^k = c_j, \quad j = 1, 2, \dots, n, \quad (1.6)$$

где

$$y_i^k > 0, \quad a_{ij}^k \geq 0 \ (i \neq j), \quad c_j \geq 0.$$

Система (1.6), очевидно, имеет размерность $n \times r$. Выясним теперь, какие из переменных y_i^k исходной задачи могут быть положительными, исходя из того, что должны выполняться условия (1.6), и, соответственно, — условия (1.4). Рассмотрим все возможные случаи.

а) Если $c_j > 0$ и $a_{jj}^k > 0$, то для выполнения каждого j -го линейно-независимого условия системы (1.6) необходимо, чтобы была положительна хотя бы одна из k_j переменных y_j^k . В противном случае, в j -м уравнении (1.6) слева будет неположительное число

$$- \sum_{\substack{i=1 \\ i \neq j}}^m \sum_{\substack{k=1 \\ \{i,k\} \in J}}^{k_i} a_{ij}^k y_i^k \leq 0,$$

а справа положительное число $c_j > 0$, и это уравнение выполняться не может.

б) Если $c_j > 0$ и в j -й строке системы (1.6) не имеется членов $a_{jj}^k > 0$, то из тех же соображений следует, что j -е условие (1.6) не может быть выполнено, т. е. система (1.4), (1.5) несовместна. В данном случае опорного плана задачи (1.3) — (1.5) не существует.

в) Если $c_j = 0$ и $a_{jj}^k > 0$, то при наличии $a_{ij}^k > 0 \ (i \neq j)$ для выполнения каждого j -го условия системы (1.6) необходимо иметь $y_j^k > 0$ хотя бы для одного k . В противном случае j -е условие системы (1.6) может выполняться только, если в (1.6) все $a_{ij}^k = 0 \ (i \neq j)$, т. е. если j -я строка (1.6) является нулевой.

г) Если $c_j = 0$ и в j -й строке системы (1.6) не имеется членов $a_{jj}^k > 0$, то данное j -е условие может выполняться только в том случае, если строка нулевая.

Из проведенного рассмотрения можно сделать следующий вывод: для выполнения каждого j -го линейно-незави-

симого условия системы (1.6) необходимо иметь не менее одного $y_j^k > 0$. В данном выводе предполагается, что система (1.6) совместна, так как в противном случае опорного плана вообще не существовало бы. Нулевые строки системы (1.6) учитываются здесь как частный случай линейно-зависимых.

Из общей теории линейного программирования известно, что система (1.6), соответствующая опорному плану с r положительными переменными y_i^k , содержит r линейно-независимых условий ($r \leq n$).

Таким образом, мы имеем r линейно-независимых условий, для выполнения каждого j -го из них хотя бы одно $y_j^k > 0$, а всего количество положительных y_j^k равно r . Отсюда для каждого j , соответствующего номеру линейно-независимого условия системы (1.6), одно $y_j^k > 0$. Если же говорить не только об r линейно-независимых, но о всех n условиях системы (1.6) (где $r \leq n$), то для каждого j величина $y_j^k > 0$ не более, чем для одного k , что и требовалось доказать.

Из доказанного свойства, в частности, следует, что оптимальный план также содержит не более, чем по одной положительной переменной y_j^k для каждого j . Это утверждение может быть сформулировано в следующем виде.

Свойство 1.1°. Если условия задачи (1.3) — (1.5) совместны, то существует решение, при котором для каждого j положительна не более, чем одна из k_j переменных y_j^k , $k=1, 2, \dots, k_j$.

Свойство 1.1 можно интерпретировать также в терминах, связанных с симплекс-методом. Назовем элементы a_{jj}^k , $k=1, \dots, k_j$, $j=1, \dots, n$ элементами квазидиагонали матрицы ограничений задачи (1.3) — (1.5). Для иллюстрации этого понятия на рис. 4.2 элементы квазидиагонали обведены. Запишем теперь свойство 1.1 следующим образом.

Свойство 1.1'. Решение задачи (1.3) — (1.5) симплекс-методом может быть получено выбором разрешающих элементов только из элементов квазидиагонали.

Это утверждение очевидно, так как, выбирая разрешающие элементы на квазидиагонали, мы всегда для каждого j имеем $y_j^k > 0$ только для одного k (вектор A_j^k , вводимый в базис, в данном случае может заменить в базисе только вектор с тем же индексом j). Из такой формулировки свойства 1.1 становится понятно следующее свойство S^k -задач линейного программирования.

Свойство 1.2. Качественное решение задачи (1.3) — (1.5) не зависит от величины компонент правой части ограничений.

Доказательство. Из приведенной выше интерпретации свойства 1.1 индекс j разрешающей строки при решении задачи симплекс-методом определяется индексом i разрешающего столбца, т. е. не зависит от величины компонент c_j правой части ограничений, что и требовалось доказать.

Напомним, что под качественным решением задачи понимается определение векторов, входящих в оптимальный базис.

§ 2. Дополнительные свойства S^k -задач, возникающих при статической оптимизации нелинейных многосвязных систем

В статическом режиме при учете нелинейностей, сказывающихся в членах главной диагонали S -матрицы, уравнения многосвязного объекта записываются в виде

$$(a_{ii}x_i + \varphi_i(x_i)) - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j = b_i, \quad i = 1, \dots, m, \quad (2.1)$$

где $\varphi_i(x_i)$ — выпуклые дифференцируемые функции. Задача статической оптимизации нелинейного многосвязного объекта (или системы) после кусочно-линейной аппроксимации нелинейных функций сводится к следующей задаче линейного программирования (см. главу II):

$$\sum_{j=1}^n c_j x_j \rightarrow \max \quad (2.2)$$

при

$$(a_{ii} + \alpha_i^k) x_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j \leq b_i^k, \quad k = 1, \dots, k_i, i = 1, \dots, m, \quad (2.3)$$

$$0 \leq x_j \leq \beta_j, \quad j = 1, \dots, n, \quad (2.4)$$

где α_i^k — угловой коэффициент прямой, линеаризирующей $\varphi_i(x_i)$ на k -м участке оси x_i ; $b_i^k = b_i - p_i^k$, где p_i^k — отрезок, отсекаемый k -й линеаризирующей прямой на оси $\varphi_i(x_i)$; k_i — число отрезков, на которые разбивается $\varphi_i(x_i)$ при кусочно-линейной аппроксимации. Элементы $a_{ij} \geq 0$ для

$i \neq j$, элементы $c_j \geq 0$, знаки элементов b_i^k — произвольные. Структура знаков элементов матрицы задачи (2.1) — (2.4) проиллюстрирована на рис. 4.3 на примере простой задачи с тремя переменными x_j . Каждое i -е нелинейное условие аппроксимировано в данном примере тремя линейными ограничениями.

		x_1	x_2	x_3	$(\leq) b_i^k$	
$i=1$	{	+	-	-	.	Условия (2.3)
		+	-	-	.	
		+	-	-	.	
$i=2$	{	-	+	-	.	
		-	+	-	.	
		-	+	-	.	
$i=3$	{	-	-	+	.	
		-	-	+	.	
		-	-	+	.	
		1			.	Условия (2.4)
			1		.	
				1	.	
$C \rightarrow$		+	+	+		Условия (2.2)

Рис. 4.3. Знаковая структура элементов задачи (2.2) — (2.4).

Задача (2.2) — (2.4) является S^k -задачей, так как матрицу условий (2.4) можно рассматривать как частный случай S -матрицы. Кроме специальной структуры знаков элементов матрицы, данная задача обладает еще дополнительной спецификой. Опишем эту дополнительную специфику задачи. Элементы $(a_{ii} + \alpha_i^k)$ будут в данном случае являться элементами квазидиагонали матрицы ограничений. Дополнительная специфика рассматриваемой S^k -задачи заключается в том, что ограничения, входящие в i -ю группу соответствующих условий, отличаются только элементами квазидиагонали матрицы и правыми частями. Эта специфика обуславливает появление весьма полезных дополнительных свойств данной S^k -задачи.

Перейдем от исходной задачи (2.2) — (2.4) к двойственной (число ограничений задачи (2.2) — (2.4) значительно

превышает число переменных):

$$\sum_{i=1}^m \sum_{k=1}^{k_i} b_i^k y_i^k + \sum_{j=1}^n \beta_j y_j^0 \rightarrow \min \quad (2.5)$$

при

$$\sum_{k=1}^{k_j} (a_{jj} + \alpha_j^k) y_j^k - \sum_{\substack{i=1 \\ i \neq j}}^m \sum_{k=1}^{k_i} a_{ij} y_i^k + y_j^0 \geq c_j, \quad j = 1, 2, \dots, n, \quad (2.6)$$

$$y_i^k \geq 0, \quad i = 1, \dots, m, \quad k = 1, \dots, k_i, \quad y_j^0 \geq 0, \quad j = 1, 2, \dots, n. \quad (2.7)$$

Структура знаков элементов данной двойственной задачи для того же трехмерного примера приведена на рис. 4.4. Переменные y_i^k соответствуют условиям (2.3), переменные y_j^0 — условиям (2.4), переменные y_j^π являются дополнительными и обращают условия (2.6) в равенства.

Матрица A_a									y_j^0	y_j^π	(=) C
y_1^k			y_2^k			y_3^k					
+	+	+	-	-	-	-	-	-	1	-1	+
-	-	-	+	+	+	-	-	-	1	-1	+
-	-	-	-	-	-	+	+	+	1	-1	+
Строка B_a .									Строка β_a .		

Рис. 4.4. Знаковая структура задачи (2.5) — (2.7).

Введем следующие обозначения (см. рис. 4.4): A_a — матрица аппроксимированной задачи, соответствующая переменным y_i^k в условиях (2.6); A_i^k — отдельные столбцы матрицы A_a ; B_a , β_a — соответствующие части вектора коэффициентов в целевой функции (2.5).

Исходный опорный план задачи (2.5) — (2.7) удобно взять в точке $y_j^0 = c_j$. Тогда, считая, что задача (2.5) — (2.7) сведена к задаче максимизации, мы можем записать вектор Δ исходных оценок столбцов A_i^k следующим образом:

$$\Delta = B_a - \beta_a A_a.$$

Аналогично вычисляются исходные оценки Δ_j^A столбцов, соответствующих переменным y_j^A :

$$\Delta^A = 0 - \beta_a (-E) = \beta_a,$$

где E — единичная матрица.

Заметим, что появление отрицательной оценки Δ_j^A будет означать неограниченность целевой функции (2.5) на рассматриваемом множестве, так как соответствующие столбцы матрицы будут содержать (как показывается ниже) только неположительные элементы в течение всего процесса решения задачи алгоритмами типа симплекс-метода.

Итак, будем рассматривать следующую задачу:

$$- \sum_{i=1}^m \sum_{k=1}^{k_i} \Delta_i^k y_i^k - \sum_{j=1}^n \Delta_j^A y_j^A \rightarrow \max \quad (2.8)$$

при

$$\sum_{k=1}^{k_j} (a_{jj} + \alpha_j^k) y_j^k - \sum_{\substack{i=1 \\ i \neq j}}^m \sum_{k=1}^{k_i} a_{ij} y_i^k - y_j^A + y_j^0 = c_j, \quad j = 1, \dots, n, \quad (2.9)$$

$$a_{ij} \geq 0, c_j \geq 0, y_j^k \geq 0, y_j^A \geq 0, y_j^0 \geq 0, i = 1, \dots, m, \\ k = 1, \dots, k_i, j = 1, \dots, n. \quad (2.10)$$

Отметим, что рассмотрение могло бы проводиться и на основе исходной задачи (2.2) — (2.4). Проведенные преобразования следует рассматривать как вспомогательные и сделаны они только для удобства последующего изложения.

Приступим к описанию свойств S^k -задач. Общие свойства S^k -задач, описанные в § 1, в данном случае, естественно, остаются в силе. Уточним только, что переменные y_j^A и y_j^0 следует учитывать среди j -х переменных y_j^k . Кроме того, рассматриваемая задача всегда совместна — существует опорный план $y_j^0 = c_j$, где $c_j \geq 0$.

Теперь — о свойствах, обусловленных дополнительной спецификой задачи (2.8) — (2.10). Как и при рассмотрении свойств линейных задач оптимизации многосвязных систем (см. § 1 предыдущей главы), будем пользоваться здесь модифицированными жордановыми исключениями [39] и записывать только матрицу коэффициентов выражения внебазисных векторов через векторы базиса. Пусть A'_a — матрица,

получаемая после преобразования матрицы A_a по рекуррентным формулам модифицированных жордановых исключений (м. ж. и.); $A_i^{k'}$ — отдельные столбцы матрицы A_a .

Свойство 2.1. Если при преобразованиях матрицы A_a по рекуррентным формулам м. ж. и. разрешающий элемент лежит на квазидиагонали, то матрица A_a после введения масштабных коэффициентов на переменные получается такой, что для каждого i столбцы $A_i^{k'}$ ($k=1, \dots, k_i$) будут отличаться по абсолютной величине своих компонент только элементами квазидиагонали.

Доказательство. Если разрешающим является некоторый элемент $a_{rr}^l = (a_{rr} + \alpha_r^l)$, то после преобразования по рекуррентным формулам м. ж. и. для $i=r$ столбцы $A_i^{k'}$ каждой i -й группы ($i \neq r$) будут отличаться только элементами квазидиагонали. В этом легко убедиться из формул преобразования. До преобразования i -я группа столбцов A_i^k ($k=1, \dots, k_i$) была такова, что все элементы a_{ij} строки j были одинаковы, кроме $j=i$ (см. (2.9)). После преобразования те же элементы j -й строки изменяются следующим образом:

$$a'_{ij} = -a_{ij} + \frac{a_{ir}a_{rl}}{a_{rr}^l}.$$

Элементы a_{ir} — одинаковые для всех k -х столбцов i -й группы. Значит, элементы a'_{ij} при $i \neq j$ также одинаковые. Для строки $j=r$ элементы a'_{ij} также одинаковы, так как формулы преобразований здесь

$$a'_{ir} = \frac{a_{ir}}{a_{rr}^l}.$$

Для столбцов $A_r^{k'}$ ($i=r$) рассмотрение тех же формул преобразований показывает, что элементы a'_{ij} j -й строки будут одинаковыми по абсолютной величине, если умножить каждый k -й столбец $A_r^{k'}$ для $k \neq l$ на множитель

$$\eta_r^k = \left| \frac{1}{\alpha_r^l - \alpha_r^k} \right|, \quad k = 1, \dots, k_r, \quad k \neq l,$$

где α_r^l — коэффициент из (2.9), соответствующий разрешающему столбцу; α_r^k — коэффициент, соответствующий k -му столбцу r -й группы (при $k \neq l$). Случай $\alpha_r^l = \alpha_r^k$ рассмотрен в свойстве 2.2 (см. ниже). Из этого свойства следует, что такие столбцы можно не учитывать.

Умножение столбцов $A_r^{k'}$ на коэффициент η_r^k должно сопровождаться введением соответствующих масштабных коэффициентов на переменные, чтобы условия задачи не изменились при умножении.

Что и требовалось доказать.

*Свойство 2.2. Векторы A_r^k матрицы ограничений задачи (2.8) — (2.10), имеющие элементы $\alpha_r^k \geq \alpha_r^l$, не будут входить в оптимальный базис *).*

Доказательство. После преобразования по рекуррентным формулам м. ж. и. с разрешающим элементом a_{rr}^l столбцы A_r^k , имеющие $\alpha_r^k \geq \alpha_r^l$ (и соответственно $a_{rr}^k \geq a_{rr}^l$), будут состоять из неотрицательных компонент. Действительно,

$$a'_{rj} = -a_{rj} + \frac{a_{rr}^k a_{rj}}{a_{rr}^l} = a_{rj} \left(-1 + \frac{a_{rr}^k}{a_{rr}^l} \right),$$

$$j = 1, \dots, n, \quad j \neq r, \quad k \neq l,$$

где $a_{rj} \geq 0$, $a_{rr}^l > 0$. Отсюда, если $a_{rr}^k \geq a_{rr}^l$, то $a_{rj} \geq 0$. Неотрицательны также и соответствующие оценки $\Delta_r^{k'}$ столбцов A_r^k , имеющих $a_{rr}^k \geq a_{rr}^l$,

$$\Delta_r^{k'} = \Delta_r^k - \frac{a_{rr}^k \Delta_r^l}{a_{rr}^l},$$

где

$$a_{rr}^l > 0, \quad \Delta_r^l < 0, \quad |\Delta_r^l| \geq |\Delta_r^k|.$$

Структура знаков элементов матрицы A_a обеспечивает при решении задачи (2.7) — (2.9) симплекс-методом следующее: при преобразованиях матриц A_a' по рекуррентным формулам м. ж. и. с разрешающими элементами, находящимися на квазидиагонали, приращения элементов матриц A_a' и приращения оценок $\Delta_j^{k'}$ будут неотрицательными для столбцов, состоящих из неотрицательных компонент. В то же время эти приращения будут неположительными для остальных столбцов матрицы A_a' . Из последнего следует, что специальная знаковая структура исходной матрицы

*) Напомним, что здесь A_r^l — разрешающий столбец ($i=r$); A_r^k — k -й столбец в r -й группе ($k \neq l$); $a_{rr}^l = a_{rr} + \alpha_r^l$ — разрешающий элемент, а $a_{rr}^k = a_{rr} + \alpha_r^k$ — r -е элементы столбцов A_r^k , где $k \neq l$. Вместо разницы элементов α_r^l и α_r^k можно использовать разницу $a_{rr}^l - a_{rr}^k = \alpha_r^l - \alpha_r^k$.

ограничений сохраняется у матрицы, составленной из столбцов A_i^k при $i \neq r$, и столбцов A_r^k , имеющих элементы $\alpha_r^k < \alpha_r^l$. Столбцы же A_r^k , имеющие элементы $\alpha_r^k \geq \alpha_r^l$, сохраняют неотрицательность своих компонент и неотрицательность оценок Δ_r^k в процессе решения задачи.

Следовательно, эти столбцы не могут войти в оптимальный базис, что и требовалось доказать.

Если столбец A_r^l выбирается разрешающим не сразу, а после нескольких преобразований матрицы с разрешающими элементами, лежащими на квазидиагонали, то доказанное свойство остается в силе, так как соотношения типа « $>$ », « $<$ » между элементами a_{rr}^k и a_{rr}^l сохраняются при подобных преобразованиях.

Свойство 2.3. Векторы, однажды выведенные из базиса в процессе решения задачи (2.8) — (2.10) симплекс-методом, больше не будут вводиться в базис.

Доказательство. В процессе решения S^k -задачи (2.8) — (2.10) симплекс-методом в базис вводятся разрешающие столбцы A_r^l . При этом из базиса выводятся векторы, коэффициентами выражения которых через векторы базиса при использовании м. ж. и. являются компоненты столбцов A_r^l . Из формул м. ж. и. следует неотрицательность компонент столбца A_r^l и неотрицательность соответствующей оценки Δ_r^l :

$$a'_{rj} = \frac{a_{rj}}{a_{rr}^l} \geq 0, \quad j \neq r;$$

$$a'_{rr} < \frac{1}{a_{rr}^l} > 0, \quad \Delta_r^l = \frac{\Delta_r^l}{a_{rr}^l} > 0.$$

А отсюда, как и при доказательстве предыдущего свойства, можно заключить, что благодаря специальной структуре знаков элементов матрицы A_a столбец A_r^l не может иметь отрицательную оценку Δ_r^l в процессе решения задачи симплекс-методом, т. е. не будет больше вводиться в базис, что и требовалось доказать.

Свойства 2.2 и 2.3 можно объединить следующим образом: будем говорить, что столбец A_i^k обладает S -видом, если он содержит не более одного положительного элемента и этот положительный элемент находится в i -й строке.

Свойство 2.4. Векторы A_i^k , не имеющие S -вида, не могут войти в оптимальный базис задачи (2.8) — (2.10).

При решении рассматриваемой задачи конечными методами весьма полезно также использование следующего свойства.

Свойство 2.5. Если на данной итерации решения задачи (2.8) — (2.10) симплекс-методом некоторый столбец A_i^k имеет отрицательную оценку Δ_i^k , то эта оценка останется отрицательной по крайней мере до тех пор, пока не будет введен в базис какой-либо столбец с тем же индексом i .

Доказательство. Как следует из свойства 2.4., при решении задачи симплекс-методом можно не рассматривать столбцы, которые теряют S -й вид при преобразованиях в процессе решения. А для остальных столбцов $A_i^{k'}$ ($i \neq r$) и их оценок $\Delta_i^{k'}$ преобразования по рекуррентным формулам м. ж. и. с разрешающим вектором A_r^l вызывают только неположительные приращения. Иными словами, если оценка $\Delta_i^{k'}$ была отрицательной на данной итерации, то она останется отрицательной и на последующих итерациях, по крайней мере до тех пор, пока разрешающий столбец не окажется в i -й группе ($i=r$), что и требовалось доказать.

Нетрудно проследить связь данных задач с S -задачами, возникающими при статической оптимизации линейных многосвязных систем. S -задачи являются частным случаем S^k -задач, при котором каждая i -я группа столбцов A_i^k состоит только из одного столбца. Свойства 2.3 и 1.1' S^k -задач по аналогии со свойством 2 S -задач (см. § 1 предыдущей главы) звучат так: *при решении S^k -задачи симплекс-методом*

а) каждый столбец матрицы ограничений только один раз может быть разрешающим;

б) разрешающие элементы всегда находятся на квазидиагонали.

Свойства 2.5 и 1.1 аналогичны соответственно свойствам 3 и 1 S -задач. По аналогии с S -задачами, нетрудно убедиться также и в том, что если матрица

$$A = [a_{ij}]$$

линейной части условий (2.1) симметрична, то она останется симметричной по абсолютным значениям в процессе решения S^k -задачи симплекс-методом. Условия сохранения свойства симметрии матрицы при преобразованиях, описанные в § 8 предыдущей главы, в данном случае выполняются.

§ 3. Свойства S^k -задач, возникающих в марковском программировании

Во второй главе рассматривались постановки и примеры задач марковского программирования, т. е. задач, возникающих при оптимизации марковских процессов с доходами. Данные задачи записываются в терминах S^k -задач. Задача оптимизации марковских процессов с доходами при наличии переоценки формулируется в виде следующей S^k -задачи:

$$\left. \begin{aligned} & \sum_{j=1}^N \sum_{k \in K_j} q_j^k \pi_j^k \rightarrow \max \\ \text{при} \quad & \sum_{k \in K_j} \pi_j^k - \beta \sum_{i=1}^N \sum_{k \in K_i} p_{ij}^k \pi_i^k = \varepsilon_j, \quad j = 1, \dots, N, \\ & \pi_i^k \geq 0, \quad i = 1, \dots, N, \quad k \in K_i, \end{aligned} \right\} \quad (3.1)$$

где $0 \leq \beta < 1$, $0 \leq p_{ij}^k \leq 1$, $\varepsilon_j \geq 0$, или в виде двойственной к ней задачи

$$\left. \begin{aligned} & \sum_{j=1}^N \varepsilon_j v_j \rightarrow \min \\ \text{при} \quad & v_i - \beta \sum_{j=1}^N p_{ij}^k v_j \geq q_i^k, \quad i = 1, \dots, N, \quad k \in K_i. \end{aligned} \right\} \quad (3.2)$$

В задаче (3.1) S -матрицы располагаются по горизонтали (см. задачи (1.2) и (1.3) — (1.5)); в задаче (3.2) S -матрицы располагаются по вертикали (см. (1.1)). Решение данных задач определяет номера оптимальных стратегий k , которые должны быть выбраны в каждом i -м состоянии марковского процесса бесконечной длительности. Поэтому нас интересует именно качественное решение задачи. Для задачи (3.1) это будет означать определение номеров ненулевых переменных π_i^k в оптимальной точке; для задачи (3.2) — определение номеров ограничивающих условий, обращающихся в равенства в оптимальной точке. Количественные значения самих переменных π_i^k в данной задаче физического смысла не имеют.

Задача оптимизации марковских процессов с доходами без учета переоценки формулируется в виде S^k -задачи с

одним дополнительным условием (см. вторую главу):

$$\left. \begin{aligned} & \text{при} \quad \sum_{j=1}^N \sum_{k \in K_j} q_j^k \pi_j^k \rightarrow \max \\ & \sum_{k \in K_j} \pi_j^k - \sum_{i=1}^N \sum_{k \in K_i} \pi_i^k p_{ij}^k = 0, \quad j = 1, \dots, N, \\ & \sum_{i=1}^N \sum_{k \in K_i} \pi_i^k = 1, \end{aligned} \right\} \quad (3.3)$$

где $p_{ij}^k \geq 0$, или в виде двойственной к ней задачи

$$\left. \begin{aligned} & \text{при} \quad g \rightarrow \min \\ & g + v_i - \sum_{j=1}^N p_{ij}^k v_j \geq q_i^k, \quad i = 1, \dots, N, \quad k \in K_i. \end{aligned} \right\} \quad (3.4)$$

Приступим к описанию свойств данных задач. Как и в предыдущих параграфах, будем для определенности вести рассмотрение на основе одной из двух двойственных друг относительно друга задач, а именно — на основе задачи (3.1) и задачи (3.3). Начнем с S^k -задачи (3.1), возникающей при наличии переоценки. Общие свойства S^k -задач, описанные в § 1 данной главы, естественно, присущи и этой задаче. Однако можно заметить некоторую дополнительную специфику задачи (3.1), благодаря которой свойства 1.1, 1.1° первого параграфа уточняются в том смысле, что если $\varepsilon_j > 0$, то для каждого j не только не более, но точно по одному $\pi_j^k > 0$. Запишем это.

Свойство 3.1. *Любой опорный план задачи (3.1) при $\varepsilon_j > 0$ содержит по одной положительной переменной π_j^k для каждого j .*

Доказательство. Каждое j -е ограничение задачи (3.1) имеет вид

$$\sum_{k \in K_j} (1 - \beta p_{jj}^k) \pi_j^k - \sum_{\substack{i=1 \\ i \neq j}}^N \sum_{k \in K_i} (\beta p_{ij}^k) \pi_i^k = \varepsilon_j, \quad (3.5)$$

где $0 \leq \beta < 1$, $0 \leq p_{ij}^k \leq 1$, $\pi_i^k \geq 0$, $\varepsilon_j \geq 0$. По аналогии с доказательством свойства 1.1 можно показать, что для выполнения каждого j -го линейно-независимого условия (3.5) при $\varepsilon_j > 0$

необходимо иметь хотя бы одно $\pi_j^k > 0$. В то же время число линейно-независимых условий (3.5) в данной задаче равно N . Это видно из следующего: ранг матрицы ограничений задачи (3.1) равен N , так как всегда можно выделить подматрицу, составленную таким образом, что в нее входит по одному столбцу для каждого j , и эта подматрица удовлетворяет условиям Адамара [19], т. е. имеет ранг N . Таким образом, каждый опорный план задачи (3.1) должен содержать N положительных переменных π_j^k , причем для каждого j не менее одного $\pi_j^k > 0$, где $j=1, \dots, N$. Отсюда, для каждого j одно $\pi_j^k > 0$, что и требовалось доказать.

По аналогии со свойством 1.1° из доказанного следует

Свойство 3.1°. Если условия задачи (3.1) совместны, то при $\varepsilon_j > 0$ существует решение, в котором для каждого j положительна одна из переменных π_j^k , $k \in K_j$.

Заметим, что, исходя из условий выполнения равенств (3.5), можно доказать свойство, аналогичные 3.1 и 3.1°, так же и для случая, когда $\varepsilon_j \geq 0$, но $p_{ij}^k > 0$.

Перейдем теперь к рассмотрению задачи (3.3), возникающей при оптимизации процессов без переоценки. Эта задача отличается от S^k -задач наличием одного дополнительного условия: $\sum_k \sum_i \pi_i^k = 1$. Однако данная задача обладает свойствами [18], аналогичными некоторым общим свойствам S^k -задач.

Свойство 3.2. Любой опорный план задачи (3.3) содержит не более, чем по одной положительной переменной π_j^k для каждого j .

Доказательство. Каждое j -е ограничение задачи (3.3) (кроме последнего) имеет вид

$$\sum_{k \in K_j} (1 - p_{ij}^k) \pi_j^k - \sum_{k \in K_j} \sum_{\substack{i=1 \\ i \neq j}}^N p_{ij}^k \pi_i^k = 0, \quad (3.6)$$

где $0 \leq p_{ij}^k \leq 1$, $\pi_j^k \geq 0$. Для выполнения каждого j -го ненулевого линейно-независимого условия (3.6) при наличии ненулевых π_i^k необходимо иметь хотя бы одно $\pi_j^k > 0$, так как в противном случае в левой части условий (3.6) будет отрицательное число. Нулевой план $\Pi = [\pi_i^k]$ недопустим из-за ограничения $\sum_k \sum_i \pi_i^k = 1$. Поэтому число ненулевых перемен-

ных π_i^k будет определяться рангом системы ограничений задачи (3.3). Ранг этой системы, составленной из $(N+1)$ уравнения, не больше N . Отсюда, любой опорный план содержит не более N переменных $\pi_j^k > 0$ и для каждого j не более одного $\pi_j^k > 0$, что и требовалось доказать. Аналогично записывается, естественно, то же самое свойство и для оптимального плана задачи (3.3).

Заметим, что данное свойство, так же как и свойство 3.1° для процессов с переоценкой, является, по сути дела, некоторой интерпретацией известного свойства оптимальности чистых стратегий в задачах марковского программирования.

Свойство 3.2 можно также интерпретировать следующим образом: *при решении задачи (3.3) симплекс-методом разрешающие элементы находятся на квазидиагонали матрицы, составленной из коэффициентов условий (3.6), или в строке*

$$\sum_{k \in K_i} \sum_{t=1}^N \pi_i^k = 1.$$

§ 4. Алгоритмы решения S^k -задач. I

Специальные свойства S^k -задач позволяют получить определенные преимущества при их решении. На основании общего свойства 1.1 данных задач (см. § 1) из каждой i -й группы столбцов задачи (1.3) — (1.5) только один может входить в базис. Это позволяет разбить задачу на « m » частей и рассматривать отдельно каждый i -й блок столбцов матрицы. При этом каждая i -я группа столбцов может быть задана неявно, в виде некоторого множества точек, определяемого выпуклой областью. На основании того же свойства упрощаются конечные процедуры решения задачи, могут быть построены конечные и итерационные алгоритмы решения, не использующие преобразования исходной матрицы. Второе общее свойство S^k -задач (см. § 1, свойство 1.2) позволяет разделить решение задачи на два этапа: качественное решение (определение оптимального базиса) и количественное решение (определение значений переменных в оптимальной точке).

Дополнительная специфика S^k -задач, возникающих при статической оптимизации нелинейных многосвязных систем, позволяет получить дополнительные алгоритмические преимущества. Данные S^k -задачи характерны тем (см. § 2),

что столбцы A_i^k , входящие в каждую i -ю группу, отличаются только i -ми элементами. Использование этого свойства позволяет существенно уменьшить объем запоминаемой информации и количество вычислений, так как в данном случае можно оперировать не всей матрицей, а только подматрицей, в которую входит по одному столбцу A_i^k из каждой i -й группы; в остальных столбцах требуется запоминать только по одному i -му элементу.

Рассмотрим прежде всего алгоритмы решения S^k -задач, возникающих при оптимизации многосвязных систем.

1. Использование симплекс-метода с учетом специфических свойств задачи (2.8) — (2.10). Алгоритмические особенности, определяемые спецификой задачи, в данном случае будут заключаться в следующем.

а) Записывать в памяти машины следует по одному столбцу A_i^k из каждой i -й группы и элементы квазидиагонали. Иными словами, вместо каждой i -й матрицы размера $n \times k_i$ следует записывать только два окаймляющих ее вектора (n -мерный и k_i -мерный). Это позволяет значительно увеличить размерность задач, допустимых с точки зрения возможностей машины *).

Подобная запись возможна не только при использовании метода обратной матрицы (когда преобразуется только подматрица базисных векторов, а вся матрица остается без изменения в процессе решения задачи), но и при использовании основного алгоритма симплекс-метода, при котором преобразуется вся матрица (§ 2, свойство 2.1). Заметим также, что если матрица A линейной части условий (2.1) симметрична, то свойство симметрии матрицы по абсолютным значениям ее элементов также сохраняется в процессе решения. Запоминать в этом случае следует только наддиагональные элементы матрицы A , составленной таким образом, что в нее входит только по одному столбцу A_i^k для каждого i .

б) При введении в базис каждого вектора A_i^l в процессе решения задачи алгоритмом типа симплекс-метода следует

*) Для рассматриваемой задачи оптимизации нелинейных многосвязных объектов возможность увеличения числа столбцов A_i^k означает, что можно значительно увеличить число аппроксимирующих отрезков, т. е. провести сколь угодно точную кусочно-линейную аппроксимацию нелинейных функций $\Phi_i(x_i)$ в (2.1) без чрезмерного увеличения размерности задачи.

стирать из памяти машины векторы A_r^k , имеющие элементы $\alpha_r^k \geq \alpha_r^l$. Стиранию подлежат также векторы, выведенные из базиса (см. § 2, свойства 2.2 и 2.3). Это позволяет уменьшать массив используемой информации в процессе решения, что может компенсировать увеличение числа запоминаемых ненулевых элементов преобразуемой части матрицы.

Отметим также следующее. Решение задачи ведется в данном случае только в одну сторону в том смысле, что вектор, однажды выведенный из базиса, не может быть обратно введен в базис. При прочих равных условиях (та же исходная точка и т. д.) подобное построение алгоритма обеспечивает наименьшее число итераций, необходимых для решения задачи.

в) Разрешающие элементы при решении задачи симплекс-методом находятся на квазидиагонали. Номер разрешающей строки находить не требуется — он равен индексу i разрешающего столбца. Вектор правых частей ограничений не нужен в процессе решения. Этот вектор следует использовать только после выделения оптимального базиса для определения количественных значений переменных в оптимальной точке.

Приведем описание алгоритма. Как и обычно, решение начинается с выбора начального опорного плана. В задаче (2.8) — (2.10) этот план очевиден: $y_j^0 = c_j$. Если же в рассматриваемой задаче не будет переменных y_j^0 (это случай, когда в исходной задаче (2.2) — (2.4) не имеется условий (2.4)), то следует ввести искусственные переменные с достаточно большими по абсолютной величине коэффициентами целевой функции. Будем считать в этом случае, что соответствующие переменные y_j^0 являются искусственными. Если составить базис из единичных векторов при переменных y_j^0 и выразить все остальные векторы через данный базис (см. переход от задачи (2.5) — (2.7) к (2.8) — (2.10)), то будет получена рассматриваемая задача. Заметим, что в S^k -задачах, так же как и обычно при использовании искусственных переменных, следует проверять, не имеется ли искусственных векторов в оптимальном базисе. Если подобные векторы имеются, значит, условия задачи несовместны.

Итак, приступим к решению задачи (2.8) — (2.10), считая, что начальный опорный план $y_j^0 = c_j$ у нас имеется всегда. Записываем в память машины матрицу $A = [a_{ij}]$ размера $n \times m$ и элементы $a_{ij}^k = a_{ij} + \alpha_j^k$, $k = 1, \dots, k_i$ для каждого j .

Элемент a_{ij}^k в данном случае записывается вместо вектора A_{ij}^k . (Заметим, что полная матрица ограничений задачи (2.8) — (2.10) имеет размер $n \times m \times k$.) Записываем также начальные значения оценок Δ_i^k внебазисных векторов условий. Если все оценки неотрицательны, значит, начальный опорный план оптимален. Если же имеются отрицательные оценки Δ_i^k , то выбираем для введения в базис вектор A_r^l , имеющий оценку

$$\Delta_r^l = \min_{\Delta_i^k < 0} \Delta_i^k.$$

Пусть решение ведется основным алгоритмом симплекс-метода. Тогда будем преобразовывать исходную матрицу A , элементы a_{ij}^k и оценки Δ_i^k по рекуррентным формулам м. ж. и., беря разрешающим r -й элемент a_{rr}^l вектора A_r^l . До преобразования можно стереть из памяти машины векторы, имеющие элементы $\alpha_r^k \geq \alpha_r^l$ (т. е. $a_{rr}^k \geq a_{rr}^l$), а после проведения преобразований можно стереть и элемент α_r^l . Оставшиеся преобразованные элементы $a_{rr}^{k'}$ и оценки $\Delta_r^{k'}$ следует умножить на соответствующие коэффициенты

$$\eta_r^k = \frac{1}{a_{rr}^l - a_{rr}^k},$$

не забывая, что при этом мы вводим масштабные коэффициенты на переменные. Затем мы вновь проверяем неотрицательность полученных оценок $\Delta_r^{k'}$ и, если имеются $\Delta_r^{k'} < 0$, вновь стираем из памяти машины соответствующие векторы, берем разрешающим элемент $a_{rr}^{l'}$ вектора $A_r^{l'}$ с оценкой $\Delta_r^{l'} = \min_{\Delta_i^{k'} < 0} \Delta_i^{k'}$

и т. д. Заметим, что если элемент $a_{rr}^{l'}$ является неположительным, то это означает неограниченность целевой функции задачи на рассматриваемом множестве. В частности, неограниченность будет, если $\Delta_r^{l'} < 0$. Заметим также, что масштабные коэффициенты, вводимые на второй итерации,

$$\eta_r^{k'} = \frac{1}{a_{rr}^{l'} - a_{rr}^{k'}}.$$

Решение продолжается, как и обычно, до получения всех неотрицательных оценок векторов условий. Это означает, что найдено качественное решение — выделен оптимальный базис. Для получения количественных значений перемен-

ных в оптимальной точке достаточно выразить вектор C через векторы полученного базиса.

Предположим теперь, что задача (2.8) — (2.10) решается модифицированным алгоритмом симплекс-метода. Тогда после выделения вектора A_r^l , вводимого в базис, следует вычислить матрицу A_6^{-1} , обратную к матрице A_6 , составленной из векторов базиса. При этом, если обратная матрица вычисляется методом исключения Гаусса, следует брать за разрешающий r -й элемент столбца A_r^l . Одновременно с вычислением матрицы A_6^{-1} вычисляется, как и обычно, вектор оценок условий задачи λ_j относительно данного базиса. Затем вновь обращаемся к исходной матрице векторов условий, стираем из памяти машины векторы, имеющие элементы $\alpha_r^k \geq \alpha_r^l$, и вектор A_r^l , и после этого вычисляем оценки $\Delta_i^{k'}$ векторов условий относительно нового базиса следующим образом: сначала рассчитываем некоторую величину d_i , оди-

наковую для всех векторов A_i^k i -й группы $\left(d_i = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j \right)$,

а затем уже определяем оценки $\Delta_i^{k'}$ каждого вектора A_i^k i -й группы

$$\Delta_i^{k'} = d_i + (a_{ii}^k \lambda_i + \Delta_i^k), \quad \Delta_i^{n'} = -\lambda + \Delta_i^n. \quad (4.1)$$

Далее, если имеются элементы $\Delta_i^{k'} < 0$, то вновь выбираем вектор, вводимый в базис, и продолжаем решение.

Определение оценок $\Delta_i^{k'}$ для i -й группы векторов A_i^k может быть значительно упрощено в том случае, если один из векторов этой группы является базисным вектором. Покажем это. Пусть столбец A_i^l входит в базис. Тогда оценка $\Delta_i^{l'}$ этого столбца

$$\Delta_i^{l'} = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j + a_{ii}^l \lambda_i + \Delta_i^l = 0.$$

Отсюда

$$d_i = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j = - a_{ii}^l \lambda_i - \Delta_i^l.$$

Подставим это выражение d_i в оценки $\Delta_i^{k'}$ столбцов той же i -й группы (для $k \neq l$). Тогда для тех i , для которых один из

k_i векторов A_i^k является базисным, вместо оценки (4.1) получим

$$\Delta_i^{k'} = (\alpha_{ii}^k - \alpha_{ii}^l) \lambda_i + (\Delta_i^k - \Delta_i^l). \quad (4.2)$$

Как мы видим, данная оценка может быть определена без вычисления $-\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j$. Кроме уменьшения количества вы-

числений это важно также в следующем отношении. При использовании основного алгоритма симплекс-метода требуется хранить в памяти только по одному полному столбцу из каждой i -й группы столбцов A_i^k ($k=1, \dots, k_i$); если же используется модифицированный алгоритм, то требуется запоминать по два полных столбца A_i^k : один в основной матрице внебазисных векторов, другой — в обратной матрице. Однако применение оценки (4.2) позволит не запоминать в матрице внебазисных векторов полный столбец (не запоминать элементы a_{ij} для $j \neq i$) для тех i , для которых один из векторов является базисным.

Оценки $\Delta_i^{k'}$ для тех i , для которых ни одного вектора A_i^k не входит в базис:

$$\Delta_i^{k'} = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j + \Delta_i^k. \quad (4.3)$$

Это следует из (4.1) при $\lambda_i = 0$. Оценки $\Delta_i^{k'}$ остаются теми же:

$$\Delta_i^{k'} = -\lambda_i + \Delta_i^k. \quad (4.4)$$

Таким образом, при решении рассматриваемой задачи модифицированным алгоритмом симплекс-метода весьма удобно вместо оценок (4.1) пользоваться оценками (4.2) — (4.4).

2. Использование блочного введения векторов в базис. В базис, соответствующий любому опорному плану S^k -задачи, входит не более чем по одному вектору из каждой i -й группы векторов A_i^k ($k=1, \dots, k_i$). Естественно возникает желание заменять векторы базиса одновременно во всех m группах векторов A_i^k за одну итерацию. Это позволит значительно уменьшить число итераций, необходимых для решения задачи. Использование блочного введения векторов в базис в рассматриваемых задачах весьма удобно по двум причинам.

а) Не требуется находить векторы, выводимые из базиса при одновременном введении блока векторов в базис. Вектор

i -й группы может заменить в базисе векторы только той же i -й группы.

б) Если исходные оценки Δ_i^k векторов, одновременно вводимых в базис, отрицательны, то введение в базис каждого вектора из вводимого блока дает увеличение целевой функции задачи. Это говорит о том, что при блочном введении векторов в базис в данном случае сохраняются преимущества «индивидуального» введения.

Доказательство подпункта а) следует из свойства 1.1 S^k -задачи. Подпункт б) следует из свойства 2.5.

Кроме того, при блочном введении векторов в базис остаются в силе те алгоритмические особенности S^k -задач, о которых говорилось в предыдущем пункте. Алгоритм, описанный выше, в основном остается тем же, только выбор вводимых в базис векторов A_i^k проводится одновременно во всех m группах столбцов A_i^k , и введение этих векторов в базис также происходит одновременно.

3. Решение задачи конечным алгоритмом без преобразования матрицы. Недостатком алгоритмов, приведенных выше, является необходимость преобразования матрицы на каждой итерации. При использовании основного алгоритма симплекс-метода преобразуется вся матрица ограничений, при использовании модифицированного симплекс-метода преобразуется подматрица базисных векторов и тем самым на каждой итерации определяется обратная матрица базиса. Обратная матрица нужна в S^k -задачах только для определения строки оценок условий задачи

$$\Lambda = -\Delta_6 A_6,$$

где A_6 — базисная матрица; $-\Delta_6$ — вектор коэффициентов целевой функции задачи (2.8) — (2.10), соответствующих базисным переменным. Но вектор Λ можно определить и без вычисления обратной матрицы. Для этого достаточно решить систему линейных алгебраических уравнений

$$\Lambda A_6 = -\Delta_6. \quad (4.5)$$

Остановимся кратко на сильных и слабых сторонах использования конечных алгоритмов, не требующих преобразования исходной матрицы. Как уже указывалось в главе III, при преобразованиях матрицы происходит увеличение числа ненулевых элементов и накопление ошибок округ-

ления. Поэтому при достаточно большой размерности задачи использовать преобразование матрицы становится невозможно. Если же применять конечные алгоритмы, не требующие преобразования матрицы, то в случае решения общей задачи линейного программирования на каждой итерации придется решать по две системы линейных алгебраических уравнений: одну — для определения вектора Λ (и отсюда — вектора, вводимого в базис), другую — для вычисления коэффициентов выражения этого вектора через векторы базиса (и отсюда — нахождения вектора, выводимого из базиса). В S^k -задачах использование конечных алгоритмов, не требующих преобразования матрицы, более эффективно по следующим причинам:

а) на каждой итерации нужно решать только одну систему — для определения Λ (векторы, выводимые из базиса, в данном случае находить не требуется);

б) в рассматриваемых S^k -задачах значительно уменьшается число итераций, необходимых для решения задачи, за счет того, что:

1) процесс ведется только в одну сторону (векторы, однажды выведенные из базиса, не могут быть введены обратно в базис);

2) векторы, имеющие коэффициенты $a_{rr}^k \geq a_{rr}^l$, сразу отбрасываются из рассмотрения;

3) введение векторов в базис происходит не по одному за итерацию, а блоками.

Приведем описание алгоритма. Записываем, как и ранее, матрицу $A = [a_{ij}]$ размера $n \times m$, элементы a_{ij}^k и коэффициенты целевой функции задачи (2.8) — (2.10). Пусть исходный базис в данной задаче будет составлен из единичных векторов при переменных y_j^0 . Проверяем оптимальность данного базиса ($\Delta_i^k \geq 0$), и если имеются отрицательные оценки, то находим в каждой i -й группе векторов A_i^k ($k=1, \dots, k_i$) столбец A_i^l с минимальной отрицательной оценкой Δ_i^l для введения в базис. Заменяем векторами A_i^l базисные векторы с теми же индексами i и формируем таким образом новую матрицу A_6 базисных векторов. Затем определяем строку Λ из решения системы (4.5) и находим оценки $\Delta_j^{k'}$, $\Delta_j^{l'}$ (4.2) — (4.4) внебазисных векторов относительно нового базиса. Вновь проверяем оптимальность. Если имеются отрицательные оценки $\Delta_j^{k'}$, то отбрасываем из рассмотрения «лишние» векторы (т. е. векторы, выведенные из базиса, и векторы с

коэффициентами $a_{rr}^k \geq a_{rr}^l$), и приступаем к следующей итерации — выбираем векторы, вводимые в базис, формируем матрицу A_6 и т. д. Заметим, что если при решении системы (4.5) будет получен вектор Λ , имеющий отрицательные компоненты, то это будет означать неограниченность целевой функции задачи на рассматриваемом множестве (доказательство аналогично приведенному в § 1 главы III). Неограниченность будет также в том случае, если на некоторой итерации будут получены отрицательные значения оценок векторов, не имеющих положительных элементов. Это прежде



Рис. 4.5. Блок-схема процедуры решения задачи (2.8) — (2.10) конечным алгоритмом, не требующим преобразования матрицы.

всего касается оценок Δ_j^l . Заметим также, что количественные значения переменных в оптимальной точке определяют-ся из решения системы

$$A_6^{\text{опт}} Y = C. \quad (4.6)$$

Блок-схема алгоритма приведена на рис. 4.5.

4. Итерационный алгоритм. В алгоритме, блок-схема которого приведена на рис. 4.5, на каждой итерации требуется решить систему линейных алгебраических уравнений (4.5). Причем, система решается в конечном счете для того, чтобы определить имеются ли векторы, более предпочтительные для введения в базис (в отношении величины целевой функции), чем те, из которых составлена базисная матрица A_6 данной итерации. Возникает мысль, что систему (4.5) можно решать не до конца, а только до выявления более предпочтительных для введения в базис векторов. Как только появляются подобные векторы A_i^l , нужно заменить ими базисные векторы, имеющие те же индексы i , и продолжать решение системы уже для измененной матрицы A_6 . Таким образом, замена базисных векторов будет происходить непосредственно в процессе решения системы (4.5). Процесс будет заканчиваться, когда система (4.5) решается до конца и не имеется векторов «более предпочтительных» для введения в базис.

Подобное построение алгоритма (как и при решении S -задач в главе III) возможно, если для решения системы (4.5) используется метод, обеспечивающий монотонное возрастание Λ , причем данный алгоритм, при котором не требуется решать систему до конца, будет, как и в главе III, иметь наибольшие преимущества в случае применения для решения системы (4.5) итерационного метода. Это вполне понятно, так как итерационный метод отличается тем, что обычно позволяет за относительно небольшое число шагов получить неплохое приближение к решению системы. Приближенное решение уже может использоваться для выявления векторов, имеющих отрицательные оценки Δ_i^k , затем векторы базиса заменяются, и решение системы продолжается. Таким образом, итерационный алгоритм решения задачи (2.8) — (2.10) заключается в решении системы (4.5) итерационным методом с заменой базисных векторов непосредственно в процессе решения системы.

Указанный итерационный алгоритм представляет собой обобщение известных итерационных методов решения систем линейных алгебраических уравнений на случай задач линейного программирования рассматриваемого класса и сохраняет все преимущества данных итерационных методов (матрица при решении не преобразуется, алгоритм прост, сглаживаются случайные ошибки вычислений и т. д.). С дру-

гой стороны, итерационный алгоритм решения рассматриваемой задачи сохраняет и все недостатки обычных итерационных процедур (ограниченная область применения, медленная скорость сходимости к решению для некоторых задач). Заметим, однако, что в сравнении с итерационными методами, используемыми в настоящее время для решения общей задачи линейного программирования (например, методами, связанными с бесконечными итерационными процессами решения матричных игр), указанные итерационные методы решения систем линейных уравнений являются, бесспорно, более отработанными и более эффективными в тех задачах, где они применимы. В рассматриваемых S^k -задачах могут использоваться следующие условия сходимости итерационных методов для $a_{ii}^k > 0$

$$|a_{ii}^k| > \left| \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \right|, \quad i = 1, \dots, m, \quad k = 1, \dots, k_i. \quad (4.7)$$

Столбцы, имеющие элементы $a_{ii}^k \leq 0$, не будут входить в матрицу $\tilde{A}_6$, так как это означает неограниченность целевой функции данной задачи.

Приступим к рассмотрению процедуры решения задачи итерационным методом. В оптимальной точке задачи (2.8) — (2.10) оценки всех векторов условий задачи должны быть неотрицательны:

$$\Delta_i^{k'} = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j + a_{ii}^k \lambda_i + \Delta_i^k \geq 0, \\ i = 1, \dots, m, \quad k = 1, \dots, k_i, \quad (4.8)$$

$$\Delta_i^{\pi'} = - \lambda_i + \Delta_i^{\pi} \geq 0, \quad i = j = 1, \dots, n, \quad (4.9)$$

$$\Delta_i^{0'} = \lambda_i \geq 0, \quad i = j = 1, \dots, n, \quad (4.10)$$

где λ_j определяются из решения системы (4.5), составленной из n условий (4.8) — (4.10), обращающихся в оптимальной точке в равенства (мы говорим n условий, предполагая, что ограничения задачи линейно независимы; в противном случае следует говорить «не более n »).

Определение оптимального базиса задачи (2.8) — (2.10) заключается именно в выделении из условий (4.8) — (4.10) n условий-равенств (составляющих систему (4.5)) таким

образом, чтобы выполнялись все остальные условия (4.8) — (4.10). Условия (4.9) и те из условий (4.8), которые имеют элементы $a_{ii}^k \leq 0$, не могут входить в систему (4.5) (это случай неограниченности). Предположим, что остальные строки (4.8) удовлетворяют условиям сходимости итерационных методов (4.7). Предположим далее, что мы задались исходным значением вектора $\Lambda^{(0)} = 0$ и составили систему (4.5) из n условий-равенств (4.8) — (4.10) следующим образом: если при $\Lambda = \Lambda^{(0)} = 0$ для данного i имеются оценки $\Delta_i^{k'} < 0$, то в систему (4.5) входит строка с оценкой $\Delta_i^{l'} = \min_{\Delta_i^{k'} < 0} \Delta_i^{k'}$; если

же при данном Λ не имеется отрицательных оценок $\Delta_i^{k'}$ для некоторого i , то в систему (4.5) входит строка с оценкой $\Delta_i^{0'} = 0$. Запишем эту систему:

$$\left. \begin{aligned} \Delta_i^{l'} &= - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j + a_{ii}^l \lambda_i + \Delta_i^l = 0, & i \in M, \\ \Delta_i^{0'} &= \lambda_i = 0, & i \notin M, \end{aligned} \right\} \quad (4.11)$$

где $a_{ij}^l \geq 0$, $a_{ii} > 0$; M — множество индексов, для которых имеются отрицательные $\Delta_i^{k'}$ (при $\Lambda = 0$ $\Delta_i^{k'} = \Delta_i^k$);

$$\Delta_i^{l'} = \min_{\Delta_i^{k'} < 0} \Delta_i^{k'}. \quad (4.12)$$

Пусть система (4.11) решается методом итераций. Тогда на каждом шаге итерационного процесса будут получаться следующие значения переменных:

$$\left. \begin{aligned} \lambda_i^{(p)} &= \frac{1}{a_{ii}} \left(\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j^{(p-1)} - \Delta_i^l \right), & i \in M, \\ \lambda_i^{(p)} &= 0, & i \notin M, \end{aligned} \right\} \quad (4.13)$$

где $p-1$, p — номера шагов итерационного процесса; $\lambda_i^{(0)} = 0$. При каждом значении вектора $\Lambda^{(p)}$ мы имеем в системе (4.11)

$$\begin{aligned} \Delta_i^{l'}(\Lambda^{(p)}) &= - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j^{(p)} + a_{ii}^l \lambda_i^{(p)} + \Delta_i^l, & i \in M, \\ \Delta_i^{0'}(\Lambda^{(p)}) &= \lambda_i^{(p)}, & i \notin M, \end{aligned}$$

а в системе (4.8) —

$$\Delta_i^{k'}(\Lambda^{(p)}) = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j^{(p)} + a_{ii}^{(k)} \lambda_i^{(p)} + \Delta_i^k, \quad (4.14)$$

$$i = 1, \dots, m, \quad k = 1, \dots, k_i.$$

Отсюда

$$\left. \begin{aligned} \Delta_i^k(\Lambda^{(p)}) &= \Delta_i^{k'}(\Lambda^{(p)}) - \Delta_i^{l'}(\Lambda^{(p)}) = \\ &= (a_{ii}^k - a_{ii}^{l'}) \lambda_i^{(p)} + (\Delta_i^k - \Delta_i^{l'}), \quad i \in M, \quad k = 1, \dots, k_i, \\ \Delta_i^k(\Lambda^{(p)}) &= \Delta_i^{k'}(\Lambda^{(p)}) - \Delta_i^{q'}(\Lambda^{(p)}) = \\ &= - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j^{(p)} + \Delta_i^k, \quad i \notin M, \quad k = 1, \dots, k_i. \end{aligned} \right\} \quad (4.15)$$

Будем решать систему (4.11) не до конца, а только до того момента, пока не получим $\delta_i^k(\Lambda^{(p)}) < 0$, т. е. пока не получим значения вектора $\Lambda^{(p)}$, при котором некоторые условия (4.8) будут иметь значения $\Delta_i^{k'}(\Lambda^{(p)})$ меньшие, чем соответствующие значения $\Delta_i^{l'}(\Lambda^{(p)})$, $\Delta_i^{q'}$ условий (4.11) при тех же i . Если появляются такие $\Delta_i^{k'}(\Lambda^{(p)})$, то в соответствии с условием (4.12) заменяем i -ю строку системы (4.11) и продолжаем решение по формуле (4.13) от значения $\Lambda = \Lambda^{(p)}$. Когда система (4.11) решена до конца (т. е. получено Λ^* , при котором с достаточной точностью будут $\Delta_i^{l'}(\Lambda^*) = 0$) и при этом Λ^* не имеется элементов $\delta_i^k(\Lambda^{(p)}) < 0$, то задача решена.

В процессе решения задачи все время уменьшается число просматриваемых элементов $\Delta_i^{k'}(\Lambda^{(p)})$, так как на каждом шаге отбрасываются из рассмотрения условия (4.15), имеющие элементы $a_{ii}^k \geq a_{ii}^{l'}$ для $i \in M$. Замена строк в системе (4.11) при появлении $\delta_i^k(\Lambda^{(p)}) < 0$ сводится к изменению коэффициентов a_{ii}^k , Δ_i^l в соответствующих строках (4.13), если $i \in M$. Если же $\delta_i^k(\Lambda^{(p)}) < 0$ для $i \notin M$, то множество M расширяется и на $p+1$ -м шаге решения системы в (4.13) появляется дополнительная строка, соответствующая $\lambda_i^{(p+1)} \neq 0$.

В том случае, если для некоторого i величина $\delta_i^k(\Lambda^{(p)}) < 0$ одновременно для нескольких k , то в соответствии с (4.12) вводим в систему (4.11) ту i -ю строку, которая имеет

$$\min_{\delta_i^k < 0} \delta_i^k(\Lambda^{(p)}).$$

Если $\delta_i^k(\Lambda^{(p)}) < 0$ для строки, в которой $a_{ii}^k \leq 0$, то это означает неограниченность целевой функции задачи на рассматриваемом множестве. Неограниченность будет также в том случае, если некоторое

$$\Delta_i^{\pi'}(\Lambda^{(p)}) = -\lambda_i^{(p)} + \Delta_i^{\pi} < 0. \quad (4.16)$$

Таким образом, предлагаемый алгоритм заключается в проведении итерационного процесса (4.13), на каждом шаге которого a_{ii}^l , Δ_i^l берутся из строк (4.8), имеющих минимальное отрицательное значение $\Delta_i^{k'}(\Lambda^{(p)})$. Если же для некоторого i не имеется значений $\Delta_i^{k'}(\Lambda^{(p)}) < 0$, то соответствующие $\lambda_i^{(p)}$, $i \in M$ в (4.13) приравняются нулю. Строки, имеющие $a_{ii} > a_{ii}^l$, являются «лишними» и на каждом шаге отбрасываются из рассмотрения. Определение минимальных отрицательных оценок $\Delta_i^{k'}(\Lambda^{(p)})$ ведется по формулам (4.15.)

Блок-схема предложенного алгоритма приведена на рис. 4.6. Отметим, что мы до сих пор говорили только о качественном решении задачи (о выделении оптимального базиса). Для получения количественного решения достаточно решить тем же итерационным методом систему (4.6), где $A_6^{\text{спт}}$ в данном случае это матрица системы (4.11) на последней итерации качественного решения.

Перейдем к обоснованию предложенного алгоритма. Выше уже отмечалось, что для решения рассматриваемой задачи требуется выделить из неравенств (4.8) — (4.10) n условий-равенств (4.11), причем сделать это нужно таким образом, чтобы вектор Λ^* , обращающий эти условия в равенства, удовлетворял также и остальным условиям (4.8) — (4.10). В предложенном алгоритме система (4.11) на каждом p -м шаге процесса состоит из условий (4.8) — (4.10), имеющих минимальное значение $\Delta_i^{k'}(\Lambda^{(p)})$, $\Delta_i^{q'}(\Lambda^{(p)})$ для каждого i . Поэтому, если данная система решена до конца, то есть получено значение $\Lambda^{(p)} = \Lambda^*$, при котором $\Delta_i^{l'}(\Lambda^*) = 0$, то все остальные условия (4.8) — (4.10) выполняются:

$$\Delta_i^{k'}(\Lambda^*) \geq \Delta_i^{l'}(\Lambda^*) = 0.$$

Таким образом, при получении точного решения Λ^* системы (4.11) исходная задача будет решена. Требуется доказать только, что при выполнении условий сходимости (4.7) предлагаемая процедура сходится к точному решению системы (4.11). Для этого достаточно убедиться в монотонном

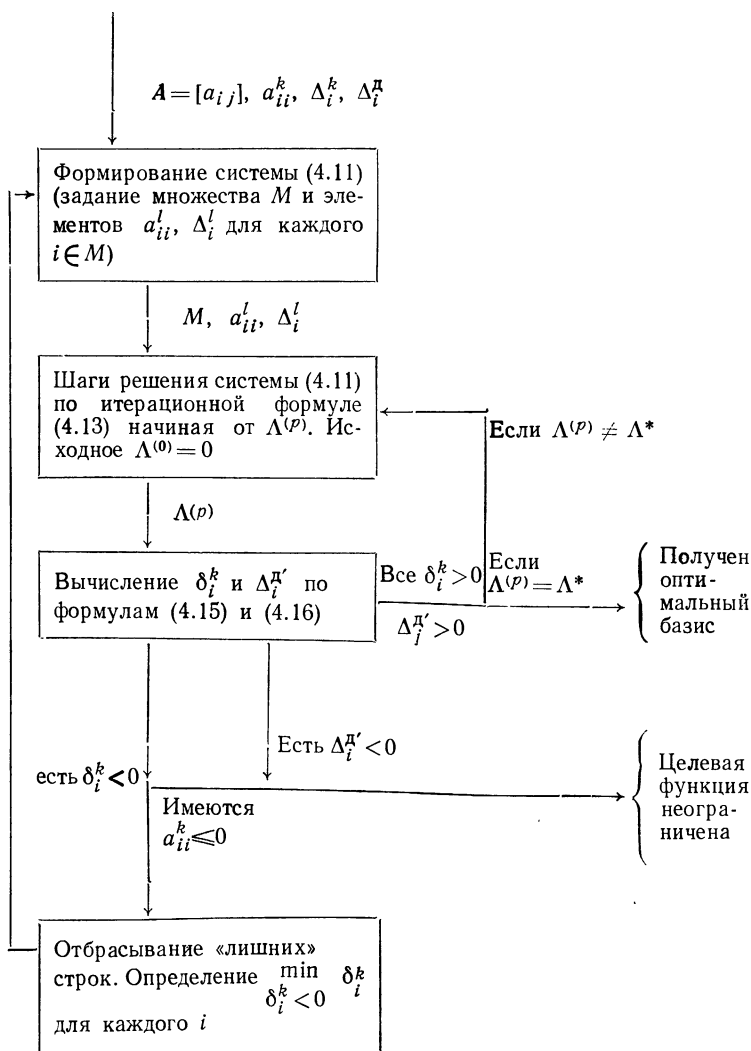


Рис. 4.6. Блок-схема процедуры решения задачи (2.8)—(2.10) итерационным алгоритмом

возрастании вектора $\Lambda^{(p)}$ на каждом шаге итерационного процесса. (Под «монотонным возрастанием» в данном случае понимается то, что на каждом шаге процедуры хотя бы одна компонента вектора $\Lambda^{(p)}$ возрастает, а все остальные компоненты не убывают *). Значения компонент вектора $\Lambda^{(p)}$ не могут возрастать до бесконечности, так как предлагаемая процедура применяется только при выполнении условий сходимости (4.7), обеспечивающих сходимость процесса к единственному решению системы (4.11), составленной из любых строк (4.8) — (4.10).

Скорость сходимости итерационного процесса в данном случае та же, что и в случае решения методом итераций систем линейных алгебраических уравнений. Правило остановки процесса также остается тем же: итерационный процесс (4.13) останавливается, как только на некотором шаге ни одна компонента вектора $\Lambda^{(p)}$ не изменяет своего значения в пределах заданной точности вычислений.

Таким образом, для обоснования алгоритма необходимо убедиться в монотонном возрастании вектора $\Lambda^{(p)}$ на шагах решения задачи. Кроме того, нужно обосновать процедуру определения неограниченности целевой функции и процедуру отбрасывания «лишних» условий на каждом шаге алгоритма.

Прежде всего — о монотонном возрастании. Докажем следующее свойство: *для монотонного возрастания вектора $\Lambda^{(p)}$ на шагах итерационного процесса (4.13) достаточно выбрать исходную точку $\Lambda^{(0)}$ таким образом, чтобы удовлетворялись условия*

$$-\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j^{(0)} + a_{ii}^l \lambda_i^{(0)} + \Delta_i^l \leq 0, \quad i \in M. \quad (4.17)$$

*) Здесь имеется определенная терминологическая трудность. С одной стороны, в данном случае нельзя пользоваться термином «неубывание», так как это означает, что $\Lambda^{(p+1)} \geq \Lambda^{(p)}$, а равенства в ходе процесса быть не может (как только появляется равенство, процесс останавливается). С другой стороны, нельзя говорить и «возрастание», если понимать под этим, что $\Lambda^{(p+1)} > \Lambda^{(p)}$ (так как не все компоненты вектора $\Lambda^{(p+1)}$ обязательно больше соответствующих компонент вектора $\Lambda^{(p)}$). Поэтому мы говорим «монотонное возрастание», но поясняем, что понимается под этим в данном случае. Это нам представляется здесь наиболее подходящим, так как подчеркивает главное: на каждом шаге процесса хотя бы одна компонента вектора $\Lambda^{(p)}$ возрастает.

Доказательство. Пусть имеется вектор $\Lambda^{(0)}$, удовлетворяющий условиям (4.17). Тогда,

$$\lambda_i^{(0)} \leq \frac{1}{a_{ii}^l} \left(\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j^{(0)} - \Delta_i^l \right), \quad i \in M.$$

В то же время из (4.13)

$$\lambda_i^{(1)} = \frac{1}{a_{ii}^l} \left(\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j^{(0)} - \Delta_i^l \right), \quad i \in M.$$

Из сравнения этих двух выражений имеем

$$\lambda_i^{(1)} \geq \lambda_i^{(0)}, \quad i \in M. \quad (4.18)$$

Условия (4.17) являются не только достаточными для выполнения (4.18), но и необходимыми. Действительно, рассматривая все пройденные соотношения в обратном порядке, мы убедимся, что если выполняются условия (4.18), то будут выполняться и неравенства (4.17).

Таким образом, пока мы подтвердили только то, что условия (4.17) являются необходимыми и достаточными для монотонного возрастания вектора Λ на первом шаге итерационного процесса. Заметим, что это имеет место при решении методом итераций любой системы линейных алгебраических уравнений, независимо от ее специфических свойств. Однако сохранение монотонного возрастания вектора Λ на последующих шагах итерационного процесса обуславливается спецификой рассматриваемой системы.

Опишем соотношение между векторами $\Lambda^{(p+1)}$ и $\Lambda^{(p)}$ для произвольного шага итерационного процесса (4.13). При этом будем отдельно рассматривать случай, когда коэффициенты a_{ii}^l , Δ_i^l в (4.13) остаются одни и те же на p -м и $(p+1)$ -м шаге, и случай, когда эти коэффициенты на $(p+1)$ -м шаге заменяются другими.

Начнем со случая, когда коэффициенты a_{ii}^l , Δ_i^l не меняются на $(p+1)$ -м шаге по сравнению с p -м. Тогда из (4.13)

$$\lambda_i^{(p+1)} - \lambda_i^{(p)} = \frac{1}{a_{ii}^l} \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} (\lambda_j^{(p)} - \lambda_j^{(p-1)}), \quad i \in M.$$

Отсюда видно, что если $\lambda_i^{(p)} \geq \lambda_i^{(p-1)}$, то $\lambda_i^{(p+1)} \geq \lambda_i^{(p)}$, так как $a_{ij} \geq 0$, $a_{ii}^i > 0$. Таким образом, в данном случае для монотонного возрастания вектора $\Lambda^{(p)}$ на шагах итерационного процесса (4.13) достаточно того, чтобы $\Lambda^{(p)}$ возрастало хотя бы на одном предыдущем шаге. С другой стороны, возрастание вектора Λ обеспечивается на первом же шаге процесса, если исходная точка $\Lambda^{(0)}$ удовлетворяет условию (4.17). Что и требовалось доказать.

Перейдем теперь к случаю, когда коэффициенты a_{ij}^i , Δ_i^i заменяются на $(p+1)$ -м шаге в соответствии с предложенным алгоритмом. Сделаем предварительно два замечания. Во-первых, по аналогии с (4.18), легко показать, что если $\Lambda^{(p)}$ удовлетворяет неравенствам

$$\Delta_i^{i'}(\Lambda^{(p)}) = - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \lambda_j^{(p)} + a_{ii}^i \lambda_i^{(p)} + \Delta_i^i \leq 0, \quad i \in M, \quad (4.19)$$

то

$$\lambda_i^{(p+1)} \geq \lambda_i^{(p)}, \quad i \in M. \quad (4.20)$$

И наоборот, если выполняются условия (4.20), значит, $\Lambda^{(p)}$ удовлетворяет неравенствам (4.19). Отсюда следует и второе замечание: если коэффициенты a_{ij}^i , Δ_i^i не меняются на шагах итерационного процесса (см. ранее рассмотренный случай), то

$$\Delta_i^{i'}(\Lambda^{(p)}) \leq 0, \quad i \in M \text{ для любого } p. \quad (4.21)$$

Предположим теперь, что в соответствии с описанным алгоритмом, мы на $(p+1)$ -м шаге вместо некоторой i -й строки системы (4.11) подставим строку, для которой

$$\Delta_i^{k'}(\Lambda^{(p)}) \leq \Delta_i^{i'}(\Lambda^{(p)}), \quad i \in M$$

или

$$\Delta_i^{k'}(\Lambda^{(p)}) \leq \Delta_i^{q'}(\Lambda^{(p)}), \quad i \notin M.$$

Тогда, учитывая (4.21), а также то, что $\Delta_i^{q'} = 0$, получим

$$\Delta_i^{k'}(\Lambda^{(p)}) \leq 0, \quad i \in M,$$

где множество M является расширенным за счет i -х строк, имеющих $\Delta_i^{k'}(\Lambda^{(p)}) \leq \Delta_i^{q'}(\Lambda^{(p)})$. А отсюда уже очевидно, что на основании (4.19) — (4.20) и в этом случае $\lambda_i^{(p+1)} \geq \lambda_i^{(p)}$. Монотонное возрастание вектора $\Lambda^{(p)}$ доказано для всех случаев.

Выбор исходной точки $\Lambda^{(0)}$, удовлетворяющей условиям (4.17), в данном случае очевиден:

$$\Lambda^{(0)} = 0.$$

(Условия (4.17) при этом $\Lambda^{(0)}$ удовлетворяются, так как система (4.11) по построению содержит все $\Delta_i^l < 0$.)

Перейдем теперь к обоснованию процедуры определения неограниченности. Докажем следующее свойство: *целевая функция задачи (2.8) — (2.10) является неограниченной на рассматриваемом множестве, если на некотором шаге итерационного процесса (4.13) появляется элемент $\Delta_i^{p'} < 0$ или $\delta_i^k < 0$ для строки (4.14), имеющей $a_{ii}^k \leq 0$.*

Доказательство. Запишем значение элемента $\Delta_i^{p'}$ в (4.9) при данном $\Lambda^{(p)}$:

$$\Delta_i^{p'}(\Lambda^{(p)}) = -\lambda_i^{(p)} + \Delta_i^{p'}.$$

Элемент $\lambda_i^{(p)}$ на шагах итерационного процесса может изменяться от нулевого значения только в сторону возрастания. Поэтому, если окажется, что $\Delta_i^{p'} < 0$ при данном $\lambda_i^{(p)}$, то отрицательность $\Delta_i^{p'}$ сохранится на всех последующих шагах итерационного процесса. Иными словами, данное условие (4.9) не может быть выполнено, система (4.8) — (4.10) несовместна. А так как условия (4.8) — (4.10) представляют собой не что иное, как ограничения задачи, двойственной к (2.8) — (2.10), то несовместность (4.8) — (4.10) означает неограниченность исходной задачи (2.8) — (2.10).

Совершенно аналогично можно убедиться в том, что если в (4.15) $\delta_i^k < 0$ при данном $\Lambda^{(p)}$, то δ_i^k остается отрицательным и на всех последующих шагах итерационного процесса. В то же время, если $\delta_i^k(\Lambda^{(p)}) < 0$, то $\Delta_i^{k'}(\Lambda^{(p)}) < \Delta_i^{l'}(\Lambda^{(p)}) \leq 0$. Отсюда — система (4.8) — (4.10) в данном случае несовместна, т. е. целевая функция задачи (2.8) — (2.10) неограничена на рассматриваемом множестве, что и требовалось доказать.

Остается обосновать только процедуру отбрасывания «лишних» условий на каждом шаге алгоритма. Отбрасывание «лишних» условий означает, что при использовании итерационного алгоритма (так же как и в конечных алгоритмах, описанных выше в пп. 1—3 данного параграфа) некоторая группа условий может вообще не рассматриваться. Кроме того, решение в данном случае ведется только в одну сторону, т. е. если некоторая l -я строка заменена в системе (4.11)

другой строкой, то эту l -ю строку не потребуется обратно вводить в систему (4.11), — в процессе решения ее можно не рассматривать.

Итак, докажем следующее свойство: *при решении задачи (2.8) — (2.10) итерационным алгоритмом (4.13) можно на каждом шаге процесса отбрасывать из рассмотрения строки (4.15), имеющие коэффициенты $a_{ii}^k \geq a_{ii}^l$ и строки, выведенные из системы (4.11).*

Доказательство. Предлагаемая процедура строится таким образом, что в систему (4.11) входят строки (4.14), имеющие $\min \Delta_i^{k'} \quad (\text{или } \min \delta_i^{k'}, \text{ если отсчет ведется относи-})$
 $\Delta_i^{k'} < 0 \quad \delta_i^{k'} < 0$

тельно $\Delta_i^{l'}$). Задача заключается именно в нахождении строк (4.14), входящих в систему (4.11) в оптимальной точке, т. е. на последнем шаге алгоритма. Если же оказывается, что данная строка, начиная с некоторого шага итерационного процесса, не может иметь отрицательные значения δ_i^k на последующих шагах, то данная строка не войдет в систему (4.11). Это ясно заранее, поэтому данная строка может больше не рассматриваться.

Пусть на p -м шаге итерационного процесса сформулирована система (4.11), состоящая из строк, имеющих $\min \delta_i^k (\Lambda^{(p)}).$ После этого для строк, не вошедших $\delta_i^k (\Lambda^{(p)}) < 0$

в систему (4.11), $\delta_i^k (\Lambda^{(p)})$ будут неотрицательными. Неотрицательность δ_i^k сохранится и на последующих шагах процесса, если $a_{ii}^k \geq a_{ii}^l$ ($i \in M$). Это очевидно из (4.15): $\lambda_i^{(p)}$ может только возрастать, поэтому, если $\delta_i^k (\Lambda^{(p)}) \geq 0$ на данном шаге ($i \in M$), то тем более $\delta_i^k (\Lambda^{(p)}) \geq 0$ на последующих шагах. Далее, пусть строки, имеющие $a_{ii}^k \geq a_{ii}^l$, отброшены. Тогда коэффициент a_{ii}^k строки, входящей в систему (4.11), является наибольшим для данного i ($i \in M$). Отсюда, если строка выводится из системы (4.11), то она также может быть отброшена.

Если из системы (4.11) выводится i -я строка при $i \notin M$, то она также больше не будет вводиться в (4.11) и может не рассматриваться. Это очевидно, так как тогда

$$\Delta_i^{q'} (\Lambda^{(p)}) = \lambda_i^{(p)} \geq 0$$

($\Delta_i^{q'}$ не может стать на последующих шагах отрицательным). Поэтому множество M в процессе решения задачи может только расширяться. Итак, свойство доказано полностью.

В заключение отметим следующее. Мы построили в данном подпункте алгоритм, в котором отыскание оптимального базиса происходит непосредственно в процессе решения системы линейных уравнений относительно Λ . Подобное построение алгоритма может быть осуществлено не только на основе метода итераций, но при использовании для решения любого другого метода, обеспечивающего монотонное возрастание Λ в процессе решения.

Например, в случае использования метода Зейделя, процедура (4.13) примет вид

$$\lambda_i^{(p)} = \begin{cases} \frac{1}{a_{ii}^l} \left(\sum_{j=1}^{i-1} a_{ij} \lambda_j^{(p)} + \sum_{j=i+1}^n a_{ij} \lambda_j^{(p-1)} - \Delta_i^l \right), & i \in M, \\ 0, & i \notin M, \end{cases}$$

где множество M и элементы a_{ii}^l , Δ_i^l выбираются, как и ранее, в соответствии со значением $\delta_i^k(\Lambda^{(p)})$ в (4.15).

Отметим также, что все описанные в данном параграфе алгоритмы, естественно, оказываются пригодными не только для решения задач, возникающих при статической оптимизации нелинейных многосвязных систем, но и в любых S^k -задачах, S -матрицы в которых отличаются только диагональными элементами.

§ 5. Алгоритмы решения S^k -задач. II

В данном параграфе мы рассмотрим общие алгоритмы решения S^k -задач. При этом в качестве иллюстрации будем отдельно рассматривать решение S^k -задач, возникающих в *марковском программировании*. Алгоритмы, описанные в предыдущем параграфе, без труда переформулируются на данные задачи. (Не удастся перенести только процедуру отбрасывания «лишних» условий.) Поэтому будем вести изложение в том же порядке, что и в предыдущем параграфе.

1. Использование симплекс-метода с учетом специфических свойств задачи (1.3) — (1.5). Особенности задачи (1.3) — (1.5), как мы уже отметили ранее, позволяют разбить матрицу ограничений на m блоков, и рассматривать каждый i -й блок векторов $A_i^k, k=1, \dots, k_i$ отдельно (так как известно, что из каждого i -го блока векторов A_i^k только один может войти в базис, причем заменить в базисе вектор A_i^k может только вектор той же i -й группы).

Подобная декомпозиция особенно полезна в том случае, если столбцы A_i^k заданы не явно, а выбираются из некоторого выпуклого многогранника (тогда $k \in K_i$). В задачах марковского программирования [18] последнее означает, что для каждого состояния задается не конечное множество, а некоторый выпуклый многогранник возможных стратегий; выбор стратегии на данном шаге процедуры оптимизации производится отдельно для каждого состояния.

Опишем алгоритм. Для простоты изложения совместим этап нахождения начального допустимого плана с этапом отыскания оптимального плана. Для этого достаточно ввести в задачу (1.3) — (1.5) искусственные переменные $y_j^0 \geq 0$ с достаточно большими по абсолютной величине отрицательными коэффициентами $-b_{mj}$ целевой функции $|b_{mj}| \gg \max_{i, k} (-b_i^k)$. Возьмем начальный опорный план в точке $y_j^0 = c_j$.

Тогда вектор Δ исходных оценок столбцов A_i^k записывается

$$\Delta = B - B_m \cdot A,$$

где $\Delta = [\Delta_i^k]$ — вектор оценок столбцов A_i^k задачи (1.3) — (1.5), $-B = [-b_i^k]$ — вектор коэффициентов целевой функции задачи (1.3) — (1.5), $A = [a_{ij}^k]$ — матрица ограничений задачи (1.3) — (1.5), $-B_m = [-b_{mj}]$ — вектор коэффициентов при искусственных переменных. Итак, вместо задачи (1.3) — (1.5) будем рассматривать следующую задачу:

$$\left. \begin{aligned} & - \sum_{i=1}^m \sum_{k \in K_i} \Delta_i^k y_i^k \rightarrow \max \\ \text{при} \quad & \sum_{k \in K_j} a_{ij}^k y_j^k - \sum_{i=1}^m \sum_{k \in K_i} a_{ij}^k y_i^k + y_j^0 = c_j, \quad j=1, \dots, n, \\ & y_i^k \geq 0, \quad y_j^0 \geq 0, \quad i=1, \dots, m, \\ & k \in K_i, \quad j=1, \dots, n. \end{aligned} \right\} \quad (5.1)$$

Здесь мы пишем K_i , предполагая, что если имеется конечное множество столбцов A , то $k=1, \dots, k_i$. В общем же случае $k \in K_i$. Совершенно аналогично записываются задачи марковского программирования (3.1), (3.3) при начальном опорном плане в точке $y_j^0 = c_j$.

Отыскание оптимального плана удобнее в данном случае вести модифицированным алгоритмом симплекс-метода, так

как число столбцов здесь значительно больше числа строк, а записывать столбцы приходится полностью (в отличие от задач, рассмотренных в предыдущем параграфе).

Процедура решения задачи, как и обычно, заключается (а) в отыскании вектора A_i^l с отрицательной оценкой Δ_i^l (или $\Delta_i^l = \min \Delta_i^k$, $\Delta_i^k < 0$) для введения в базис; (б) в переходе к новому базису. Затем вновь отыскивается вектор, имеющий оценку $\Delta_i^{l'} < 0$ и т. д. (где $\Delta_i^{l'}$ — оценка относительно измененного базиса). План оптимален, если не имеется отрицательных оценок векторов условий. Процедура (б) перехода от одного базиса к другому в данных задачах несколько упрощается только в том отношении, что не требуется находить вектор, выводимый из базиса — его номер определяется номером i вводимого вектора. В этой процедуре, как и обычно, используется не вся матрица ограничений, а только матрица, обратная к базисной.

Остановимся подробнее на процедуре (а) отыскания вектора, вводимого в базис [18]. Оценки Δ_i^k условий задачи (5.1), как известно, вычисляются в модифицированном алгоритме симплекс-метода следующим образом:

$$\Delta_i^{k'} = \Lambda A_i^k + \Delta_i^k, \quad (5.2)$$

где вектор Λ определяется вместе с вычислением обратной матрицы. Если множество K_i конечное (т. е. заранее заданы векторы A_i^k , $k=1, \dots, k_i$, $i=1, \dots, m$), то оценки $\Delta_i^{k'}$ вычисляются отдельно для каждого i последовательной подстановкой в (5.2) векторов A_i^k , $k=1, \dots, k_i$. Если же векторы A_i^k и оценки Δ_i^k выбираются из выпуклого многогранного множества

$$\varphi_i(\bar{A}_i^k) \geq 0, \quad (5.3)$$

где $\bar{A}_i^k = [A_i^k, \Delta_i^k]$, то непосредственное вычисление оценок $\Delta_i^{k'}$ для каждого $\bar{A}_i^k$ невозможно, так как число векторов $\bar{A}_i^k$ может быть бесконечным. Однако вычислять оценки $\Delta_i^{k'}$ для всех векторов $\bar{A}_i^k$ и не требуется. Достаточно знать только один вектор, обладающий минимальной оценкой $\Delta_i^{l'} = \min \Delta_i^{k'}$. Определение вектора, обладающего для каждого $\Delta_i^{k'} < 0$

го i минимальной оценкой $\Delta_i^{l'}$, производится с помощью решения вспомогательных задач линейного программирования

с целевой функцией

$$\Delta_i^{k'} = \Lambda A_i^k + \Delta_i^k \rightarrow \min, \quad (5.4)$$

где неизвестный вектор A_i^k выбирается из множества (5.3). Решив отдельные вспомогательные задачи (5.3) — (5.4) для всех i , мы либо убедимся в оптимальности данного базиса (если все $\min_k \Delta_i^{k'} \geq 0$, $i = 1, \dots, m$), либо будем вводить в базис вектор A_i^l , обладающий наименьшей отрицательной оценкой из всех i -х векторов с $\Delta_i^l < 0$. Задачи (5.3) — (5.4) можно решать не для всех i , а только до того момента, пока не появится $\Delta_i^{k'} < 0$. В этом случае, в базис будет вводиться первый полученный вектор, имеющий отрицательную оценку $\Delta_i^{k'} = \Delta_i^l < 0$.

Конечность процедуры доказывается из следующих соображений. Если число столбцов A_i^k конечно (множество $K_i = \{1, \dots, k_i\}$), то рассматриваемая задача является обычной задачей линейного программирования, а описанная процедура — применением симплекс-метода к данной задаче. Если же число столбцов A_i^k бесконечно (столбцы выбираются из области, ограниченной выпуклым многогранником), то для доказательства конечности процедуры необходимо уяснить следующее. Решение задачи линейного программирования (5.3) — (5.4) находится в крайних точках выпуклого многогранника (5.3), т. е. из бесконечного числа столбцов A_i^k для нас представляет интерес только конечное число столбцов, соответствующих крайним точкам многогранника (5.3). Если выписать все эти столбцы в матрицу задачи (5.1), то получится задача линейного программирования с конечным числом столбцов. Описанный алгоритм есть не что иное, как применение симплекс-метода к решению данной задачи. Конечность алгоритма очевидна.

Отметим, что описанная процедура остается той же не только при решении задачи S^k -задачи (5.1) или S^k -задачи (3.1) марковского программирования, но и при решении задачи (3.3), возникающей при оптимизации марковских процессов без переоценки. Нулевые значения правых частей ограничений задачи (3.3) в данном случае не вызывают затруднений при решении, так как качественное решение данных задач не зависит от правых частей ограничений.

2. Использование блочного введения векторов в базис. При решении задачи (5.1), так же как и в

задачах, рассмотренных в предыдущем параграфе, оказывается весьма эффективным использование блочного введения векторов в базис. Алгоритм симплекс-метода, описанный выше в п. 1, при одновременном введении векторов A_i^l в базис изменяется только в том отношении, что при переходе к новому базису в качестве разрешающего вместо элемента $a_{ii}^{l'}$ будет рассматриваться матрица $\tilde{A}$, составленная из i -х элементов векторов $A_i^{l'}$ (где $A_i^{l'}$ — вектор A_i^l , выраженный через векторы базиса). Процедура отыскания векторов, вводимых в базис, остается той же: векторы A_i^l , имеющие оценки $\Delta_i^{l'} = \min_k \Delta_i^{k'} < 0$, выбираются отдельно для каждого i , причем множество векторов A_i^k может, как и ранее, быть конечным, либо может задаваться выпуклым многогранником.

Для обоснования данного алгоритма необходимо убедиться в том, что целевая функция задачи на каждом шаге увеличивается. Покажем это. Пусть на данной итерации алгоритма векторы базиса составляют матрицу A_{61} , а на последующей итерации — базисную матрицу A_{62} . Значение целевой функции на данной итерации

$$L_1 = -\Delta_1 A_{61}^{-1} C,$$

на последующей итерации —

$$L_2 = -\Delta_2 A_{62}^{-1} C,$$

где Δ_1, Δ_2 — векторы, составленные из компонент Δ_i^k задачи (5.1), соответствующих базисным векторам данной и последующей итераций. Изменение целевой функции на данном шаге алгоритма

$$L_2 - L_1 = (-\Delta_2 A_{62}^{-1} + \Delta_1 A_{61}^{-1}) C.$$

Учитывая, что $-\Delta_6 A_6^{-1} = \Lambda_3$

$$L_2 - L_1 = (\Lambda_2 - \Lambda_1) C. \quad (5.5)$$

Отсюда для обоснования алгоритма необходимо показать, что $\Lambda_2 \geq \Lambda_1$, причем хотя бы для одной компоненты λ_i вектора Λ , $\lambda_{i2} > \lambda_{i1}$, если только данный шаг не является последним шагом процедуры.

Оценки $\Delta_i^{k'}$ (см. (5.2)) векторов A_i^k , вошедших в матрицу A_{62} , составляют вектор

$$\Delta'_2 = \Lambda_1 A_{62} + \Delta_2 \leq 0. \quad (5.6)$$

Причем, оценки $\Delta_i^{k'}$, вошедшие в Δ'_2 , равны нулю, если соответствующий вектор A_i^k являлся базисным на данной итерации (т. е. если A_i^k входил в матрицу A_{61} , а теперь входит в матрицу A_{62}). Если же вектор A_i^k матрицы A_{62} не входил ранее в матрицу A_{61} , то для него в (5.6) $\Delta_i^{k'} < 0$. Если $A_{62} = A_{61}$, значит, получен оптимум: $\Delta'_2 = 0$. Подставим в (5.6) значение Δ_2 , выраженное через вектор Λ_2 , полученный на следующей итерации $\Delta_2 = -\Lambda_2 A_{62}$:

$$\Delta'_2 = \Lambda_1 A_{62} - \Lambda_2 A_{62} = (\Lambda_1 - \Lambda_2) A_{62} \leq 0.$$

Отсюда

$$(\Lambda_1 - \Lambda_2) = \Delta'_2 A_{62}^{-1}. \quad (5.7)$$

Вектор $\Delta'_2 \leq 0$ по построению, причем хотя бы одна его компонента $\Delta_i^{k'} < 0$, если только данный шаг не является последним. Матрица A_{62}^{-1} является неотрицательной, если только целевая функция задачи ограничена на рассматриваемом множестве. Заметим также, что в случае ограниченности все элементы главной диагонали матрицы A_{62}^{-1} положительны. Эти факты следуют из того, что любая матрица A_6 , в которую может войти не более чем по одному вектору A_i^k для каждого i , является S -матрицей. Аналогичные доказательства для S -матриц приведены в § 1 главы III). Таким образом, из (5.7) следует, что

$$\Lambda_2 \geq \Lambda_1,$$

причем хотя бы для одной компоненты $\lambda_{i2} > \lambda_{i1}$, если только базис A_{61} не является оптимальным.

Вернемся теперь к (5.5). В задаче (5.1) вектор $C \geq 0$. Поэтому в (5.5) значение $(L_2 - L_1) > 0$, если компоненты вектора C ненулевые. Обоснование алгоритма для случая, когда компоненты вектора C нулевые, легко может быть построено по аналогии с обоснованием симплекс-метода для вырожденных задач. Отметим, однако, что даже в вырожденном случае зацикливание при решении рассматриваемых задач невозможно, так как определение оптимального базиса задачи не зависит от значений компонент правой части ограничений.

Процедура не изменится при решении S^k -задачи (3.3), в которой имеется одно дополнительное условие специального вида. Однако обоснование алгоритма будет в этом случае несколько иным.

3. Решение задачи конечным алгоритмом без преобразования матрицы. Решение задачи (5.1) конечным алгоритмом, не требующим преобразования матрицы, строится совершенно аналогично решению, рассмотренному в пункте 3 предыдущего параграфа. Отличие заключается только в том, что в задаче (5.1) не удастся отбрасывать «лишние» векторы в процессе решения. Иными словами, не удастся вести решение задачи (5.1) в одну сторону — вектор, однажды выведенный из базиса, может быть вновь введен в базис. Процедура решения, как и ранее, состоит в определении вектора Λ из решения системы (4.5), а затем в нахождении внебазисных векторов, имеющих наименьшую оценку Δ_i^k , отдельно для каждого i . Если множество векторов A_i^k конечно, то нахождение $\min_k \Delta_i^k$ производится

по формуле (5.2). Если же векторы A_i^k выбираются из выпуклого многогранника, то для этого решается задача (5.3) — (5.4). Блок-схема решения задачи приведена на рис. 4.7.

Отметим, что в принципе приведенный алгоритм пригоден как при единичном введении векторов в базис, так и при блочном введении. Однако особенно удобен он в случае блочного введения, так как в этом случае уменьшается число итераций, и соответственно меньшее число раз требуется решать систему линейных алгебраических уравнений (4.5).

Данный алгоритм, записанный в обозначениях задач (3.1) или (3.3), полностью соответствует методу Ховарда [87], предложенному из совершенно иных соображений для решения задач оптимизации марковских процессов с доходами. Покажем это. Начнем с задачи оптимизации процессов с переоценкой. Одна итерация метода Ховарда заключается при решении данных задач в следующем.

1. Для каждого i -го состояния марковского процесса выбирается некоторая k -я стратегия и тем самым задаются вероятности переходов p_{ij}^k и доходы q_i^k .

2. При заданных p_{ij}^k и q_i^k решается система линейных алгебраических уравнений относительно v_i :

$$v_i = q_i^k + \beta \sum_{j=1}^N p_{ij}^k v_j, \quad i = 1, \dots, N. \quad (5.8)$$

3. С помощью полученного вектора V для каждого i -го состояния выбирается новая стратегия k , которая максимизирует величину

$$q_i^k + \beta \sum_{j=1}^N p_{ij}^k v_j. \quad (5.9)$$

Затем мы возвращаемся к пункту 1 и формируем систему (5.8), используя для каждого i -го состояния улучшенные стратегии. Процедура заканчивается, когда на двух последовательных итерациях значения v_i при решении системы (5.8) не меняются.

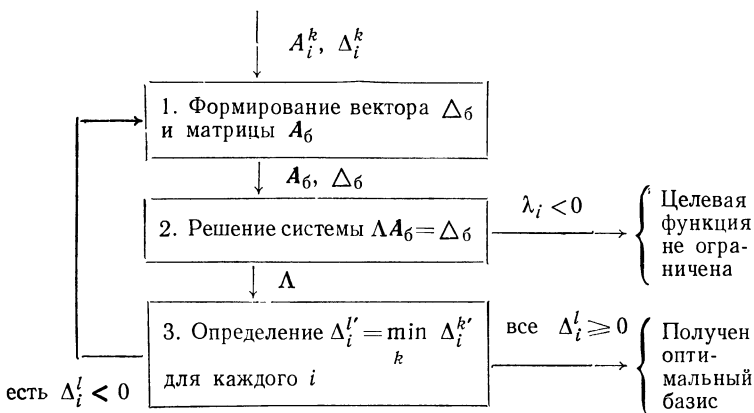


Рис. 4.7. Блок-схема процедуры решения задачи (5.1) конечным алгоритмом без преобразования матрицы.

Сравним теперь этот процесс с алгоритмом, блок-схема которого приведена на рис. 4.7. Задача оптимизации марковских процессов с доходами при наличии переоценки в терминах линейного программирования формулируется в виде S^k -задачи (3.1). Если записать решение этой задачи по алгоритму блок-схемы 4.7, то легко убедиться, что система (5.8) совпадает с системой (4.5), и соответственно совпадают подпункты 1 и 2 метода Ховарда и данного алгоритма. Соответствие в обозначениях в данном случае будет следующим: q_i^k в (3.1) соответствует $-\Delta_i^k$ в (5.1); $\beta p_{ij}^k \rightarrow a_{ij}^k$ ($j \neq i$); $(1 - \beta p_{ii}^k) \rightarrow a_{ii}^k$; $v_i \rightarrow \lambda_i$. Покажем теперь, что пункт 3 процедуры Ховарда также совпадает с пунктом 3 алгоритма, блок-схема которого приведена на рис. 4.7. Выражение (5.9) для

строки k , вошедшей в систему (5.8), записывается в виде

$$q_i^k + \beta \sum_{j=1}^N p_{ij}^k v_j = v_i.$$

Пусть для остальных строк при данном i

$$q_i^k + \beta \sum_{j=1}^N p_{ij}^k v_j = v_i^k. \quad (5.10)$$

В методе Ховарда при улучшении решения в систему (5.8) вводится строка, имеющая максимальное значение v_i^k , если это $v_i^k > v_i$. Иными словами, в систему (5.8) вводится строка, имеющая

$$\max_{(v_i^k - v_i) > 0} (v_i^k - v_i). \quad (5.11)$$

Обозначим $v_i - v_i^k$ через δ_i^k . Тогда выражение (5.11) примет вид

$$\min_{\delta_i^k < 0} \delta_i^k. \quad (5.12)$$

Остается показать, что $\delta_i^k = \Delta_i^{k'}$. А это немедленно следует из выражения (5.10), если учесть, что $v_i^k = v_i - \delta_i^k$:

$$\Delta_i^{k'} = -q_i^k - \beta \sum_{j=1}^N p_{ij}^k v_j + v_i = \delta_i^k.$$

Для строк k , вошедших в систему (5.8),

$$\Delta_i^{k'} = -q_i^k - \beta \sum_{j=1}^N p_{ij}^k v_j + v_i = 0.$$

Что и требовалось доказать.

Перейдем теперь к решению задач оптимизации марковских процессов без переоценки. В процедуре Ховарда в этом случае вместо системы (5.8) в пункте 2 решается система

$$g + v_i = q_i^k + \sum_{j=1}^N p_{ij}^k v_j, \quad i = 1, \dots, N, \quad (5.13)$$

а в пункте 3 для каждого i -го состояния выбирается новая стратегия k , максимизирующая величину

$$q_i^k + \sum_{j=1}^N p_{ij}^k v_j. \quad (5.14)$$

Сравним теперь процесс с алгоритмом, блок-схема которого приведена на рис. 4.7. Задача оптимизации марковских процессов с доходами без переоценки формулируется в виде задачи линейного программирования (3.3). Эта задача является S^k -задачей с одним дополнительным условием. Однако благодаря некоторым специфическим свойствам задачи (3.3), о которых говорилось выше в § 3, к решению данной задачи применимы алгоритмы решения S^k -задач. Если мы запишем систему (4.5), получаемую при решении данной задачи, то убедимся, что она совпадает с системой (5.13). В данном случае элементы v_i соответствуют оценкам λ_i первых N условий задачи (3.3), а элемент g соответствует оценке λ $(N+1)$ -го ограничения данной задачи. Подпункты 1 и 2 процедуры Ховарда и блок-схемы (4.7) совпадают.

Нетрудно показать, что третьи пункты этих алгоритмов также совпадают. Действительно, выражение (5.14) для строки, вошедшей в систему (5.13), имеет вид

$$q_i^k + \sum_{j=1}^N p_{ij}^k v_j = g + v_i,$$

а для остальных строк при данном i

$$q_i^k + \sum_{j=1}^N p_{ij}^k v_j = g + v_i^k. \quad (5.15)$$

При улучшении решения для каждого i в систему (5.13) по Ховарду должна вводиться строка, удовлетворяющая условию (5.11) или (5.12) (так как $(g + v_i^k) - (g + v_i) = v_i^k - v_i = -\delta_i^k$). Показать, что $\delta_i^k = \Delta_i^{k'}$ и в данном случае можно элементарно (учтя, что в (5.15) $v_i^k = v_i - \delta_i^k$)

$$\Delta_i^{k'} = -q_i^k - \sum_{j=1}^N p_{ij}^k v_j + v_i + g = \delta_i^k.$$

Для строк k , вошедших в систему (5.13),

$$\Delta_i^{k'} = -q_i^k - \sum_{j=1}^N p_{ij}^k v_j + v_i + g = 0.$$

Таким образом, мы убедились здесь в том, что итерационный метод Ховарда, предложенный в монографии [87] с совершенно иных позиций, является с точки зрения линейного

программирования симплекс-методом при блочном введении векторов в базис и использовании процедуры, не требующей определения обратной матрицы. Иными словами, с этой точки зрения процедура Ховарда является модификацией симплекс-метода, использующей специфические особенности S^k -задач, возникающих в марковском программировании. Заметим, что с позиций линейного программирования процедура Ховарда является не итерационной, а конечной.

Перейдем теперь к рассмотрению некоторых итерационных процедур решения S^k -задач.

4. Итерационный алгоритм. По аналогии с пунктом 4 предыдущего параграфа построим алгоритм, в котором система (4.5) будет решаться не до конца, а только до того момента, пока не появится вектор, более предпочтительный для введения в базис (т. е. соответственно, — для введения в систему (4.5)). Тем самым выделение оптимального базиса будет проводиться непосредственно при решении системы (4.5). При получении точного решения системы мы будем иметь и оптимальный базис. Как и в задачах предыдущего параграфа, подобное решение задачи (5.1) удобно проводить итерационным методом. Предлагаемая процедура, по сути дела, является обобщением на задачи линейного программирования рассматриваемого класса известных итерационных методов решения систем линейных алгебраических уравнений. Преимущества и недостаток данных итерационных процедур мы отмечали в пункте 4 предыдущего параграфа.

Наиболее простые условия сходимости итерационных методов для задачи (5.1) имеют вид для $a_{ii}^k > 0$

$$|a_{ii}^k| > \left| \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}^k \right|, \quad i = 1, \dots, m, \quad k = 1, \dots, k_i.$$

Алгоритм решения задачи (5.1) заключается в следующем.

1) Формируется система (4.5) из столбцов A_i^l , имеющих оценки $\Delta_i^{l'} = \min_k \Delta_i^{k'} < 0$ (на 1-й итерации $\Lambda = \Lambda^{(0)} = 0$ и $\Delta_i^{k'} = \Delta_i^k$) и столбцов A_i^0 , $i \in M$. По аналогии с (4.11)

$$\left. \begin{aligned} \Delta_i^{l'} &= - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}^l \lambda_j + a_{ii}^l \lambda_i + \Delta_i^l = 0, & i \in M, \\ \Delta_i^{0'} &= & \lambda_i = 0, & i \notin M. \end{aligned} \right\} \quad (5.16)$$

2) Производим несколько шагов решения системы (5.16) по итерационной формуле (аналогично (4.13)):

$$\lambda_i^{(p)} = \begin{cases} \frac{1}{a_{ii}^l} \left(\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}^l \lambda_j^{(p-1)} - \Delta_i^l \right), & i \in M, \\ 0, & i \notin M, \end{cases} \quad (5.17)$$

где $\lambda_i^{(0)} = 0$.

3) Вычисляем значения

$$\Delta_i^{m'} = \min_k \Delta_i^{k'} (\Lambda^{(p)}) \text{ при } \Lambda = \Lambda^{(p)},$$

и если появляются элементы $\Delta_i^{m'} < \Delta_i^{l'}$ или $\Delta_i^{m'} < \Delta_i^{o'}$, то возвращаемся к пункту 1 и заменяем i -е строки в системе (5.17). Если же все $\Delta_i^{m'} \geq \Delta_i^{l'}$, то продолжаем решение системы в п. 2 от $\Lambda^{(p)}$.

Если $\Delta_i^{m'} < \Delta_i^{l'}$ (или $\Delta_i^{m'} < \Delta_i^{o'}$) для вектора, имеющего $a_{ii}^m \leq 0$, то это будет означать неограниченность целевой функции. Это следует из (5.2) при монотонном возрастании $\Lambda^{(p)}$. Блок-схема алгоритма приведена на рис. 4.8.

Обоснование данного алгоритма в основном совпадает с обоснованием, приведенном в пункте 4 предыдущего параграфа, если не рассматривать этап «отбрасывания» лишних ограничений (в задаче (5.1) этого сделать не удастся), и если учесть некоторые отличия в формулировках задачи (2.8) — (2.10) и задачи (5.1) (в (5.1) для $i \neq j$ имеются элементы a_{ij}^k вместо элемента a_{ij} в (2.8) — (2.10); в (5.1) не имеется столбцов A_j^H и т. д.). Отметим, что подобное построение алгоритма может быть осуществлено не только на основе итерационной процедуры (5.17), которая аналогична процедуре решения системы линейных уравнений методом итерации, но и на основе любого другого метода решения систем линейных алгебраических уравнений, если этот метод обеспечивает монотонное возрастание вектора Λ в процессе решения.

Отметим также следующее. Мы не останавливались на вопросе выбора количества шагов решения системы (5.16), после которого следует переходить от пункта 2 данного алгоритма к пункту 3. Этот вопрос нужно решать в зависимости от сравнительной трудоемкости пп. 2 и 3. Если пункт 3 осуществляется легко (например, число столбцов A_i^k для

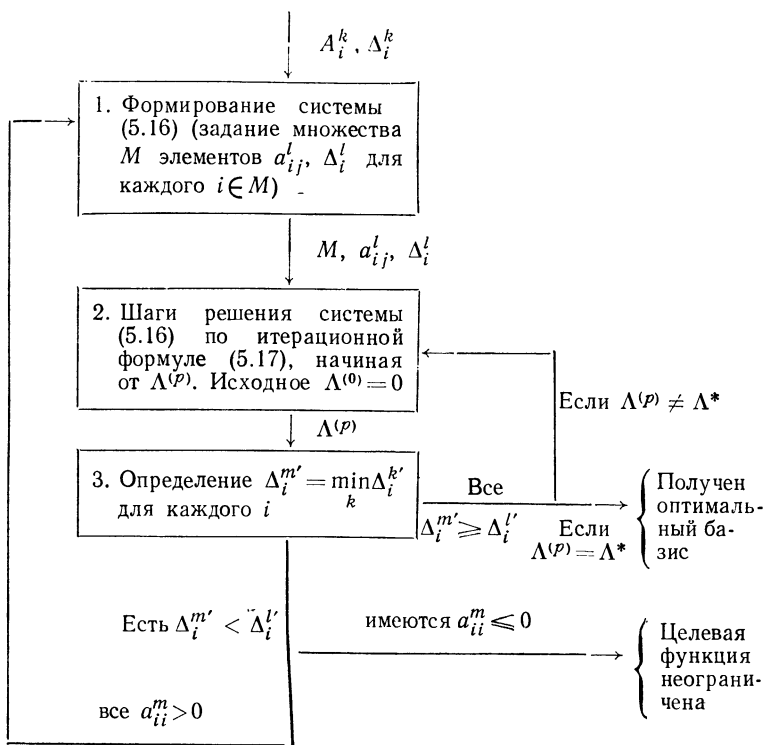


Рис. 4.8. Блок-схема процедуры решения задачи (5.1) итерационным алгоритмом.

каждого i небольшое), то проверку оценок $\Delta_i^{m'}$ следует осуществлять чаще. Если же трудоемкость пункта 3 высока, то эффективнее может оказаться проводить проверку $\Delta_i^{m'}$ через большее число шагов решения системы.

РЕШЕНИЕ НЕКОТОРЫХ ДИСКРЕТНЫХ ЗАДАЧ

Использование особенностей S -матриц позволяет построить эффективные алгоритмы решения не только непрерывных, но и некоторых дискретных задач оптимизации. В данной главе будут рассмотрены две дискретные задачи, возникающие в приложениях. В обеих задачах описание многосвязного объекта дается в виде системы линейных алгебраических уравнений, матрица которой является S -матрицей. Однако, кроме непрерывных переменных, в данной задаче имеются и дискретные. В одном случае наличие дискретных переменных вызвано возможностью снятия ограничений в зависимости от значений варьируемых параметров. В другом случае дискретные переменные непосредственно входят в число варьируемых параметров. Специфические свойства S -задач позволяют в первом случае свести решение исходной многоэкстремальной задачи к процедуре линейного программирования, а во втором случае дают возможность провести декомпозицию рассматриваемых частично целочисленных задач большой размерности, благодаря которой решение задачи значительно упрощается, а в некоторых важных на практике случаях может быть проведено даже вручную.

§ 1. Решение S -задач со снятием ограничений

1. Постановка задачи. В приложениях зачастую возникает такая ситуация, когда ограничения накладываются только при ненулевых значениях входных или выходных величин объекта, и снимаются при нулевых значениях переменных. Поясним это на примере задачи оптимизации нефтедобычи.

Основные технологические ограничения данной задачи имеют вид (см. в главе I)

$$\delta P \leq \delta P^{\text{пр.}}$$

Перейдя от депрессий δP к дебитам Q с учетом $\delta P = A^0 Q$, получим

$$A^0 Q \leq \delta P^{\text{пр}}, \quad (1.1)$$

где A^0 — матрица коэффициентов влияния по депрессиям. Каждое k -е условие системы (1.1) — это ограничение по давлению на забое k -й скважины. Если же k -я скважина оказывается в оптимальной точке выключенной, то в отношении наложения ограничений данная точка пласта (k -я скважина) ничем не отличается от бесконечного числа других точек пласта, на которых не пробурены скважины и соответственно не наложены ограничения, т. е. при $q_k = 0$ мы k -е условие типа (1.1)

$$\sum_{j=1}^n a_{ij}^0 q_j \leq \delta p_i^{\text{пр}}$$

можем снять. Это означает, что вместо условий (1.1) в задаче оптимального управления нефтедобывающими предприятиями следует записать условия

$$\gamma_i \left(\sum_{j=1}^n a_{ij} q_j \leq \delta p_i^{\text{пр}} \right), \quad i = 1, \dots, n, \quad (1.2)$$

где переменные γ_i являются целочисленными (булевыми):

$$\gamma_i = \begin{cases} 1, & \text{если } q_j > 0, \quad j = i, \\ 0, & \text{если } q_j = 0, \quad j = i. \end{cases}$$

Снятие k -го условия системы (1.1) при $\gamma_k = 0$ делает допустимой некоторую новую область пространства переменных q_j ($j = 1, \dots, n, j \neq k$), $q_k = 0$. Нетрудно показать, что полученное в итоге множество допустимых точек, задаваемое системой (1.2), в общем случае невыпукло, т. е. для решения задач оптимального управления нефтедобывающими предприятиями в общем случае оказывается неприменим аппарат линейного программирования. Данная задача относится к классу многоэкстремальных. Решение задачи, поставленной в терминах линейного программирования (без учета снятия ограничений, соответствующих выключенным скважинам), даст некоторый локальный оптимум, который может быть далек от глобального оптимума данной многоэкстремальной задачи.

Проиллюстрируем это графически на простейшем примере двумерной задачи. Пусть (рис. 5.1) прямая 1 ограничивает депрессию на забое 1-й скважины, а прямая 2 — депрессию на забое 2-й скважины. Оптимальное решение

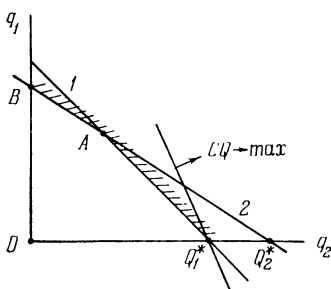


Рис. 5.1. Геометрическая иллюстрация многоэкстремальной задачи.

задачи линейного программирования лежит в точке $Q_1^* = \{0, q_2^*\}$. При выключении 1-й скважины поддерживать депрессию на ее забое нет смысла — ограничение прямой 1 можно откинуть, что сделает допустимой новую область — отрезок $Q_1^*Q_2^*$. Множество допустимых точек многогранника $OBAQ_1^*$ и отрезка $Q_1^*Q_2^*$ невыпукло. Учет многоэкстремальности позволит увеличить оптимальное значение целевой функции: решение многоэкстремальной задачи лежит в точке Q_2^* ($CQ_2^* > CQ_1^*$).

2. Решение задачи. Выведем условия, при которых глобальный оптимум сформулированной многоэкстремальной задачи совпадает с оптимумом задачи линейного программирования, получаемой при замене переменной q_i в условиях (1.2) постоянной величиной $q_i = 1, i = 1, \dots, n$. Очевидно, подобное совпадение имеет место в случае, когда в оптимальной точке условия (1.1), соответствующие $q_j = 0$ (т. е. соответствующие выключенным скважинам), будут выполняться как неравенства, так как тогда снятие данных ограничений не даст увеличения оптимума. Рассмотрим условия, при которых ограничения (1.1), соответствующие $q_j = 0$, будут в оптимальной точке задачи выполняться как неравенства. Переформулируем условия (1.1) через S -матрицу коэффициентов влияния по дебиту

$$A = (A^0)^{-1}$$

(см. в главе I) и получим

$$AX \leq B. \quad (1.3)$$

Вектор Q есть вектор дополнительных переменных, обращающих условия (1.3) в равенства, а вектор X — вектор дополнительных переменных для условий (1.1). Каждое i -е

неравенство системы (1.3) имеет вид

$$a_{ii}x_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j \leq b_i, \quad (1.4)$$

где $a_{ij} \geq 0$, $x_j \geq 0$, $i = 1, \dots, n$. Пусть $x_i = 0$. Тогда

$$a_{ii}x_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j < 0,$$

если имеются $a_{ij} > 0$ и X — ненулевой вектор. Отсюда, если $b_i > 0$, то условие (1.4) не может обратиться в равенство при $x_i = 0$. В общем случае (т. е. при любых $b_i \geq 0$, $a_{ij} \geq 0$) это следует из свойства 1.1 (см. § 1 главы I) при соответствующем изменении обозначений.

Итак, доказано, что если $b_i \geq 0$, то $q_i > 0$ при $x_i = 0$, и наоборот, $x_i > 0$ при $q_i = 0$. Таким образом, мы доказали, что если

$$(A^0)^{-1} = A, \quad (1.5)$$

где A — S -матрица, то достаточным условием совпадения локального оптимума задачи линейного программирования (при ограничениях (1.1)) с глобальным оптимумом многоэкстремальной задачи (при ограничениях (1.2)) является условие

$$B \geq 0, \quad (1.6)$$

где $B = (A^0)^{-1} \delta P^{\text{пр}}$. Геометрически результат прозрачен: при данных условиях области пространства переменных Q , дающие невыпуклость множества точек, задаваемого системой (1.2), являются недопустимыми (эти области отсекаются).

Полученные условия (1.5), (1.6) остаются справедливыми и если в задаче, кроме ограничений (1.2), имеются еще ограничения, так как введение дополнительных линейных ограничений может только сузить допустимую область. Иными словами, если множество допустимых точек, определяемое условиями (1.2), было выпукло без дополнительных ограничений.

Рассмотрим, в каких случаях будет выполняться неравенство (1.6) в задачах оптимизации нефтедобычи. Если вектор $\delta P^{\text{пр}} = [\delta p_i^{\text{пр}}]$ равен вектору значений реальных депрессий

при данном режиме работы пласта $\delta P^{\text{пр}} = \delta P_p$, то вектор $B = B_p$, где B_p — вектор реальных дебитов. Для эксплуатационных скважин в данном случае $b_i \geq 0$. Пусть допустимое множество точек расширено за счет увеличения правых частей:

$$\delta P^{\text{пр}} = \delta P_p + \Theta,$$

где $\Theta = [\theta_i]$ — вектор разности предельных и реальных значений депрессий ($\Theta \geq 0$). Тогда

$$(A^0)^{-1}(\delta P_p + \Theta) = (A^0)^{-1} \delta P_p + (A^0)^{-1} \Theta \geq 0$$

или

$$A\Theta \geq -B_p. \quad (1.7)$$

Таким образом, для выполнения условия $B \geq 0$ вектор Θ с неотрицательными компонентами θ_i должен удовлетворять системе неравенств (1.7). Если учесть, что диагональные элементы матрицы A в рассматриваемой задаче, как правило, больше по абсолютной величине суммы остальных элементов строки, т. е.

$$a_{ij} \Big|_{i=j} = a_i + \left| \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \right|, \quad i = 1, \dots, n,$$

то условия (1.7) примут вид

$$\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}(\theta_i - \theta_j) \geq -b_{pi} - a_i \theta_i, \quad i = 1, \dots, n. \quad (1.8)$$

Отсюда — выполнение условий $B \geq 0$ определяется не значениями θ_i , а их разбросом. Например, если брать значения компонент вектора Θ одинаковыми, то условие $B \geq 0$ будет всегда выполняться.

Из сказанного выше можно заключить, что условия применимости методов линейного программирования $B \geq 0$ для решения задач оптимального управления нефтедобывающими предприятиями весьма широки. Условия (1.8) обычно выполняются. Однако при постановке задач, предполагающих расширение допустимой области ограничений, например, задач планирования оптимальных организационно-технических мероприятий, обязательно следует учитывать, что задача является многоэкстремальной. Иначе можно по-

лучить абсурдное решение, например, потребуется увеличить глубину подвески насоса в скважине, которая должна быть остановлена, так как ограничение по этой скважине является существенным (обращается в оптимальной точке в равенство). Кроме того, учет многоэкстремальности позволит получить оптимальное значение целевой функции значительно большее, чем значение целевой функции в точке локального оптимума задачи, сформулированной в терминах линейного программирования.

Решение многоэкстремальной задачи при невыполнении условий (1.6) может быть достаточно просто осуществлено, когда удастся заменить невыпуклое множество точек, задаваемое системой (1.2), эквивалентным выпуклым допустимым множеством. Под эквивалентностью в данном случае понимается совпадение экстремума, полученного при решении задачи на упомянутых множествах.

Рассмотрим одну идею получения эквивалентного допустимого множества. Глобальный оптимум задачи (ограничения (1.2)) в невырожденном случае может находиться только в точке, в которой, если $q_j=0$, то $x_j \neq 0$ (если дебит равен нулю, то ограничение по данной скважине является несущественным — выполняется как неравенство). Отсюда эквивалентным допустимым множеством (в случае доказательства его выпуклости) можно считать множество, вершинами которого являются точки $\bar{X}$, обладающие тем свойством, что элементы q_j и x_j с одинаковыми индексами j не обращаются в нуль одновременно. Точки $\bar{X}$ всегда могут быть получены путем преобразования по рекуррентным формулам модифицированных жордановых исключений исходной матрицы A при разрешающих элементах, лежащих только на главной диагонали. Условие неотрицательности при этом накладывается только на вектор Q , так как в данном случае $x_j \neq 0$ может быть только при $q_j=0$, а тогда условие (1.2) отбрасывается ($y_i=0$); иными словами, x_j может быть любым: $x_j \geq 0$.

§ 2. Решение частично целочисленных S -задач

Рассмотрим задачу, в которой S -матрица $A=[a_{ij}]$ размера $m \times n$ дополнена некоторой подматрицей $D=[d_{ij}]$ размерности $m \times l$, а n -мерный вектор непрерывных переменных

$X=[x_j]$ дополнен R -мерным вектором целочисленных переменных $U=[u_j]$. Кроме этого, имеется группа ограничений, накладываемых только на целочисленные переменные. Задача имеет вид

$$L = CX + C_u U \rightarrow \max \quad (2.1)$$

при

$$AX - DU \leq B, \quad (2.2)$$

$$D_u U \leq B_u, \quad (2.3)$$

$$X \geq 0, \quad (2.4)$$

$$u_j = 0, 1, 2, \dots, K. \quad (2.5)$$

Матрица D предполагается неотрицательной — тогда матрица $[A - D]$ ограничений (2.2) является S -матрицей *). Вектор $B \geq 0$, как и обычно в S -задачах (см. формулировку S -задачи (0.1) — (0.3) в главе III). Ограничения, накладываемые только на целочисленные переменные, не обязательно должны иметь вид линейных условий (2.3). Эти ограничения могут быть произвольного вида. Подобные частично целочисленные задачи, возникающие на практике, обладают достаточно большой размерностью, и решение их известными методами весьма затруднительно.

Посмотрим, что дает в данном случае использование специфических особенностей S -задач. При фиксированных значениях переменных u_j задача (2.1) — (2.5) превращается в S -задачу линейного программирования. Предположим, что решение рассматриваемой задачи строится следующим образом: поиск оптимальных значений переменных u_j проводится комбинаторными методами (например, методом «ветвей и границ»), а в качестве оценки каждого фиксированного варианта выбора значений u_j берется оптимальное значение целевой функции соответствующей задаче линейного программирования.

Пусть, например, $U = U^l$, где U^l удовлетворяет условиям (2.3), (2.5). При данном варианте выбора значений u_j задача (2.1) — (2.5) превратится в следующую S -задачу линейного

*) Предлагаемый ниже алгоритм решения частично целочисленных S -задач может быть использован и в том случае, когда вместо условия $D \geq 0$ имеется более слабая предпосылка: матрица D удовлетворяет условию $B + D U \geq 0$ при любых $U \geq 0$.

программирования:

$$\left. \begin{aligned} CX &\rightarrow \max, \\ AX &\leq B + DU^i, \\ X &\geq 0. \end{aligned} \right\} \quad (2.6)$$

Оценка данного варианта выбора значений u_j :

$$L^i = \tilde{C}\tilde{A}(\tilde{B} + \tilde{D}U^i) + C_u U^i, \quad (2.7)$$

где $\tilde{C}$ — вектор, составленный из элементов c_j вектора C при $j \in M$ (M — множество индексов переменных x_j , вошедших в оптимальный базис S -задачи (2.6)); $\tilde{A}$ — матрица из элементов a_{ij} матрицы A при $i \in M$, $j \in M$; $\tilde{B}$ — вектор, составленный из элементов b_i вектора B при $i \in M$; $\tilde{D}$ — матрица, составленная из i -х строк матрицы D при $i \in M$. Матрицы и векторы $\tilde{A}$, $\tilde{D}$, $\tilde{C}$, $\tilde{B}$ соответствуют оптимальному базису S -задачи (2.6) (см. § 1 главы III). Если множество M состоит из r чисел, то размерность вектора-строки $\tilde{C}$ будет равна r , матрица $\tilde{A}$ будет размера $r \times r$, размер вектор-столбца $\tilde{B}$ равен r , а матрица $\tilde{D}$ будет иметь размер $r \times R$. Изменение значений вектора U вызывает изменение правой части задачи линейного программирования (2.6). В то же время одно из свойств S -задач заключается в том, что оптимальный базис не зависит от значений компонент вектора правой части ограничений (см. § 1 главы III). Это означает, что $\tilde{A}$, $\tilde{C}$, $\tilde{B}$, $\tilde{D}$ остаются одни и те же в оценке (2.7) для любого варианта выбора значений u_j . Таким образом, в рассматриваемой задаче появляется удивительная на наш взгляд возможность представления самой оценки варианта в виде функционала от варьируемых дискретных переменных. Иными словами, если обычно при использовании комбинаторных методов приходится оценивать каждый вариант (или группу вариантов) выбора значений дискретных переменных, то в данном случае оказывается возможным отыскивать сразу вариант, обладающий максимальной оценкой (2.7). Для этого достаточно решить задачу, в которой искомыми переменными являются только целочисленные переменные u_j :

$$\sigma U \rightarrow \max \quad (2.8)$$

при ограничениях (2.3), (2.5), где

$$\sigma = \tilde{C}\tilde{A}\tilde{D} + C_u. \quad (2.9)$$

Таким образом, предлагаемый алгоритм решения задачи будет заключаться в следующем.

а) Производим качественное решение S -задачи (2.6) с помощью алгоритмов, предложенных в главе III. Значение $U^l \geq 0$ при этом может быть любое, например, $U^l = 0$. После решения имеем множество M и соответственно матрицы $\tilde{A}$, $\tilde{D}$ и вектор $\tilde{C}$ для оптимального базиса.

б) Вычисляем вектор-строку σ по формуле (2.9) и решаем задачу (2.8), (2.3), (2.5). Получаем отсюда значения целочисленных переменных u_j в оптимальной точке.

в) Вычисляем значение $B + DU$ правой части задачи (2.6) при оптимальном векторе U и находим количественное решение задачи (2.6) из решения системы $\tilde{A}\tilde{X} = \tilde{B} + \tilde{D}U$. Получает отсюда значения непрерывных переменных x_j в оптимальной точке.

Итак, использование особенностей S -задач позволило в данном случае провести декомпозицию частично целочисленной задачи, т. е. позволило разложить задачу (2.1) — (2.5) с непрерывными и дискретными переменными на две подзадачи меньшей размерности: одну — относительно непрерывных переменных (подзадача (2.6)), другую — относительно дискретных переменных (подзадача (2.8), (2.3), (2.5)). Причем декомпозиция в данном случае не требует решения подзадач на каждом шаге процедуры (как это обычно имеет место при использовании разложения). Для решения достаточно решить подзадачи только по одному разу. Подзадача (2.6) является S -задачей и эффективно решается алгоритмами, предложенными ранее. Подзадача (2.8), (2.3), (2.5) — это задача целочисленного линейного программирования. Однако зачастую, в важных для практики случаях она решается исключительно просто, даже вручную. В следующем параграфе мы рассмотрим один из таких случаев.

§ 3. Пример частично целочисленной S -задачи]

Частично целочисленные S -задачи возникают в задачах оптимизации нефтедобычи, когда требуется определить не только оптимальные режимы скважин, но и оптимальный комплекс проводимых на промысле геолого-технических мероприятий. Дебиты скважин в этом случае являются непрерывными переменными, а величины, характеризующие

проведение того или иного геолого-технического мероприятия, являются дискретными переменными (в частности, булевыми).

Одной из важных задач выбора геолого-технических мероприятий является задача определения оптимального варианта перевода фонтанных скважин на механизированную добычу. Эта задача заключается в следующем. На определенном этапе разработки нефтяного месторождения возникает такая ситуация, что некоторые фонтанные скважины перестают фонтанировать или начинают давать незначительное количество жидкости, и требуется решить вопрос о том, какие скважины целесообразно перевести на механизированную эксплуатацию. Перевод скважин на механизированную добычу связан с дополнительными затратами на установку насосов и последующую их эксплуатацию. Ставится задача: определить, какие скважины следует перевести на механизированную добычу, и назначить их режимы таким образом, чтобы достигнуть максимума дебита чистой нефти по месторождению при наличии ограничений, определяемых предельными давлениями фонтанирования скважин, ограничений на затраты и других ограничений технологического, производственного или планового характера. Задача может ставиться также на минимум затрат при заданном плане отбора нефти с месторождения.

Математическую модель нефтяного пласта, работающего в жестком режиме при линейном законе фильтрации, записываем так же, как и в задачах управления дебитами скважин см. (1.24) в главе I)

$$A^0 Q = \delta P. \quad (3.1)$$

Допустимость использования подобной модели пласта в рассматриваемых задачах объясняется следующим. Перевод скважин на механизированную эксплуатацию обычно проводится в тот период разработки нефтяных месторождений, когда имеется еще достаточно различный контур разделения нефти и подпирающей жидкости. Поэтому, задаваясь рядом геологических ограничений, определяемых генеральным планом разработки месторождения, мы можем удовлетворить требования максимальной конечной нефтеотдачи пласта (эти требования — недопустимость прорыва воды и т. п.). Геологические ограничения формулируются в виде задания допустимой области изменения дебитов и депрессий для

скважин, близких к водо-нефтяному контакту. Таким образом, принятая модель позволяет решить задачу оптимизации в рамках, определяемых генеральным планом разработки пласта.

Итак, пусть мы имеем модель объекта в виде системы линейных алгебраических уравнений (3.1). Перейдем теперь к описанию ограничений рассматриваемой задачи. Если скважина эксплуатируется фонтанным способом, то депрессия (или забойное давление) по ней ограничивается условием устойчивого фонтанирования, которое в случае линейности записывается в виде

$$\delta p_i \leq \delta p_i^{\Phi},$$

где $\delta p_i^{\Phi} = p_k - p_i^{\Phi}$, p_i^{Φ} — предельное давление фонтанирования, p_k — контурное давление. Учитывая, что $\delta p_i = \sum_{j=1}^n a_{ij}^0 q_j$ (из 3.1)), ограничение депрессий по фонтанным скважинам имеет вид

$$\sum_{j=1}^n a_{ij}^0 q_j \leq \delta p_i^{\Phi} \quad (3.2)$$

Если скважина эксплуатируется механизированным способом, то ограничение депрессий по ней отличается обычно только значением правой части:

$$\sum_{j=1}^n a_{ij}^0 q_j \leq \delta p_i^M, \quad (3.3)$$

где $\delta p_i^M = p_k - p_i^M$, а p_i^M — предельно допустимое забойное давление для механизированных скважин. При переводе фонтанной скважины на механизированную добычу ограничение (3.2) снимается, а вместо него накладывается ограничение (3.3), причем установка насоса на фонтанной скважине позволяет увеличить депрессию по ней (и, соответственно, дебит), так как условие (3.3) шире условия (3.2): ($\delta p_i^M > \delta p_i^{\Phi}$). Обозначим через θ_i разность предельных значений депрессий в случае механизированной и фонтанной эксплуатации

$$\theta_i = \delta p_i^M - \delta p_i^{\Phi}.$$

Тогда ограничения (3.2) и (3.3) можно объединить:

$$\sum_i a_{ij}^0 q_j \leq \delta p_i^{\Phi} + \theta_i u_i, \quad (3.4)$$

где

$$u_i = \begin{cases} 0, & \text{если скважина эксплуатируется} \\ & \text{фонтанным способом;} \\ 1, & \text{если скважина механизирована.} \end{cases} \quad (3.5)$$

В том случае, если скважина была уже ранее механизирована, переменная $u_i = 1$ и условие (3.4) рассматриваемой задачи превращается в условие (3.3).

Выбор значений θ_i определяет максимально возможное изменение дебитов скважин, получаемое в результате перевода некоторых скважин на механизированную эксплуатацию. Причем, если дебит данной, ранее механизированной скважины сильно изменится при переводе некоторой другой скважины на механизированную добычу, то может потребоваться смена насоса и на данной скважине. Однако было бы нецелесообразно каждый раз при установке насоса на некоторой скважине менять насосы и на других скважинах, так как смена насоса — операция довольно трудоемкая. Поэтому мы заранее выбираем значения θ_i таким образом, чтобы при установке насосов изменения дебитов остальных скважин не выходили из определенных, допустимых пределов, при которых не будет требоваться смена насосов на всех скважинах. Задача определения значений максимально допустимых θ_i может быть легко сформулирована в виде самостоятельной задачи линейного программирования. Эта вспомогательная задача линейного программирования ставится для конечного варианта механизации нефтедобычи. Кроме ограничений на дебиты скважин, во вспомогательную задачу входят также ограничения на депрессии и различные геологические условия, о которых мы говорили выше. Мы не будем подробно останавливаться на рассмотрении задачи определения θ_i , чтобы не уводить изложение в сторону.

Продолжим описание ограничений задачи определения оптимального варианта установки насосов на фонтанных скважинах. В эту задачу остается включить только ограничения по затратам

$$\sum_{i=1}^n \varphi_i(u_i) \leq b, \quad (3.6)$$

где $\varphi_i(u_i)$ — затраты на установку насоса на i -й скважине и его эксплуатацию; b — предельно допустимые суммарные

затраты. На практике ограничение по затратам на перевод скважин чаще всего сводится к ограничению числа устанавливаемых насосов

$$\sum_{i=1}^n u_i \leq b_u, \quad (3.7)$$

где b_u — целое число. Будем рассматривать сначала задачу при ограничениях (3.7), а затем укажем, как изменится алгоритм решения, если вместо условия (3.7) будет условие более общего вида (3.6).

В качестве целевой функции рассматриваемой задачи, как принято в задачах оптимизации нефтедобычи, можно считать либо условие максимизации суммарного дебита чистой нефти по месторождению, либо условие минимизации затрат, необходимых для выполнения заданного плана добычи нефти. Будем для определенности считать, что задача ставится на максимум добычи чистой нефти по пласту. Целевая функция тогда

$$L = C^0 Q \rightarrow \max, \quad (3.8)$$

где $C^0 = [c_j^0]$ — вектор-строка коэффициентов нефтесодержания продукции скважин. Таким образом, задача определения оптимального варианта перевода фонтанных скважин на механизированную добычу сводится к задаче (3.4), (3.5), (3.7), (3.8).

Перейдем от этой задачи к S -задаче с частично целочисленными переменными. Для этого, по аналогии с задачей управления режимами скважин, будем вместо матрицы A^0 коэффициентов влияния скважин по депрессиям использовать S -матрицу A коэффициентов влияния по дебитам. Условия (3.4), записанные в виде равенств в матричной форме, будут иметь вид

$$A^0 Q + EX = \delta P^\Phi + \Theta U, \quad (3.9)$$

где E — единичная матрица, $U = [u_i]$, $Q = [q_j]$, $X = [x_j]$, $\delta P^\Phi = [\delta p_i^\Phi]$ — векторы-столбцы, Θ — диагональная матрица, коэффициенты главной диагонали которой равны θ_i . Умножим (3.9) слева на матрицу A . Тогда получим

$$EQ + AX = B + DU, \quad (3.10)$$

где $B = A \cdot \delta P \Phi$ — вектор дебитов скважин при предельных значениях депрессий до перевода на механизированную добычу; $D = A \cdot \Theta$ — матрица, характеризующая максимально возможное изменение дебитов скважин при изменении депрессий переводимых скважин на величину θ_i ; $(B + DU)$ — вектор дебитов скважин при данном U (т. е. при данном варианте перевода фонтанных скважин на механизированную добычу). Из процедуры определения значений θ_i , о которой говорилось выше, видно, что вектор $(B + DU)$ неотрицателен. Это означает, что рассматриваемая задача может быть решена с помощью алгоритма, предложенного в предыдущем параграфе. Запишем прежде всего задачу (3.4), (3.5), (3.7), (3.8) в виде, соответствующем частично целочисленной задаче (2.1) — (2.5):

$$L = CX + C_u U \rightarrow \max \quad (3.11)$$

при

$$AX - DU \leq B, \quad (3.12)$$

$$D_u U \leq b_u, \quad (3.13)$$

$$X \geq 0, \quad (3.14)$$

$$u_j = \begin{cases} 0, \\ 1. \end{cases} \quad (3.15)$$

Поясним запись (3.11) — (3.15). Целевая функция (3.11) получается при переходе от задачи (3.4), (3.5), (3.7), (3.8) к задаче, сформулированной через S -матрицу A . При этом переходе мы записываем условие (3.9) в виде (3.10). Если же подставить Q из (3.10) в целевую функцию (3.8), то получим

$$Q = -AX + B + DU,$$

$$L = CX + C_u U,$$

где

$$C = -C^0 A, \quad C_u = C^0 D = C^0 \cdot A \cdot \Theta.$$

В исходной точке задачи (3.11) — (3.15) значение целевой функции

$$L_{исх} = C^0 B = C^0 A \cdot \delta P \Phi.$$

Продолжим рассмотрение задачи (3.11) — (3.15). Условие (3.12) следует из (3.10), если считать вектор Q за вектор

добавочных переменных, обращающих условия (3.12) в равенства. Ограничение (3.13) — это ограничение по затратам. Данное условие соответствует (3.7), если D_u — вектор-строка, состоящая из единичных компонент.

В соответствии с алгоритмом предыдущего параграфа, решение данной задачи будет заключаться в следующем.

а) Считаем, что $U=0$ и находим качественное решение S -задачи:

$$\left. \begin{array}{l} CX \rightarrow \max \\ \text{при} \\ AX \leq B, X \geq 0. \end{array} \right\} \quad (3.16)$$

Задачу (3.16) решаем с помощью предложенных в главе III алгоритмов (см. решение подобной задачи в § 6 главы III). Заметим, что решение особенно упрощается, если все $c_j \leq 0$. В этом случае исходный базис является оптимальным: $X=0$, множество M — нулевое. Напомним, что в задачах оптимизации нефтедобычи множество M представляет собой множество номеров скважин, отключаемых (или переводимых на нижний предел по дебиту) при оптимизации. Случай $C \leq 0$ означает, что не имеется отключаемых скважин. Это происходит обычно при оптимизации пластов, в которых не имеется сильно обводненных скважин. А в рассматриваемых здесь задачах подобная ситуация часто будет иметь место, так как задачи определения оптимального варианта перевода скважин на механизированную добычу, как мы уже отмечали, решаются обычно на довольно раннем этапе разработки нефтяных месторождений.

б) Вычисляем вектор $\sigma = [\sigma_i]$ по формуле (2.9). Заметим, что при $C \leq 0$, $\sigma = C_u$ (так как при $M=0$, $\tilde{C} = \tilde{A} = \tilde{D} = 0$). Задача (2.8), (2.3), (2.5) имеет вид

$$\sum_{i=1}^n \sigma_i u_i \rightarrow \max \quad (3.17)$$

при ограничениях (3.7), (3.5).

Решение целочисленной задачи (3.17), (3.7), (3.5) записывается сразу, без проведения каких-либо вычислений. Покажем это. Прежде всего убедимся в том, что условия целочисленности (3.5) в данной задаче могут быть заменены линейными условиями:

$$0 \leq u_i \leq 1; \quad (3.18)$$

и решение при этом не изменится. Действительно, условия (3.18) в данной задаче эквивалентны условиям целочисленности (3.5), так как матрица ограничений (3.7), (3.18) является матрицей, в которой каждый столбец содержит два ненулевых элемента, равных (+1), и обладает свойством унимодулярности [43] (может быть разбита на два «правильных» класса, в один из которых входит условие (3.7), в другой — часть неравенств (3.18)):

$$u_i \leq 1.$$

Решение полученной задачи линейного программирования (3.17), (3.7), (3.18) производится элементарно: достаточно расположить переменные u_i в порядке, соответствующем уменьшению коэффициентов σ_i , и переводить переменные u_i в ненулевые ($u_i=1$) в этом порядке до тех пор, пока выполняется ограничение (3.7).

в) Находим количественное решение S -задачи (2.6) при оптимальном U , с помощью рекомендаций § 6 главы III. В этом случае, если $C \leq 0$, количественное решение будет иметь вид

$$X=0, \quad Q=B+DU.$$

Подведем некоторый итог. В качестве примера частично целочисленной задачи в данном параграфе была рассмотрена задача определения оптимального варианта перевода фонтанных скважин на механизированную эксплуатацию. Размерность этой задачи будет следующая: если рассматривается нефтяное месторождение, имеющее, например, 500 скважин, то число непрерывных переменных в данной задаче будет равно 500, кроме этого, в общем случае будет 500 целочисленных переменных. Число ограничений также будет не менее пятисот. На практике размерность задачи может быть и значительно выше. Решение частично целочисленных задач такой размерности на практике в настоящее время не представляется возможным. Однако использование специфических особенностей S -задач позволило внести в данном случае существенное упрощение. Исходная задача была разбита на две подзадачи, одна из которых (относительно непрерывных переменных) является S -задачей линейного программирования и достаточно эффективно решается в условиях большой размерности, а другая — относительно

дискретных переменных — решается сразу, без проведения вычислений.

Заметим, что если вместо ограничения (3.7) в задачу входит ограничение на затраты более общего вида — (3.6), то описанная процедура решения будет отличаться только тем, что в пункте (б) вместо задачи (3.17), (3.7), (3.5) будет решаться задача (3.17), (3.6), (3.5). Эта задача является обычной задачей распределения ресурсов и решение ее может быть легко осуществлено, например, методом динамического программирования [4].

РЕШЕНИЕ ЛИНЕЙНЫХ ЗАДАЧ ОПТИМИЗАЦИИ
НА ЭЛЕКТРИЧЕСКИХ МОДЕЛИРУЮЩИХ УСТРОЙСТВАХ

§ 1. Вступительные замечания

Вопросам решения задач математического программирования на различных аналоговых установках в литературе уделяется большое внимание. Рассматриваются два пути: 1) использование структурных моделей (т. е. АВМ общего назначения и специализированных АВМ); 2) использование моделей-аналогов.

Применение структурных моделей эффективно при решении задач математического программирования весьма небольшого размера. Поэтому здесь мы на данном вопросе не останавливаемся (см. [9,76]). Использование моделей-аналогов более перспективно в отношении решения задач большой размерности. Кроме того, модели-аналоги используются для получения математического описания многосвязных объектов, и решение задачи оптимизации непосредственно на этапе моделирования позволило бы снять многие проблемы. Настоящая глава посвящена решению линейных задач оптимизации многосвязных систем на электрических моделях-аналогах.

Модели для решения некоторых простых задач линейного программирования известны давно. Академик Л. В. Канторович еще в 1949 г. предложил гидравлическую модель для решения транспортной задачи. Однако впервые тесную связь между математическим программированием и теорией электрических цепей, содержащих источники тока, сопротивления и диоды, показал Дж. Деннис [35]. Развивая дальше идеи об аналогии данных задач оптимизации с электрическими цепями, в последнее время были созданы специализированные модели для решения некоторых задач математического программирования. В настоящее время имеется большое число работ [9, 13, 15, 30, 35, 72, 88], посвященных решению задач математического программирования на

различных моделях-аналогах. Но в данных работах предлагаются модели, специально предназначенные для решения задач математического программирования. Ниже мы покажем, что решение задач линейного программирования, возникающих при оптимизации многосвязных систем, можно провести на стандартных моделирующих устройствах непосредственно на этапе моделирования многосвязного объекта.

Поясним эту мысль. В общем случае для того, чтобы провести решение задачи оптимизации непосредственно на модели объекта, необходимо дополнить эту модель устройствами, моделирующими ограничения. После этого можно будет организовать процесс изменения параметров объекта таким образом, чтобы в рамках заданных ограничений достигался оптимум. Если же использовать специфические особенности задач линейного программирования, возникающих при оптимизации многосвязных объектов, то оказывается возможным решать данные задачи непосредственно на модели без дополнительных устройств, моделирующих ограничения.

В данной главе предлагаются два метода решения задачи. Поэтому изложение построено следующим образом: в § 2 и 4 даются описания первого и второго метода; в § 3 и 5 приводятся примеры решения задач, соответственно, по первому и второму методу; в заключение в § 6 проводится сравнение методов.

§ 2. Метод решения I

Имеем задачу

$$L = CX \rightarrow \max \quad (2.1)$$

при

$$AX = B', \quad (2.2)$$

$$B' \leq B, \quad (2.3)$$

$$X \geq 0, \quad (2.4)$$

где $A = [a_{ij}]$ — S -матрица, $B = [b_i]$ — вектор-столбец с неотрицательными компонентами. S -задача (2.1) — (2.4) получена после сведения к данному виду задач линейного программирования, возникающих при оптимизации многосвязных объектов (см. первую главу).

Заметим, что матрицу A в задаче (2.1) — (2.4) можно всегда считать квадратной, так как нас интересуют только те столб-

цы и строки матрицы A , которые содержат неотрицательные элементы. Число этих строк и столбцов в матрице A одинаково. Строки, все элементы в которых отрицательны, не могут стать в процессе решения данной задачи симплекс-методом разрешающими, т. е. эти строки «лишние». Столбцы, содержащие все отрицательные элементы, могут стать разрешающими, но в этом случае линейная форма задачи будет не ограничена на множестве допустимых решений. Проверить ограниченность линейной формы можно, если следить за знаком оценок Δ_j этих столбцов.

Покажем, что данные задачи можно достаточно просто решать на широко известных электрических моделях *) [29, 81], описываемых по 1-му закону Кирхгофа системой

$$AV = J, \quad (2.5)$$

где A — S -матрица, $V=[v_j]$ — вектор напряжений в узлах схемы, J — вектор токов, подаваемых в соответствующие узлы. Запишем условия (2.3) в виде равенств

$$AX + EX^0 = B, \quad (2.6)$$

где $X^0 \geq 0$. Решение задачи (2.1) — (2.4) алгоритмами, предложенными в третьей главе, заключается в определении, какая из двух переменных x_j, x_j^0 для каждого j (см. (2.6)) является ненулевой в оптимальной точке. В исходной точке все $x_j^0 > 0$ **) ($X^0 = B$), а все $x_j = 0$. Каждая итерация решения заключается в том, что одна компонента вектора X^0 обращается в нуль ($x_k^0 = 0$), а компонента вектора X с тем же индексом ($j=k$) становится отличной от нуля ($x_k > 0$). Подобная замена ненулевых переменных осуществляется, естественно, в тех случаях, когда это приводит к увеличению целевой функции (т. е. если $\delta L_k = C\delta X > 0$), причем, как показано в третьей главе, если переменную x_j целесообразно перевести в ненулевые ($\delta L_j > 0$), то эта переменная останется ненулевой и в оптимальной точке. Множество индексов ненулевых переменных x_j обозначается через M . Решение

*) Схемы некоторых стандартных электрических моделей, реализующих систему (2.5), будут приведены в следующем параграфе при рассмотрении примеров.

**) В более общей формулировке, включая вырожденный случай, следует вместо $x_j^0 > 0$ говорить «...переменная x_j^0 является базисной».

считается законченным, если при выделенном множестве M больше не имеется переменных x_j , для которых $\delta L_j > 0$.

Посмотрим, как можно осуществить этот процесс на модели (2.5). Удобнее всего поставить в соответствие вектору X^0 вектор задающих токов J в узлах. В этом случае компонентам вектора X будет соответствовать разница предельных и реальных значений напряжений в узлах. Покажем это. Пусть вектор задающих токов узлов

$$J = k_J X^0,$$

где k_J — коэффициент пропорциональности для токов. Если установить такие задающие токи на модели, то узловые напряжения из (2.5)

$$V = A^{-1}J = k_v A^{-1}X^0, \quad (2.7)$$

где k_v — коэффициент пропорциональности для напряжений. Запишем условия (2.6) относительно напряжений. Для этого умножим выражение (2.6) слева на матрицу A^{-1} . Имеем

$$EX + A^{-1}X^0 = A^{-1}B. \quad (2.8)$$

Выражение (2.8) можно записать в виде неравенства

$$A^{-1}X^0 \leq A^{-1}B.$$

Отсюда, учтя (2.7), получим

$$V \leq V_{\text{пр}},$$

где V — вектор реальных значений узловых напряжений, полученных при некоторых значениях задающих токов в узлах: $J = k_J X^0$, $V = k_v A^{-1}X^0$; $V_{\text{пр}}$ — вектор предельно допустимых значений узловых напряжений, полученных при токах $J = k_J B$, $V_{\text{пр}} = k_v A^{-1}B$. Итак, выражение (2.6) при данной аналогии между электрическими величинами и переменными задачами записывается относительно напряжений (см. (2.8)) следующим образом:

$$k_v X + V = V_{\text{пр}},$$

откуда

$$k_v X = V_{\text{пр}} - V, \quad (2.9)$$

что и требовалось доказать.

Заметим, что при такой электрической аналогии задача (2.1) — (2.4) имеет следующую физическую интерпретацию.

Задача заключается в выборе задающих токов и соответственно напряжений в узлах электрической цепи. Условия (2.2) — это уравнения, связывающие узловые напряжения и токи. Условия (2.3) ограничивают сверху изменение узловых напряжений; вектор B при этом является вектором задающих токов, при котором значения узловых напряжений будут предельными. Кроме этого, имеются условия неотрицательности.

Итак, мы приняли следующую аналогию между переменными задачи и электрическими величинами (см. (2.7) и (2.9)):

$$X^0 \sim J, \quad X \sim V_{\text{пр}} - V, \quad (2.10)$$

где $\sim$ — знак отношения эквивалентности.

Такая аналогия наиболее удобна для реализации на модели алгоритма, предложенного в главе III. Действительно, нам нужно, чтобы в процессе решения задачи все время одна из двух переменных x_j, x_j^0 для каждого j равнялась нулю, а другая была положительной. Принятая аналогия (2.10) позволяет это очень просто осуществить: если в j -й узел включен задающий ток ($x_j^0 > 0$), то для $x_j = 0$ достаточно иметь узловое напряжение в j -м узле равным предельному. Если же задающий ток в j -м узле отключен ($x_j^0 = 0$), то напряжение в узле падает, $v_j < v_{j\text{пр}}$ и соответственно $x_j > 0$.

Аналогию мы ввели. Теперь посмотрим, как реализовать алгоритм. В исходной точке все задающие токи ненулевые ($x_j^0 > 0$), причем токи выбраны таким образом ($X^0 = B, J = k_j B$), что значения узловых напряжений равны предельным значениям: $V = V_{\text{пр}}$. Поэтому все $x_j = 0$. На каждом шаге нам нужно обращать в нуль одну компоненту x_j^0 и переводить в ненулевые компоненту x_j с тем же индексом. При этом все остальные компоненты вектора X , не входящие в базис, остаются нулевыми. На модели для осуществления этого достаточно отключить задающий ток k -го узла, сохранив напряжения, подаваемые на все остальные узлы на прежнем уровне $v_j^{\text{пр}}$ ($x_j = 0$ для $j \notin M$, где M — множество индексов переменных x_j , входящих в базис; из (2.9) $v_j = v_{j\text{пр}}$ для $j \notin M$). Знак приращения целевой функции задачи ($\delta L = C \delta X$) при отключении задающего тока в данном узле определяет целесообразность отключения (т. е. определяет знак оценки Δ_j данного столбца матрицы A задачи (2.1) — (2.4)).

Таким образом, процедура решения задачи заключается в определении узлов, отключение задающего тока в которых дает положительное приращение целевой функции задачи ($\delta L > 0$).

Условие положительности разрешающего элемента, иначе, условие ограниченности линейной формы (2.1) на допустимом множестве (2.2) — (2.4) на данной итерации (при данном отключении) следует проверять, измеряя напряжение в узле после отключения задающего тока: если $v_k < v_k^{np}$, то после итерации $x_k > 0$, т. е. направляющий элемент положителен.

Отметим следующее:

а) Из структуры матрицы **A** следует, что условия (2.3) — (2.4) будут в процессе решения выполняться автоматически, т. е. не требуется никаких дополнительных устройств для моделирования ограничений (2.3) — (2.4). Это становится понятным из следующих соображений. Пусть имеем ограничения (2.3) в виде (2.6). Исходный опорный план: $X^0 = B$, $B \geq 0$. Структура матрицы **A** обеспечивает то, что при переходе к другим опорным планам при решении задачи симплекс-методом приращения элементов b_i будут неотрицательными, т. е. компоненты векторов X , X^0 не могут в процессе решения стать отрицательными, условия (2.3), (2.4) выполняются, что и требовалось доказать. Это ясно также из свойства 4 (см. § 1 главы III).

б) Если ток данного узла целесообразно отключить ($\delta L > 0$), то этот ток останется равным нулю и в оптимальной точке. Это вытекает из свойства 3 (см. § 1 главы III).

в) Варьирование производится только отключением токов. Задающие напряжения, как уже отмечалось выше, в узлах, задающий ток в которых не равен нулю, сохраняются постоянными.

Таким образом, предложенный подход, использующий специальные свойства *S*-задачи, позволил построить легко осуществимый процесс решения данной задачи на сеточной модели или на электрической цепи, описываемой системой $AV = J$. Ход решения задачи на электрической модели предложенным методом будет заключаться в следующем.

1) Устанавливаем значения задающих токов в узлах пропорциональными значениями компонент вектора B :

$$J = k_j X^0,$$

где $X^0 = V$ в исходной точке. При таких токах значения узловых напряжений будут равны предельно допустимым: $V = V_{\text{пр}}$, и соответственно $X = 0$ в исходной точке.

2) Отключаем задающий ток j -го узла, сохраняя узловые напряжения во всех остальных узлах на прежнем уровне (модель работает в режиме заданных напряжений). Вычисляем изменение целевой функции, вызванное подобным отключением:

$$\delta L_j = C \cdot \delta X = \sum_{j \in M} c_j \cdot \delta x_j$$

(напомним, что $\delta x_j = 0$ для $j \notin M$, где M — множество индексов узлов, в которых отключен задающий ток). Если $\delta L_j > 0$, то задающий ток j -го узла оставляем отключенным уже до конца решения задачи. Если же $\delta L_j \leq 0$, то вновь включаем задающий ток j -го узла на прежний уровень.

Переходим к следующему узлу и проделываем ту же операцию отключения задающего тока. И так далее, для $j = 1, 2, \dots, n$. Дойдя до последнего узла, вновь возвращаемся к тем узлам, в которых задающий ток остался включенным, и поочередно проверяем целесообразность отключения $\delta L \geq 0$ тока в них. Решение считается законченным, если $\delta L_j \leq 0$ для $j \notin M$. Иными словами, решение окончено, если, просмотрев все узлы, задающий ток в которых остался включенным, мы не обнаруживаем целесообразности отключения задающего тока ($\delta L \leq 0$).

Как показало решение практических задач, для реализации процедуры оказывается вполне достаточным совершить «обход» узлов от одного до трех раз максимум.

В результате решения будет получено итоговое множество номеров отключенных узлов и оптимальный базис (качественное решение) задачи будет: $x_j > 0$ для $j \in M$; $x_j = 0$ для $j \notin M$; $x_j^0 > 0$ для $j \notin M$; $x_j^0 = 0$ для $j \in M$. Количественные значения ненулевых переменных замеряются как токи $J_j = k_j x_j^0$ для узлов $j \notin M$ и как разность предельных и реальных узловых напряжений $v_{j\text{пр}} - v_j = k_v x_j$ для узлов $j \in M$.

Следует сказать несколько слов о вычислении оценок δL . Если вычислять данные оценки прямо как произведение векторов C и δX ,

$$\delta L = \sum_{j \in M} c_j \delta x_j, \quad (2.11)$$

то нужно на каждом p -м шаге решения помнить значения

узловых напряжений, которые были на предыдущем шаге в узлах с отключенным током:

$$\begin{aligned}\delta x_j &= x_j^{(p)} - x_j^{(p-1)} = \frac{1}{k_v} [(v_{jnp} - v_j^{(p)}) - (v_{jnp} - v_j^{(p-1)})] = \\ &= \frac{1}{k_v} (v_j^{(p-1)} - v_j^{(p)}), \quad j \in M. \quad (2.12)\end{aligned}$$

(Для узла, который отключен на данном p -м шаге, значение $v_j^{(p-1)}$ равно предельному.) Но запоминать на каждой скважине узловые напряжения $v_j^{(p-1)}$, как правило, неудобно. Гораздо удобнее вычислять значения целевой функции на p -м шаге:

$$L^{(p)} = \sum_{j \in M} c_j x_j^{(p)}, \quad (2.13)$$

где

$$x_j^{(p)} = \frac{1}{k_v} (v_{jnp} - v_j^{(p)}).$$

Тогда требуется запоминать только одну величину — максимальное значение линейной формы L^* , полученное до k -го шага. Если $L^{(p)}$ оказывается больше, чем L^* , то это будет означать, что оценка $\delta L > 0$. Если же $L^{(p)} \leq L^*$, значит, $\delta L \leq 0$:

$$\delta L = L^{(p)} - L^*. \quad (2.14)$$

На первом шаге $L_{\text{исх}}^* = \sum_{j \in M} c_j x_j^{(p)} = 0$, так как в исходной

точке множество M пустое.

Заметим, что в некоторых S -задачах в целевой функции вместо вектора C при переменных X может быть задан вектор C^0 при переменных X^0 . (Например, это имеет место в задачах оптимизации нефтедобычи.) Решать такие задачи на модели можно с помощью описанной выше процедуры, но для вычисления оценок удобно использовать вектор C^0 , иначе пришлось бы предварительно вычислить вектор $C = -C^0 A$ (исходный единичный базис $X^0 = B$, поэтому исходные оценки векторов при переменных X равны $\Delta = -C = -C^0 A$). В данном случае оценки δL следует вычислять следующим образом:

$$\delta L^{(p)} = \sum_{j \in M} c_j^0 \cdot \delta x_j^{(p)}, \quad (2.15)$$

где

$$\delta x_j^{(p)} = \frac{1}{k_j} (J_j^{(p)} - J_j^{(p-1)}), \quad j \notin M, \quad (2.16)$$

а $J_j^{(p)}$, $J_j^{(p-1)}$ — значения задающих токов в j -м узле на p -м и $(p-1)$ -м шаге. Чтобы не запоминать на данном шаге значения задающих токов предыдущего шага, можно брать, как и ранее, $\delta L^{(p)}$ из формулы (2.14) и вычислять $L^{(p)}$ аналогично (2.13):

$$L^{(p)} = \sum_{j \in M} c_j^0 x_j^{(p)}, \quad (2.17)$$

где

$$x_j^{(p)} = \frac{1}{k_j} J_j^{(p)}.$$

На первом шаге

$$L_{исх}^* = \sum_{j=1}^n c_j^0 x_{jисх}^0,$$

где

$$x_{jисх}^0 = \frac{1}{k_j} J_{jисх} = b_i, \quad i = j.$$

Выражения (2.11) — (2.14) и (2.15) — (2.17) позволяют получить ясное физическое представление о ходе процесса решения задачи. На каждой итерации при отключении задающего тока в некотором k -м узле будут увеличиваться токи в остальных узлах с индексами $j \notin M$. Это нужно для сохранения на прежнем уровне узловых напряжений в данных узлах, т. е. мы имеем отрицательное приращение тока в k -м узле и положительное приращение токов в остальных узлах. Выражения (2.16), (2.15), (2.17) показывают нам, что дает подобное изменение тока в k -м узле (т. е. изменение переменной x_k^0) как в отношении остальных токов узлов (остальных переменных), так и в отношении целевой функции. Это часто бывает очень важно при исследовании задачи, например, при рассмотрении различных вариантов изменения исходных данных (расширение правых частей ограничений и т. п.). Аналогичное замечание можно сделать и относительно выражений (2.12), (2.11). Отключение задающего тока на некотором k -м узле вызовет отрицательное приращение узлового напряжения на этом узле и отрицательные приращения узловых напряжений на остальных

узлах с индексами $j \in M$. Выражения (2.12), (2.11), (2.13) показывают, что дает подобное изменение переменной x_k как в отношении остальных переменных, так и в отношении целевой функции.

§ 3. Примеры решения задач по методу I

В предыдущем параграфе мы рассмотрели метод решения задач линейного программирования типа (2.1) — (2.4) на электрической модели, описываемой системой линейных алгебраических уравнений вида (2.5): $AV=J$. Матрица $A=[a_{ij}]$ в уравнениях (2.5) для электрической цепи, составленной только из активных сопротивлений и источников тока, будет обладать следующими особенностями:

$$1) \quad |a_{ii}| \geq \left| \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \right|; \quad (3.1)$$

$$2) \quad a_{ij} \leq 0, \quad i \neq j; \quad (3.2)$$

$$3) \text{ матрица симметричная: } a_{sw} = a_{ws}. \quad (3.3)$$

Этим условиям должна удовлетворять и задача оптимизации, решаемая на данной модели. В общем случае, используя известные приемы [29, 81], заключающиеся в применении реактивных проводимостей, можно снять ограничения (3.1), (3.2). Но при этом следует учитывать, что в рассматриваемых задачах оптимизации условия (3.2) есть условия того, что матрица A является S -матрицей. Условия (3.1) — это ослабленные условия Адамара [19], обеспечивающие хорошую обусловленность матрицы A . Поэтому снятия ограничений (3.2) в данном случае не требуется, а выполнение условий (3.1) желательно обеспечить при постановке задачи. Поэтому мы будем рассматривать примеры решения на модели S -задач двух типов: а) задачи с симметричной матрицей A ; б) задачи с несимметричной матрицей A .

а) Решение задач с симметричной матрицей A . Известно довольно много объектов, описываемых системой линейных алгебраических уравнений с симметричной матрицей. Такие объекты встречаются при расчете электрических цепей, при решении дифференциальных уравнений в частных производных эллиптического типа (после конечно-разностной аппроксимации), в строительной механике, в

гидродинамике и в других областях. При постановке и решении задач оптимизации в упомянутых областях может оказаться весьма удобным решать данные задачи непосредственно на этапе моделирования. Решение может быть осуществлено на электрических моделях [29, 81], предназначенных для решения систем линейных алгебраических уравнений (см. ниже рис. 6.3) или на сеточных электроинтеграторах.

Рассмотрим в качестве примера решение на модели задачи оптимального управления режимом нефтяного пласта. Электромоделирование процессов разработки нефтяных пластов получило в настоящее время широкое распространение, поэтому возможность проведения оптимизации одновременно с моделированием для данных задач особенно важна. При решении задачи оптимизации непосредственно на модели отпадает необходимость в трудоемкой операции съема модельных данных (матрицы A и других), отпадает необходимость в программировании и т. д. Причем при решении данных задач ЭЦВМ не имеет преимуществ перед моделью в отношении точности получаемого решения, так как исходные данные задачи оптимизации снимаются с сеточной модели, и решение задачи с точностью выше точности исходных данных смысла не имеет.

Обычно моделирование нефтяного пласта производится на сеточном электроинтеграторе. Задача оптимизации режима нефтяного пласта заключается в определении оптимальных дебитов (т. е. отборов нефти) по скважинам может быть сформулирована в виде S -задачи (см. задачу (2.15) в главе I). Описанная в предыдущем параграфе процедура решения задачи на модели интерпретируется в терминах задачи оптимального управления нефтедобывающим предприятием следующим образом. Пусть токи J в узлах сетки соответствуют дебитам скважин, а напряжения V — депрессиям (т. е. разности давлений в каждом узле относительно давления на контуре питания). Электрическая сетка, моделирующая нефтяной пласт, изображена на рис. 6.1. В исходной точке все скважины включены на предельные депрессии $V=V_{\text{пр}}$, т. е. дебиты $J=X^0=B$. Процедура решения в данном случае заключается в определении номеров скважин, отключение которых (или переключение на наименьший режим по дебиту) дает положительное приращение целевой функции задачи. Причем, если скважину целесообразно отключить ($\delta L_j > 0$),

то она останется отключенной и в оптимальной точке. Очевидно, что выключены могут быть сильно обводненные скважины. Поэтому для решения задачи, по сути дела, достаточно только проверить целесообразность ($\delta L_j > 0$) отключения сильно обводненных скважин.

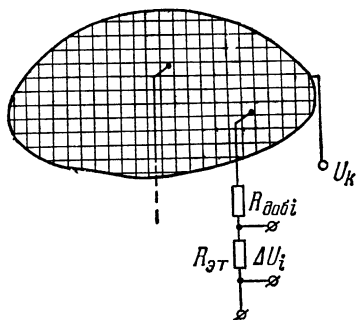


Рис. 6.1. Электрическая сетка, моделирующая нефтяной пласт.

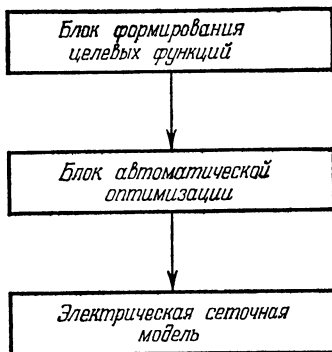


Рис. 6.2. Блок-схема устройства для решения задачи с использованием декомпозиции.

Описанная процедура оптимизации может быть осуществлена автоматически с помощью небольшой приставки к сетке, состоящей из сумматора и распределителя. Сумматор предназначен для расчета на каждом шаге величины целевой функции, а распределитель — для перехода к следующему шагу (т. е. поочередного отключения скважин).

Отметим, что в некоторых случаях можно таким образом решать на сеточной модели и задачи оптимизации нефтяных месторождений в упругом режиме. При наличии некоторых производственных или плановых ограничений, не сводящихся к требуемому виду, решение может быть получено на основе принципа декомпозиции как линейная комбинация решения рассмотренной задачи при разных целевых функциях. Блок-схема устройства для решения задачи в этом случае представлена на рис. 6.2.

Решение задачи оптимизации нефтедобывающего предприятия может быть получено также на электрических схемах, предназначенных для решения систем линейных алгебраических уравнений с симметричной матрицей. Данные схемы значительно дешевле сеточных электроинтеграторов.

Поэтому в тех случаях, когда известна матрица коэффициентов влияния, использование подобных схем предпочтительнее. Схема изображена на рис. 6.3, где проводимости g_{ij} следующие:

$$g_{ij} = k_g a_{ij} \quad \text{при } i \neq j, \quad (3.4)$$

$$g_{i0} = k_g \left(a_{ii} - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} \right) \quad \text{при } i = j; \quad (3.5)$$

здесь k_g — коэффициент пропорциональности.

Каждый шаг решения задачи в этом случае, как и прежде, будет заключаться в отключении тока одного узла при

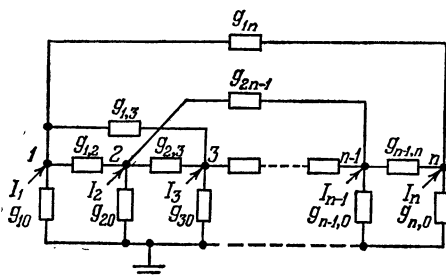


Рис. 6.3. Схема из активных сопротивлений для решения систем линейных алгебраических уравнений.

сохранении постоянства напряжений на всех остальных узлах и в определении знака приращения δL целевой функции задачи.

Отметим, что оценки δL при решении задач оптимизации нефтедобычи удобнее считать, используя выражения (2.17), (2.14) или (2.15), (2.16). Это объясняется тем, что в данном случае известен именно вектор C^0 , например, это может быть, физически, вектор коэффициентов нефтесодержания скважин (X^0 — дебиты скважин, C^0 — нефтесодержание продукции скважин).

В данных задачах может оказаться весьма плодотворным исследование выражений (2.15) — (2.17), получаемых в процессе решения. Действительно, уменьшение дебита некоторой k -й скважины (уменьшение переменной x_k^0) вызывает увеличение дебитов остальных скважин (увеличение остальных переменных x_k^0). Выражения (2.16) позволяют количест-

венно оценить подобное изменение дебитов по каждой скважине. Приращение целевой функции (2.15) складывается из следующего. Если c_j^0 — коэффициент нефтесодержания продукции скважин, то $c_k^0 \cdot \delta x_k^0$ — это потеря в добыче чистой нефти, вызванная уменьшением дебита k -й скважины на данном шаге решения. Величины $c_j^0 \cdot \delta x_j^0$ для $j \neq k$ показывают, насколько при этом увеличится добыча чистой нефти по остальным скважинам. Сумма (2.15) показывает, что больше — потеря нефти на k -й скважине или суммарное увеличение добычи на остальных скважинах, получаемое за счет уменьшения дебита k -й скважины. Если k -я скважина сильно обводнена, то $c_k^0 \delta x_k^0$ может оказаться меньше, чем $\sum_{j \neq k} c_j^0 \delta x_j^0$, и значение δL_k будет положительным.

Кроме очевидной физической интерпретации, исследование выражений (2.15), (2.17), (2.16) полезно из следующих соображений. Значения δx_j^0 , δL показывают, как влияет изменение дебита данной скважины на производительности других скважин и на значение функции цели, что очень важно при рассмотрении вопроса о проведении на промысле различных геолого-технических мероприятий. Заметим, что при этом можно определить также изменения депрессий по скважинам. Это можно сделать, измерив величины

$$\delta x_j = \frac{1}{k_v} [v_j^{(p-1)} - v_j^{(p)}].$$

Конкретная процедура решения задачи будет полностью соответствовать процедуре, описанной в предыдущем параграфе.

Приведем пример решения задачи оптимизации режима нефтяного месторождения на электрической сеточной модели. Рассмотрим нефтяной пласт с семью действующими скважинами. Пласт моделировался *) на электроинтеграторе ЭИ-12. Схематическое изображение электрической сетки, моделирующей данное нефтяное месторождение, приведено на рис. 6.4. На рисунке узлы, в которых расположены скважины, заключены в треугольник. Пласт кусочно-однородный **). Границы каждой зоны однородности обведены

*) Моделирование было выполнено старшим инженером МИНХ и ГП Мироновым Н. К.

**) Естественно, предположение о кусочно-однородном характере пласта не является обязательным для применения методов оптимизации.

жирной линией. Значения сопротивления между узлами для каждой зоны однородности записаны также прямо на рисунке. Эти значения электрических сопротивлений пропорциональны гидродинамическим сопротивлениям пласта.

Заданы следующие величины: а) давление на контуре пласта: $p_k = 200 \text{ атм}$; б) из технологических соображений

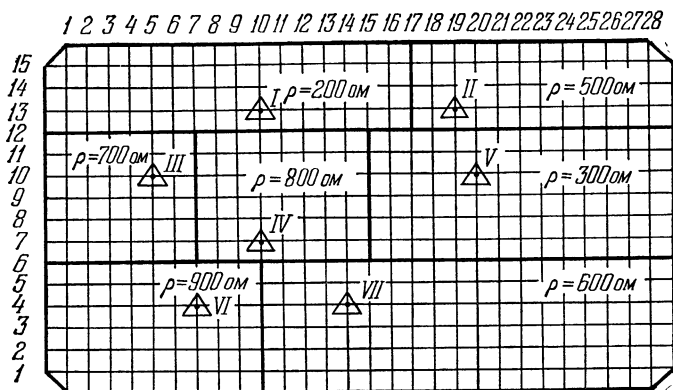


Рис. 6.4. Модель нефтяного пласта с семью скважинами.

определены предельные забойные давления $p_j^{\text{пр}}$, ниже которых нельзя опускать давления на забоях скважин; в) коэффициенты нефтесодержания c_j^0 скважин. Требуется определить режимы работы скважин, позволяющие получить максимальную суммарную добычу чистой нефти по месторождению при удовлетворении имеющимся ограничениям (см. формулировку задачи (2.15) в главе I). Вместо предельных забойных давлений можно рассматривать предельные депрессии $\delta p^{\text{пр}} = p_k - p_j^{\text{пр}}$. Это депрессии, выше которых нельзя увеличивать значения депрессий по скважинам.

Установим в исходной точке депрессии по скважинам равными предельным значениям. Замерим дебиты скважин Q , полученные при этом: $Q = B = A \cdot \delta p^{\text{пр}}$. (Аналогию между терминологией электрических цепей и терминами описания нефтедобычи мы уже проводили: $J \sim Q$, $V \sim \delta P$. Здесь для ясности мы пользуемся терминологией нефтедобычи.) Все заданные значения приведены в таблице 6.1. Забойные давления, депрессии и дебиты приведены в таблице как в натурных,

так и в модельных величинах. Коэффициенты подобия натуральных и модельных значений давлений p (или депрессий), гидродинамических сопротивлений r и дебитов q брались следующие: $k_p = 1 \text{ атм}/\%$, $k_r = 2 \cdot 10^{-4} \%$ сантипуаз/дарси $\cdot \text{м}$, $k_q = 172,8 \text{ м}^3/\text{сут}/\%$.

Таблица 6.1

Заданные значения коэффициентов нефтесодержания, депрессий и дебитов скважин

№ скважин	Коэффициенты нефтесодержания, c_j^0	Натурные значения			Модельные значения		
		забойные давления, $p_j^{\text{пр}}$ (атм)	дебиты жидкости q_j ($\text{м}^3/\text{сут}$)	дебиты нефти $c_j^0 q_j$ ($\text{м}^3/\text{сут}$)	депрессии $\delta p_j^{\text{пр}} \%$ при $p_k = 200 \text{ атм}$	дебиты жидкости, $q_j \%$	дебиты нефти, $c_j^0 q_j$ (%)
1	0,9	150	183,2	164,9	50	1,060	0,9540
2	0,8	175	78,8	63,0	25	0,456	0,3648
3	0,05	160	71,8	3,6	40	0,415	0,0207
4	0,9	110	172,8	155,5	90	1,000	0,9000
5	0,05	180	81,4	4,1	20	0,471	0,0237
6	0,05	170	43,8	2,2	30	0,254	0,0127
7	0,9	165	79,1	71,2	35	0,458	0,4122
Σ			710,9	464,5		4,114	2,6881

Решение задачи оптимизации заключалось в следующем. Пусть из технологических условий допускается отключение скважин, если это окажется целесообразно. Будем проверять в первую очередь целесообразность отключения сильно обводненных скважин. Начинаем со скважины № 2. Отключение этой скважины вызвало увеличение целевой функции (таблица 6.2), поэтому оставляем эту скважину отключенной. То же самое имеет место при отключении скважин № 5 и 6. Отключение остальных скважин не дает увеличения целевой функции. Эти скважины оставляем включенными. Решение на этом заканчивается. Ход решения задачи приведен в таблице 6.2. Все значения величин в таблице — модельные. Оптимальные режимы скважин приведены в последних двух столбцах данной таблицы.

Если из технологических условий отключение скважины не допускается, а допустимым считается только уменьшение дебита некоторых скважин до некоторого уровня d_j , то ход

решения задачи остается тем же самым. Меняется только исходный режим: если при условиях $X^0 \geq 0$ мы имеем $X_{исх}^0 = B$, то при условиях $X^0 \geq D$ (где $D = [d_j]$), мы переходим к переменным $X^{0'} = X^0 - D \geq 0$ и имеем $X_{исх}^{0'} = B - D$. В данном случае, естественно, предполагается, что значения наименьших

Таблица 6.2

Ход решения задачи оптимизации на сеточной модели

№ скважин	Коэффициенты нефтесодержания, c_j^0	1-й шаг		2-й шаг		3-й шаг	
		дебит жидкости q_j	дебит нефти $c_j^0 q_j$	дебит жидкости q_j	дебит нефти $c_j^0 q_j$	дебит жидкости q_j	дебит нефти $c_j^0 q_j$
1	0,9	1,08	0,972	1,09	0,981	1,09	0,981
2	0,8	0,457	0,3656	0,514	0,4112	0,515	0,412
3	0,05	0	0	0	0	0	0
4	0,9	1,01	0,909	1,01	0,909	1,03	0,927
5	0,05	0,475	0,02375	0	0	0	0
6	0,05	0,267	0,01385	0,268	0,0139	0	0
7	0,9	0,462	0,4158	0,476	0,4284	0,484	0,4356
Σ			2,70465		2,74300		2,7556

допустимых дебитов скважин D меньше значений дебитов при наибольших допустимых депрессиях B , т. е. $B - D \geq 0$.

б) Решение задач с несимметричной матрицей. Простейшая матричная схема, моделирующая несимметричную матрицу, как известно, может быть получена из схемы, изображенной на рис. 6.3, путем включения в ее узлы катодных повторителей. Схема с катодными повторителями для решения систем линейных алгебраических уравнений с несимметричными матрицами (при $n=3$) изображена на рис. 6.5, где проводимости g_{ij} и g_{i0} рассчитываются по формулам (3.4), (3.5), а математическое описание ($AV=J$) аналогично описанию схемы рис. 6.3. Решение задачи оптимизации на данной модели совершенно аналогично решению подобных задач на модели рис. 6.3.

Рассмотрим решение одной из задач оптимизации с несимметричной матрицей, а именно — задачи оптимизации межотраслевого планирования. Данная задача, сформулиро-

ванная в главе I в виде статической модели Леонтьева, после сведения к S -задаче записывается в виде (5.21) (см. § 5 главы I). Эта S -задача линейного программирования с несимметричной матрицей A может быть решена на модели рис. 6.5 с помощью метода, предложенного в предыдущем параграфе. Чтобы не менять обозначений задачи (5.21), мы проведем аналогию между задачей (5.21) из первой главы и задачами (2.1), (2.6), (2.4), которую мы рассматривали в предыдущем параграфе. Вектор переменных Y_1 из задачи (5.21

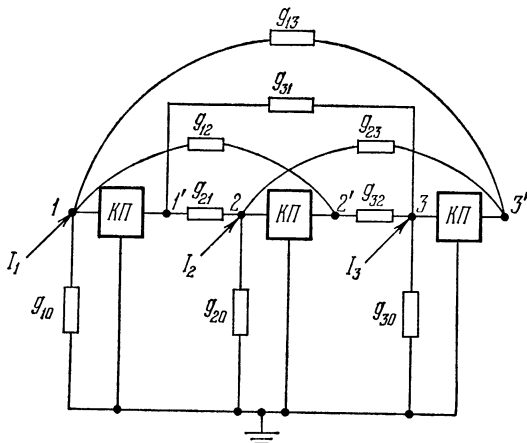


Рис. 6.5. Схема с катодными повторителями для решения систем линейных алгебраических уравнений с несимметричной матрицей.

соответствует вектору X задачи (2.1), (2.6), (2.4), вектор Y_2 — соответствует вектору X^0 , вектор $-\Delta_1$ из (5.21) соответствует вектору C из (2.1), вектор C из (5.21) соответствует вектору B из (2.6). Такое соответствие обозначений не должно вызывать удивления, так как задача (5.21) является двойственной по отношению к исходной задаче (5.15) — (5.18) оптимизации межотраслевого планирования (см. § 5 главы I). Вектор переменных Y_1 из задачи (5.21) соответствует в исходной задаче условиям (5.16), ограничивающим объем конечного продукта, вектор Y_2 соответствует производственным ограничениям (5.17) на объем производимого продукта. Если в оптимальной точке некоторая переменная y_i вектора Y_1 является положительной, то это означает (на основании известных теорем линейного программирования), что i -е

условие системы (5.16) выполняется как равенство. Если $y_i=0$, то i -е условие (5.16) выполняется как строгое неравенство. То же самое можно сказать относительно соотношения между переменными y_{i+n} вектора Y_2 и условиями системы (5.17).

Отсюда ясна физическая интерпретация процесса решения задачи (5.21). Задача (5.21) является S -задачей, поэтому для каждого i только одна из двух переменных y_i, y_{i+n} может быть положительной. В исходной точке все переменные y_{i+n} положительные (в невырожденном случае), а все переменные y_i нулевые: $Y_2=C, Y_1=0$. Для решения задачи (5.21) необходимо выяснить, какие из переменных y_i имеет смысл перевести в ненулевые вместо переменных y_{i+n} с тем же индексом i . С точки зрения исходной задачи (5.15) — (5.18) это означает выяснить, какие из требований конечного спроса можно и нужно выполнить (какие из условий (5.16) обратить в равенства). Соответствующие ограничения (5.17) будут в этом случае выполняться как строгие неравенства.

Сравнение обозначений в задаче (5.21) первой главы и задачи (2.1), (2.6), (2.4) предыдущего параграфа показывает, что в данном случае нужно принять следующую аналогию между электрическими величинами и переменными задачи (5.21): вектор J задающих токов в узлах пропорционален вектору Y_2 , вектор разности узловых напряжений ($V_{\text{пр}}-V$) пропорционален вектору Y_1 . В исходной точке узловые напряжения установлены на предельные значения: $V=V_{\text{пр}}$ ($Y_1=0$), все задающие токи узлов включены ($Y_2>0$). Каждый шаг решения, как и ранее, будет заключаться в отключении тока, подаваемого в один из узлов, и в вычислении оценки, «целесообразности» ($\delta L>0$) отключения данного тока. Оценки δL в данном случае удобнее вычислять по формулам, аналогичным (2.13, 2.14):

$$\delta L = L^{(p)} - L^*,$$

где

$$L^{(p)} = \sum_{i \in M} (-\Delta_i) y_i^{(p)}, \quad y_i^{(p)} = \frac{1}{k_v} (v_{i\text{пр}} - v_i^{(p)}).$$

Решение закончено, когда все $\delta L \leq 0$. Значения токов, подаваемых в узлы, и значения узловых напряжений в этом случае позволят судить о величине переменных Y_2 и Y_1 задачи (5.21). Это будет решение задачи, двойственной к исходной

задаче оптимизации межотраслевого планирования (5.15) — (5.18) из главы I. Если нам требуется получить решение исходной задачи, то нужно решить систему из n линейных алгебраических уравнений относительно n переменных x_j задачи (5.15) — (5.18). В эту систему уравнений входят те условия из системы (5.16) — (5.17), которые соответствуют ненулевым переменным y_i, y_{i+n} оптимального базиса (5.21). Решение системы осуществляется на той же схеме, что и на рис. 6.5. Матрица этой системы является транспонированной по отношению к матрице оптимального базиса задачи (5.21).

Таким образом, решение предложенным методом S -задач с симметричной матрицей A отличается от решения S -задач с несимметричной матрицей двумя моментами. Во-первых, решение S -задач с симметричной матрицей может осуществляться на достаточно простых электрических моделях, составленных из активных сопротивлений и источников. Для решения задач с несимметричными матрицами нужны более сложные модели, например, модели, в которых кроме сопротивлений и источников, имеются катодные повторители. Во-вторых, в случае рассмотрения S -задач с симметричной матрицей проще получить решение задачи, двойственной к исходной задаче (2.1), (2.6), (2.4). Это можно осуществить, например, если вместо задающих токов узлов $J=k_j B$ подать в исходной точке в узлы токи $J=k_j(-C)$, а затем отключить отрицательные задающие токи. Оптимальный базис при этом останется тем же, а количественные значения токов J и напряжений ($V_{пр}-V$) будут пропорциональны соответствующим переменным двойственной задачи. При рассмотрении S -задач с несимметричной матрицей для получения количественного решения двойственной задачи нужно, кроме изменения исходных задающих токов узлов, изменить набранную на модели матрицу на матрицу, транспонированную к ней (поменять местами узлы k и k' на схеме рис. 6.5 с катодными повторителями).

§ 4. Метод решения II

В предложенном выше методе решения S -задач на электрических моделирующих установках мы, в соответствии с алгоритмами главы III, последовательно проверяем, какие переменные x_j следует перевести в ненулевые (или какие задающие токи узлов, пропорциональные переменным

x_j^0 , следует перевести в нулевые). Для осуществления этой проверки необходимо иметь дополнительно к аналоговому устройству сумматор, позволяющий вычислять эффект

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

этого перевода. Но специальные свойства S -задачи позволяют построить алгоритмы решения ее на стандартных

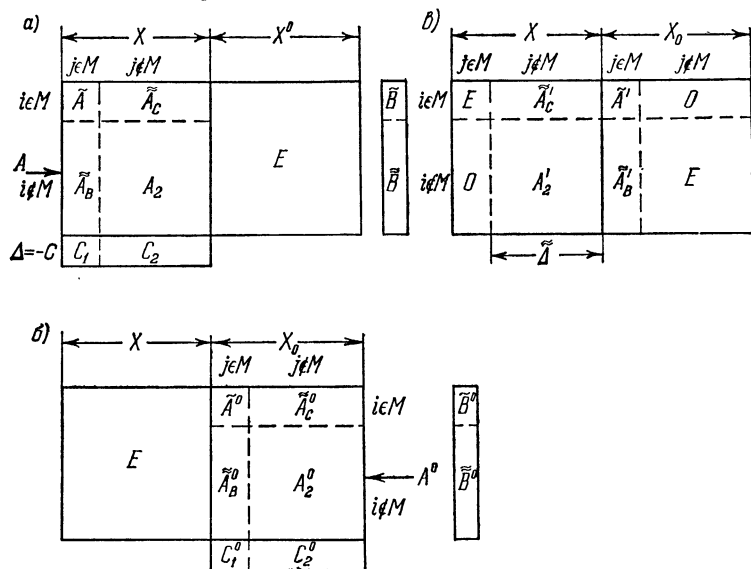


Рис. 6.6. Матрица S -задачи для различных опорных планов: а) $X=0$, $X^0>0$; б) $X>0$, $X^0=0$; в) $x_j>0$, $j \in M$; $x_j=0$, $j \notin M$; $x_j^0=0$, $j \in M$; $x_j^0>0$, $j \notin M$.

моделирующих установках вообще без привлечения каких-либо дополнительных устройств. Метод, предлагаемый ниже, дает возможность реализовать подобную процедуру.

Будем руководствоваться следующей идеей. Мы пользуемся для решения S -задачи аналоговыми устройствами (см. рис. 6.1, 6.3, 6.5), позволяющими решать системы линейных алгебраических уравнений. Поэтому, чтобы решать S -задачи на модели без дополнительного сумматора, следует определять целесообразность перевода переменных x_j в ненулевые

также с помощью решения систем линейных алгебраических уравнений. В принципе это вполне осуществимо. Трудность в том, чтобы реализовать эту идею, не перестраивая модель на каждом шаге, — только тогда алгоритм решения задачи получается достаточно простым.

Запишем S -задачу в матричном виде:

$$\left. \begin{aligned} CX \rightarrow \max, \quad AX + EX^0 = B, \\ X \geq 0, \quad X^0 \geq 0. \end{aligned} \right\} \quad (4.1)$$

Исходный опорный план для задачи (4.1) следующий: $X^0=B$, $X=0$. Симплекс-таблица для данной точки изображена на рис. 6.6, а. Ту же задачу можно записать иначе, при другом исходном базисе:

$$\left. \begin{aligned} C^0 X^0 \rightarrow \max, \quad EX + A^0 X^0 = B^0, \\ X \geq 0, \quad X^0 \geq 0, \end{aligned} \right\} \quad (4.2)$$

где

$$A^0 = A^{-1}, \quad C^0 = -CA^{-1}, \quad B^0 = A^{-1}B \geq 0.$$

Исходный опорный план для задачи (4.2) $X=B^0$, $X^0=0$. Симплекс-таблица для данной точки приведена на рис. 6.6, б. Предположим, что мы используем для решения S -задачи алгоритм из пункта 1 § 2 третьей главы, блок-схема которого представлена на рис. 3.5. В исходной точке мы имеем вектор оценок $\Delta = -C$, если задача записана в виде (4.1), или

$$\Delta = -C = C^0 A, \quad (4.3)$$

если задача записана в виде (4.2). Предположим далее, что (в соответствии с алгоритмом рис. 3.5) мы составим множество M индексов переменных x_j , которые следует перевести в ненулевые *): $x_j > 0$, $j \in M$, $x_j^0 > 0$, $j \notin M$. (Симплекс-таблица для данной точки представлена на рис. 6.6, в.) Нас интересуют оценки $\tilde{\Delta}_j$ тех переменных x_j , $j \notin M$, которые остались нулевыми. Данные оценки можно получить так, как это делается в § 2 главы III (см. выражение (2.1)). Но нам было бы интересно получить эти оценки прямо из решения системы линейных алгебраических уравнений. Это нетрудно сделать.

*) Мы, как и ранее, в § 2, говорим о «перевод в ненулевые», считая для простоты все планы невырожденными. Более обще было бы вместо «перевод в ненулевые» говорить «перевод в базисные переменные».

Действительно, пусть симплекс-таблица рис. 6.6, в получена из симплекс-таблицы рис. 6.6, б. Тогда по первой форме оптимальности [93]

$$\tilde{\Delta} = C_2^0 \cdot A_2', \quad (4.4)$$

где $C_2^0 = [c_j^0]$, $j \notin M$; $A_2' = (A_2^0)^{-1}$ — исходя из преобразования таблицы рис. 6.6, б в таблицу рис. 6.6, в по схеме блочного метода исключения. Матрицу A_2' можно получить также из таблицы рис. 6.6, а. Тогда

$$A_2' = A_2 - \tilde{A}_B \tilde{A}^{-1} \tilde{A}_C. \quad (4.5)$$

Рассмотрим теперь электрические аналоги приведенных выражений. Система уравнений, связывающих вектор токов J , подаваемых в узлы, с вектором узловых напряжений V , записывается, как уже отмечалось выше, следующим образом:

$$J = AV. \quad (4.6)$$

Эта запись соответствует случаю, когда задающие токи включены во все n узлов электрической цепи с индексами $j=1, \dots, n$. Предположим, далее, что мы отключили задающие токи в узлах с индексами $j \in M$, сохранив узловые потенциалы в остальных узлах (т. е. в узлах с индексами $j \notin M$) на прежнем уровне. Вектор задающих токов, переведенных в нулевые, обозначим J_1 . Вектор ненулевых задающих токов обозначим J_2 . Разделим соответственно вектор V и матрицу A . Тогда система (4.6) запишется

$$\begin{array}{c} \begin{array}{cc} & j \in M & j \notin M \\ \begin{array}{c} i \in M \\ i \notin M \end{array} & \begin{array}{|c|c|} \hline \tilde{A} & \tilde{A}_C \\ \hline \tilde{A}_B & A_2 \\ \hline \end{array} & \cdot \begin{array}{|c|} \hline V_1 \\ \hline V_2 \\ \hline \end{array} = \begin{array}{|c|} \hline J_1 \\ \hline J_2 \\ \hline \end{array} \end{array}$$

т. е.

$$\tilde{A}V_1 + \tilde{A}_C V_2 = 0, \quad (4.7)$$

$$\tilde{A}_B V_1 + A_2 V_2 = J_2. \quad (4.8)$$

Переменные V , заданы (сохранили те же значения, что и в (4.6)). Система (4.7), (4.8) — это система n уравнений

относительно n неизвестных V_1, J_2 . Значения V_1 из (4.7)

$$V_1 = -\tilde{A}^{-1}\tilde{A}_c V_2.$$

Подставив это выражение для V_1 в (4.8), получим систему относительно ненулевых задающих токов:

$$J_2 = A'_2 V_2, \quad (4.9)$$

где

$$A'_2 = A_2 - \tilde{A}_B A^{-1} \tilde{A}_c. \quad (4.10)$$

Таким образом, мы исключили из рассмотрения нулевые задающие токи (выразив их через ненулевые) и получили для данной точки описание в виде системы (4.9). Сравним теперь выражения (4.3) с выражением (4.6), а выражения (4.4), (4.5) с выражениями (4.9), (4.10). Из этого сравнения очевидно, что в качестве электрического аналога вектора C^0 можно использовать вектор V узловых напряжений, а электрическим аналогом вектора Δ (или $\tilde{\Delta}$) оценок внебазисных переменных x_j можно считать вектор J (или J_2) ненулевых значений задающих токов узлов. Причем, в выражениях (4.6), (4.9), связывающих электрические величины, использованы матрицы, транспонированные по отношению к матрицам, используемым в выражениях (4.3), (4.4).

Действительно, в выражениях (4.6), (4.9) умножение на матрицу происходит слева, а в выражениях (4.3), (4.4) умножение на матрицу — справа. Для эквивалентности данных выражений нужно умножать в (4.6), (4.9) векторы V, V_2 слева, но на матрицу, транспонированную к матрице A . Практически данное замечание выливается в то, что для соответствия выражения (4.3) выражению (4.6), а выражения (4.4) выражению (4.9) нужно набирать на электрической модели матрицу, транспонированную по отношению к матрице A из (4.3) и соответственно к матрице A'_2 из (4.4). Для симметричных матриц это не имеет никакого значения (здесь $A^T = A$), а для несимметричных матриц последнее замечание следует учитывать. Например, в схеме, изображенной на рис. 6.5, для перехода к транспонированной матрице достаточно поменять местами узлы 1 и 1', 2 и 2', 3 и 3'.

Перейдем к описанию алгоритма решения задачи на моделирующих устройствах. Заметим предварительно, что мы рассматриваем решение задачи (4.1), но, как следует из

приведенного обоснования, при решении задачи используется вектор C^0 из (4.2). Это несколько не усложняет процедуры решения задачи на модели. Но для ясности изложения мы отдельно рассмотрим случай, когда вектор C^0 задан, и случай, когда задан вектор C , а вектор C^0 заранее не известен. Оба эти случая встречаются на практике.

Случай 1. Вектор C^0 известен.

Алгоритм решения задачи (4.1) на модели (4.6) заключается в следующем:

1) Задаем значения компонент вектора узловых напряжений пропорциональными значениям соответствующих компонент вектора C^0 : $C^0 k_v = V$ (где k_v — коэффициент пропорциональности).

2) Замеряем направления токов, подаваемых при этом в узлы: $J = k_j \Delta$. Отрицательные токи отключаем, сохраняя узловые напряжения в остальных узлах на прежнем уровне (модель работает в режиме заданных напряжений). Если после отключения появятся еще отрицательные токи, то их также отключаем. При этом, как и ранее, сохраняем на прежнем уровне узловые напряжения в узлах, в которых задающие токи остались неотключенными. Решение закончено, когда отрицательных токов больше не имеется. Это решение заключается в выделении множества M узлов, в которых задающие токи оказались в итоге отключены ($\Delta_j = 0$, $j \in M$). Соответственно, оптимальный базис (т. е. качественное решение) задачи (4.1) имеет вид

$$x_j > 0, j \in M; x_j = 0, j \notin M, x_j^0 > 0, j \notin M; x_j^0 = 0, j \in M.$$

3) После нахождения качественного решения задачи могут быть найдены и количественные значения переменных в оптимальной точке (количественное решение). Для этого нужно изменить набранную на модели матрицу на транспонированную (для несимметричных задач), и задать исходные токи узлов пропорционально компонентам вектора B задачи (4.1). Затем, отключив токи J_j узлов с индексами $j \in M$, мы получим в оптимальной точке вектор токов $J \sim X_{\text{опт}}^0$ и вектор напряжений $V_{\text{пр}} = V \sim X_{\text{опт}}$.

Количественное решение задачи можно получить также, если после изменения набранной на модели матрицы на транспонированную установить в узлах, соответствующих $j \notin M$, напряжения $V_2 = V_{2\text{пр}} = k_v \tilde{B}^0$. Тогда компоненты вектора токов J , пропорциональны значениям ненулевых

переменных x_j^0 ($j \notin M$) в оптимальной точке: $J_2 = k_J X_2^0$, а раз-
ница предельных и реальных напряжений в узлах $j \in M$ про-
порциональна значениям ненулевых переменных x_j ($j \in M$)
в оптимальной точке

$$V_{2np} - V_2 = k_v \tilde{X}.$$

Заметим, что при количественном решении задачи мы ис-
пользуем те же электрические аналоги переменных, какие
использовались ранее в методе I. Отсюда понятно обоснова-
ние процедуры количественного решения. Это обоснование
можно записать и в терминах данного параграфа. Тогда для
 $j \notin M$ вектор ненулевых компонент x_j^0 запишется (если исхо-
дить из таблиц рис. 6.6, б, в) так:

$$X_2^0 = A_2' \tilde{B}^0,$$

где

$$X_2^0 = [x_j^0], \quad i \notin M; \quad \tilde{B}^0 = [b_i^0], \quad i \notin M; \quad A_2' = (A_2^0)^{-1}.$$

Отсюда (см. (4.9)), если $\tilde{B}^0$ пропорционально V_2 , то X_2^0
пропорционально J_2 .

Случай 2. Известны компоненты вектора C , вектор C^0
заранее не известен.

Алгоритм решения задачи на модели в этом случае свод-
ится к случаю 1. Отличие имеется только в п. 1 алгоритма.
В данном случае п. 1 записывается так:

1) Устанавливаем значения задающих токов в узлах про-
порциональными значениям соответствующих компонент
вектора $-C$: $J = -k_J C$ (при этих токах узловые напряже-
ния $V = k_v C^0$).

Далее решение продолжается в соответствии с алгорит-
мом, приведенном при рассмотрении случая 1. Таким обра-
зом, алгоритм решения в случае 1 отличается от алгоритма
случая 2 только тем, что в последнем случае вместо задания
узловых напряжений в исходной точке задаются токи.

Таким образом, предлагаемая в данном параграфе
процедура решения задачи на модели не требует использова-
ния никаких дополнительных устройств и отличается про-
стотой — для решения задачи достаточно только отклю-
чить отрицательные задающие токи.

§ 5. Примеры решения задач по методу II

Решение задач по методу II проводится на тех же моделях, что и при использовании метода I. Поэтому, естественно, остаются в силе и все те соображения об особенностях матрицы A , описывающей электрическую цепь, которые мы приводили при рассмотрении примеров решения задач по методу I в начале § 3. По-прежнему, решение S -задач с симметричной матрицей A при диагональном преобладании можно осуществлять на электрических матричных схемах из активных сопротивлений (см. рис. 6.3) или на сеточных электроинтеграторах. Решение S -задач с несимметричными матрицами при диагональном преобладании может быть осуществлено, например, на электрических матричных схемах из активных сопротивлений с катодными повторителями (см. рис. 6.5). Для решения задач с матрицами без диагонального преобладания можно использовать матричные схемы с реактивными сопротивлениями.

Ниже мы рассмотрим решение на моделях с использованием метода II тех же примеров, которые мы рассматривали ранее в § 3 при использовании метода I.

а) Решение задач с симметричной матрицей. В качестве примера подобной задачи рассмотрим, как и в § 3, задачу оптимизации нефтедобычи. В данной задаче вектор C^0 известен — это может быть, например, вектор коэффициентов нефтесодержания скважин. Поэтому решение данной задачи методом II будем строить для случая 1 (см. предыдущий параграф).

Процедура решения задачи осуществляется на моделях, изображенных на рис. 6.1 или рис. 6.3. Решение состоит в следующем. Задаем значения узловых напряжений в узлах, соответствующих скважинам, пропорциональными значениям коэффициентов нефтесодержания. Токи в узлах, которые при этом будут отрицательными, отключаем, и получаем качественное решение задачи (т. е. искомый оптимальный базис).

В терминах нефтедобычи эта процедура звучит следующим образом: на модели нефтяного пласта депрессии по скважинам задаются пропорциональными коэффициентам нефтесодержания; скважины, дебиты которых при этом оказываются отрицательными, отключаются.

Для получения количественного решения делается следующее. На скважинах, которые остались включенными, устанавливаются значения давлений (или депрессий), пропорциональные предельным значениям давлений (депрессий). Дебиты скважин (т. е. токи в узлах) тогда получаются пропорциональными компонентам $x_j^0 > 0$ ($j \notin M$) в оптимальной точке. А разница предельных и реальных значений депрессий в узлах, соответствующих отключенным скважинам, будет пропорциональна компонентам $x_j > 0$ ($j \in M$) в оптимальной точке. Остальные компоненты x_j^0 и x_j равны нулю.

б) Решение задач с несимметричной матрицей. В качестве примера подобной задачи рассмотрим, как и в § 3, задачу оптимизации межотраслевого планирования. Будем решать S -задачу (5.21) из первой главы, которая является двойственной по отношению к исходной задаче межотраслевого планирования. Соответствие между обозначениями задачи (5.21) первой главы и обозначениями, принятыми в данной главе (см. задачу (4.1)), как мы уже указывали в § 3 главы VI, будет следующим: вектора Y_1, Y_2, Δ_1, C задачи (5.21) соответствуют векторам X, X^0, C, B задачи (4.1).

Решение задачи (5.21), как и ранее, осуществляется на электрической матричной модели, предназначенной для решения систем линейных алгебраических уравнений с несимметричной матрицей, например, на модели, изображенной на рис. 6.5. Решение проводится для случая, когда неизвестны компоненты целевой функции при переменных X^0 (случай 2 из предыдущего параграфа). Процедура решения заключается в следующем. Набираем на модели матрицу $(E - A)$, транспонированную по отношению к матрице $(E - A)^T$ ограничений задачи (5.21). Устанавливаем значения задающих токов в узлах пропорциональными значениям соответствующих компонент вектора $-\Delta_1$. Затем отключаем отрицательные токи, сохраняя узловые напряжения в остальных узлах на прежнем уровне. Решение закончено, когда окажется, что отрицательных задающих токов больше не имеется. При этом будет получено качественное решение задачи (5.21), т. е. в оптимальной точке данной задачи будет $y_i > 0, i \in M; y_i = 0, i \notin M; y_{i+n} > 0, i \notin M; y_{i+n} = 0, i \in M$, где y_i — компоненты вектора Y_1, y_{i+n} — компоненты вектора Y_2 . Напомним, что в итоге нам требуется получить решение задачи оптимизации межотраслевого планирования (5.15) — (5.18), двойственной по отношению к рассматриваемой зада-

че. Качественные решения этих задач, естественно, совпадают. А для получения количественного решения задачи (5.15) — (5.18) нужно решить на модели систему уравнений, в которую входят те условия из (5.16), (5.17), которые обращаются в оптимальной точке в равенства. При этом i -е условие (5.16) обращается в равенство, если в оптимальной точке $y_i > 0$, и $(i+n)$ -е условие (5.17) обращается в равенство, если в оптимальной точке $y_{i+n} > 0$.

Таким образом, решение на модели по методу II S -задач с симметричной матрицей отличается от решения S -задач с несимметричной матрицей следующими моментами. Во-первых, при решении задач с несимметричными матрицами используются, как и в методе I, другие электрические модели. Во-вторых, при решении задач с несимметричной матрицей следует учитывать, что при качественном решении на модели должна быть набрана матрица, транспонированная по отношению к исходной. И последнее: если после качественного решения методом II задачи с несимметричной матрицей мы хотим получить количественное решение данной задачи, то следует изменить набранную на модели матрицу на матрицу, транспонированную к ней. Если же после качественного решения данной задачи мы хотим получить количественное решение двойственной задачи, то менять набранную на модели матрицу не следует. Для задач с симметричной матрицей последнее замечание, естественно, можно не учитывать.

§ 6. Сравнение методов I и II

Сравнение предложенных методов решения задачи (2.1) — (2.4) на модели удобно привести в виде таблицы 6.3.

Метод II имеет несомненное преимущество перед методом I в том отношении, что позволяет получить решение задачи на модели без использования каких-либо дополнительных устройств или проведения дополнительных расчетов (при использовании метода I требуется рассчитывать на каждом

шаге оценку $\delta L = \sum_{j=1}^n c_j \delta x_j$). Это позволяет значительно проще решать задачу на модели по методу II: если проводить решение вручную, то достаточно просто отключить отрицательные токи, ничего не вычисляя. Если же строить автоматический оптимизатор для решения задачи, то при использова-

нии метода I потребуется использовать сумматор (для вычисления δL), распределитель (для поочередного отключения задающих токов) и связующие устройства. При использовании метода II ничего этого не требуется — достаточно

Т а б л и ц а 6.3

Сравнение методов решения задачи на электрической модели

	Метод I	Метод II
Аналогия между переменными задачи и электрическими величинами	$X^0 \propto J$ $X \propto V_{\text{пр}} - V$	$\Delta \propto J \text{ (или } \tilde{\Delta} \propto J_2)$ $C^0 \propto J$
Процедура оптимизации	Исходная точка	
	Устанавливаем значения узловых напряжений равными предельным значениям: $V = V_{\text{пр}}$	Устанавливаем значения узловых напряжений равными C^0
	Процедура	
	Отключаем задающие токи в тех узлах, в которых оценка $\delta L > 0$ ($\delta L = \sum_{j=1}^n c_j \delta x_j$). Напряжения в остальных узлах сохраняем при этом на прежнем уровне	Отключаем отрицательные задающие токи. Напряжения в остальных узлах сохраняем при этом на прежнем уровне

включить в узловые точки индикаторы (например, лампочки, загорающиеся при отрицательном токе).

Кроме того, метод II имеет одно преимущество перед методом I, заключающееся в следующем. При использовании метода I мы на каждом шаге вычисляем приращение

δL_j целевой функции от замены одной переменной x_j^0 в базисе на переменную x_j . Но при большом числе переменных целевая функция задачи может настолько незначительно меняться от замены в базисе одной переменной, что уловить это изменение оказывается невозможным. При использовании же метода II мы отключаем все отрицательные токи и тем самым проверяем реакцию на замену в базисе одновременно многих переменных. При этом совершенно не обязательно отключать токи все одновременно — если ток в некотором узле стал отрицательным, то в процессе решения задачи он может получать только отрицательные приращения, т. е. уже больше не станет положительным (это следует из свойств S -задач).

В то же время метод I также имеет определенные преимущества перед методом II. Во-первых, при использовании метода I мы имеем очевидную физическую интерпретацию на всем протяжении процесса решения — мы можем постоянно наблюдать за значениями переменных X , X^0 . В ряде задач это имеет большое значение не только для понимания процесса, то и в процедурном отношении. Например, в задачах управления нефтедобычей метод I позволяет решать задачи оптимизации прямо на модели пласта, не изменяя исходных граничных условий, что требуется при использовании метода II (в методе I токи J пропорциональны дебитам скважин, а в методе II токи J пропорциональны Δ). Кроме этого, подобная физическая интерпретация позволяет исследовать на модели некоторые дополнительные вопросы, например, вопрос о проведении дополнительных мероприятий, позволяющих расширить ограничения. В задачах оптимизации нефтедобычи это будет вопрос о проведении комплекса геолого-технических мероприятий. Причем, при решении подобных вопросов как раз удобно то, что в методе I мы судим об изменении целевой функции отдельно для каждого j . И наконец, еще одним преимуществом метода I является возможность получения количественных значений переменных X , X^0 одновременно с выделением оптимального базиса — при использовании метода II для получения количественного решения необходимо изменить граничные условия.

Таким образом, вопрос о применении того или иного метода решения задачи на модели требуется решать конкретно для каждой рассматриваемой задачи, учитывая приведенные преимущества и недостатки обоих методов.

ПРИЛОЖЕНИЕ

НЕКОТОРЫЕ СВЕДЕНИЯ ИЗ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

По теории линейного программирования в настоящее время имеется большое количество учебников и монографий (см. [20, 32, 39, 93]). Поэтому здесь мы не ставим своей целью изложение основ линейного программирования. Мы приведем только некоторые сведения справочного характера, с тем, чтобы, во-первых, пояснить терминологию, принятую в книге, и во-вторых, устранить необходимость привлечения дополнительной литературы при чтении. И наконец, в-третьих, мы сочли нужным поместить в книгу данный материал потому, что, как нам кажется, приведенные ниже сведения являются основными для понимания книги, и возможно, компактное их изложение по принятой здесь методике будет способствовать лучшему пониманию того, что хотели сказать авторы в основном тексте книги.

Изложение построено следующим образом: в § 1 приведены некоторые необходимые определения; в § 2 рассмотрены основные теоретические вопросы формирования методов линейного программирования, и § 3 посвящен рассмотрению основных вычислительных схем.

§ 1. Некоторые определения

Выпуклые множества. Множество $G \in R^n$ называется *выпуклым*, если вместе с любыми двумя своими точками X_1, X_2 оно содержит соединяющий их отрезок:

$$\mu X_1 + (1 - \mu) X_2, \quad 0 \leq \mu \leq 1.$$

Если множество G выпуклое, то для любых точек $X_i \in G$ множество G будет содержать также точку

$$\sum_{i=1}^s \mu_i X_i, \quad \text{где} \quad \sum_{i=1}^s \mu_i = 1, \quad \mu_i \geq 0.$$

Выпуклым многогранным множеством называется множество G , состоящее из точек $X = [x_1, x_2, \dots, x_n] \in R^n$, удовлетворяющих линейным условиям

$$(A_i, X) = \sum_{j=1}^n a_{ij}x_j \{ \leq, =, \geq \} b_i, \quad i = 1, \dots, q. \quad (1.1)$$

Выпуклым многогранником называется выпуклое многогранное множество при добавлении условия его ограниченности.

Линейное соотношение в пространстве R^n :

$$\sum_{j=1}^n a_j x_j = b,$$

называется *уравнением гиперплоскости*. Соответствующее соотношение в двумерном пространстве определяет прямую, в трехмерном пространстве — плоскость. Пространство R^n делится гиперплоскостью на два полупространства:

$\sum_{j=1}^n a_j x_j \leq b$ и $\sum_{j=1}^n a_j x_j \geq b$. Отсюда можно сказать, что выпуклое многогранное множество — это множество, образованное пересечением конечного числа полупространств и гиперплоскостей.

Крайней точкой выпуклого множества называется точка $X \in G$, для которой не существует двух различных точек X_1 и X_2 , принадлежащих G , таких, что

$$X = \mu X_1 + (1 - \mu) X_2, \quad \text{где } 0 < \mu < 1.$$

Крайние точки выпуклого многогранного множества называют *вершинами* этого множества. Чтобы точка $X \in G$ была вершиной многогранного множества, нужно, чтобы она удовлетворяла, как строгим равенствам, не менее чем n линейно независимым ограничениям системы (1.1). Для определения компонент вектора X , соответствующего вершине многогранного множества, нужно решить систему линейных алгебраических уравнений

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad i = 1, 2, \dots, n. \quad (1.2)$$

Решение системы (1.2) можно провести с использованием преобразований по формулам модифицированных жордановых исключений — это в принципе соответствует решению системы методом полного исключения Жордана — Гаусса.

Один шаг *модифицированных жордановых исключений* (м. ж. и.) заключается в следующем. Пусть мы имеем некоторую матрицу $A=[a_{ij}]$ размера $m \times n$ (в частности, вектор $B=[b_i]$ в (1.2) можно рассматривать как один из столбцов этой матрицы):

$$A = \begin{bmatrix} a_{11} & \dots & a_{1v} & \dots & a_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{r1} & \dots & a_{rv} & \dots & a_{rn} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & \dots & a_{mv} & \dots & a_{mn} \end{bmatrix}.$$

Один шаг м. ж. и. при разрешающем элементе a_{rv} будет заключаться в преобразовании матрицы A в матрицу A' по следующим формулам:

$$\left. \begin{aligned} a'_{ij} &= a_{ij} - \frac{a_{iv}a_{rj}}{a_{rv}}, & i \neq r, & j \neq v, \\ a'_{rj} &= \frac{a_{rj}}{a_{rv}}, & j \neq v; & a'_{iv} = -\frac{a_{iv}}{a_{rv}}, & i \neq r, \\ a'_{rv} &= \frac{1}{a_{rv}}. \end{aligned} \right\} \quad (1.3)$$

Решение системы (1.2) с использованием м. ж. и. будет получено в столбце $B=[b_i]$ после n шагов м. ж. и. с разрешающими элементами $a_{11}, a'_{22}, \dots, a_{nn}^{(n-1)}$. При этом столбцы, использованные в качестве разрешающих, могут больше не рассматриваться.

Формулы м. ж. и. могут быть обобщены на случай блочных разрешающих элементов (в качестве разрешающего элемента рассматривается не элемент a_{rv} , а квадратная матрица A_{rv}). Пусть мы имеем матрицу $A=[A_{kl}]$, разбитую на блоки, где $k=1, \dots, \beta$; $l=1, \dots, s$

$$A = \begin{bmatrix} A_{11} & \dots & A_{1v} & \dots & A_{1s} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{r1} & \dots & A_{rv} & \dots & A_{rs} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{\beta 1} & \dots & A_{\beta v} & \dots & A_{\beta s} \end{bmatrix}.$$

Один шаг *блочных м. ж. и.* будет заключаться в преобразовании матрицы A в матрицу A' по формулам, аналогичным (1.3):

$$\left. \begin{aligned} A'_{kl} &= A_{kl} - A_{kv} A_{rv}^{-1} A_{rl}, & k \neq r, l \neq v, \\ A'_{rl} &= A_{rv}^{-1} A_{rl}, & l \neq v; \quad A'_{kv} = -A_{kv} A_{rv}^{-1}, & k \neq r, \\ A'_{rv} &= A_{rv}^{-1}. \end{aligned} \right\} \quad (1.4)$$

Матрица A_{rv} образуется из элементов, находящихся на пересечении разрешающих строк и столбцов. Например, при решении системы (1.2) в качестве матрицы A_{rv} можно рассматривать всю матрицу A размера $n \times n$. Отсюда очевидно, что при решении системы (1.2) по формулам (1.4) на месте матрицы A будет получена матрица A^{-1} , а на месте вектора B — вектор $X = B' = A^{-1}B$. Модифицированные жордановы исключения часто используются в линейном программировании для перехода от одной вершины выпуклого многогранного множества к другой вершине.

Общая задача линейного программирования, заданная в произвольной форме записи, имеет вид

$$\left. \begin{aligned} L &= \sum_{j=1}^n c_j x_j \rightarrow \max (\min) \\ \text{при} \quad \sum_{j=1}^n a_{ij} x_j &\{ \leq, =, \geq \} b_i, \quad i = 1, \dots, m. \end{aligned} \right\} \quad (1.5)$$

Часто используется также запись задачи линейного программирования в канонической форме:

$$\left. \begin{aligned} L &= \sum_{j=1}^n c_j x_j \rightarrow \max \\ \text{при} \end{aligned} \right\} \quad (1.6)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m, \quad (1.7)$$

$$x_j \geq 0, \quad j = 1, \dots, n, \quad (1.8)$$

где $m < n$. От задачи (1.5) нетрудно перейти к задаче (1.6) — (1.8) путем введения дополнительных переменных.

Вектор $X = [x_j]$, удовлетворяющий ограничениям задачи линейного программирования, называется ее *планом*.

План $X=[x_j]$, который соответствует вершине многогранного множества, образованного ограничениями, называется *опорным планом*. Отсюда определение опорного плана можно записать так: план $X=[x_1, \dots, x_n]$ называется *опорным* планом задачи (1.5), если он удовлетворяет как точным равенствам не менее, чем n линейно независимым ограничениям этой задачи. Для задачи, записанной в канонической форме (1.6) — (1.8), план будет опорным, если векторы A_j условий (1.7), отвечающие его положительным составляющим, линейно независимы.

Базисом n -мерного пространства называется система n линейно независимых n -мерных векторов. Любой вектор n -мерного пространства можно представить в виде линейной комбинации векторов базиса (выразить через вектора базиса). Это представление наиболее удобно сделать, когда вектора базиса единичные: $(1, 0, 0, \dots, 0)$, $(0, 1, 0, \dots, 0)$, $\dots$, $(0, 0, \dots, 1)$.

Рассмотрим задачу линейного программирования, записанную в канонической форме (1.6) — (1.8). Предположим, что левые части ограничений данной задачи линейно независимы. Опорный план задачи (1.6) — (1.8), по определению, может содержать не более, чем m положительных компонент $x_j > 0$ (так как среди n условий $x_j \geq 0$ должно быть не менее, чем $(n - m)$ равенств $x_j = 0$). Поэтому имеет смысл рассматривать m -мерное пространство, а не n -мерное ($m < n$).

Базисом опорного плана $X=[x_1, x_2, \dots, x_n]$ задачи (1.6) — (1.8) называют систему m линейно независимых векторов условий A_j , включающую, в частности, все те A_j , для которых $x_j > 0$. Мы говорим здесь «в частности», имея в виду, что количество векторов A_j , для которых $x_j > 0$, может быть и меньше, чем m . Это будет при так называемом *вырожденном* опорном плане.

Опорный план $X=[x_1, x_2, \dots, x_n]$ называется *невырожденным*, если положительны все его базисные компоненты (базисные переменные — это переменные x_j , соответствующие векторам базиса).

Задача линейного программирования называется *невырожденной*, если все ее опорные планы невырождены.

Оптимальным планом или *решением* задачи линейного программирования называется план, максимизирующий линейную форму (1.6).

§ 2. Теоретические основы формирования методов линейного программирования

1. Каждой задаче линейного программирования можно поставить в соответствие некоторую другую, так называемую *двойственную* задачу. Пусть имеется задача линейного программирования в произвольной форме записи:

$$L = \sum_{j=1}^n c_j x_j \rightarrow \max, \quad (2.1)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, \dots, m_1, \quad m_1 \leq m, \quad (2.2)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = m_1 + 1, \dots, m, \quad (2.3)$$

$$x_j \geq 0, \quad j = 1, \dots, n_1; \quad n_1 \leq n. \quad (2.4)$$

Задача (2.1) — (2.4) эквивалентна задаче (1.5). Это очевидно, так как условие минимизации целевой функции легко свести к условию максимизации (2.1) умножением (-1), ограничения неравенства со знаком $\geq$ также сводятся к неравенствам (2.2) со знаком $\leq$ умножением на (-1), а условия (2.4) $x_j \geq 0$ являются частным случаем ограничений задачи (1.5). Задача, двойственная по отношению к задаче (2.1) — (2.4), имеет вид

$$L_y = \sum_{i=1}^m b_i y_i \rightarrow \min, \quad (2.5)$$

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, \dots, n_1, \quad n_1 \leq n, \quad (2.6)$$

$$\sum_{i=1}^m a_{ij} y_i = c_j, \quad j = n_1 + 1, \dots, n, \quad (2.7)$$

$$y_i \geq 0, \quad i = 1, \dots, m_1, \quad m_1 \leq m. \quad (2.8)$$

В свою очередь задача (2.1) — (2.4) будет двойственной по отношению к задаче (2.5) — (2.8). Поэтому задачи (2.1) — (2.4) и (2.5) — (2.8) называют *двойственной* (или *взаимосопряженной*) парой задач.

Как видно из приведенных формулировок, правило составления двойственной задачи будет заключаться в следующем:

1) матрица ограничений двойственной задачи образуется путем транспонирования матрицы прямой задачи;

2) вектор коэффициентов целевой функции двойственной задачи равен вектору правой части ограничений прямой задачи, а вектор правой части ограничений двойственной задачи равен вектору коэффициентов целевой функции прямой задачи;

3) существует соотношение между переменными прямой задачи и ограничениями двойственной: ограничение двойственной задачи является неравенством, если на переменную прямой задачи с тем же индексом наложено требование неотрицательности; в противном случае это ограничение является равенством; аналогично связаны между собой ограничения прямой задачи и переменные двойственной;

4) при переходе от прямой задачи к двойственной знаки неравенств в ограничениях меняются на противоположные; максимизация целевой функции прямой задачи (или минимизация, если задача (2.5) — (2.8) прямая) заменяется минимизацией (максимизацией) целевой функции двойственной задачи.

Для задачи линейного программирования, записанной в канонической форме (1.6) — (1.8), двойственная задача примет вид

$$L_y = \sum_{i=1}^m b_i y_i \rightarrow \min, \quad (2.9)$$

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, \dots, n \quad (2.10)$$

(здесь $n_1 = n$, $m_1 = 0$).

2. Связь между решениями прямой и двойственной задачи устанавливается на основе теорем двойственности.

Основная (первая) теорема двойственности. *Если одна из задач двойственной пары имеет решение, то другая задача также разрешима. При этом для любых оптимальных планов $X^* = [x_1^*, x_2^*, \dots, x_n^*]$ и $Y^* = [y_1^*, y_2^*, \dots, y_m^*]$ этих задач справедливо равенство*

$$\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*.$$

Для разрешимости задачи линейного программирования необходимо и достаточно, чтобы множество планов задачи

было не пусто и значение целевой функции было ограничено сверху (для задачи максимизации) на множестве планов. Если же у одной из задач двойственной пары целевая функция не ограничена, то множество планов двойственной задачи будет пустым. А если у одной из задач двойственной пары множество планов пусто, то двойственная задача либо имеет неограниченную целевую функцию либо множество ее планов также пусто.

Разрешимость задачи линейного программирования может быть установлена на основе следующего следствия из первой теоремы двойственности: *для разрешимости одной из задач двойственной пары необходимо и достаточно, чтобы каждая из задач имела хотя бы один план.*

Важное следствие из первой теоремы двойственности заключается также в следующем: *для того чтобы планы $X=[x_1, \dots, x_n]$ и $Y=[y_1, \dots, y_m]$ задач двойственной пары были оптимальными, необходимо и достаточно, чтобы значения целевых функций прямой и двойственной задачи были при этих планах равными.*

Отсюда следует, что нахождение решения пары двойственных задач эквивалентно нахождению векторов X и Y , удовлетворяющих всем ограничивающим условиям прямой и двойственной задач и условию

$$\sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i y_i. \quad (2.11)$$

Например, для задачи линейного программирования в произвольной форме записи это означает следующее: решение задач (2.1) — (2.4) и (2.5) — (2.8) эквивалентно нахождению частного решения системы, состоящей из линейных неравенств (2.2), (2.4), (2.6), (2.8) и равенств (2.3), (2.7), (2.11). Задача в канонической форме записи сводится к системе (1.7), (1.8), (2.10), (2.11). В матричном виде эта система запишется так:

$$AX = B, \quad (2.12)$$

$$X \geq 0, \quad (2.13)$$

$$A^T Y \geq C, \quad (2.14)$$

$$CX = BY. \quad (2.15)$$

Вторая теорема двойственности. Прежде чем сформулировать теорему, введем некоторые определения.

Пусть рассматривается пара двойственных задач (2.1) — (2.4) и (2.5) — (2.8). Условия систем (2.2) и (2.8)

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad y_i \geq 0 \quad (2.16)$$

при фиксированном значении индекса строки i прямой задачи называют *строчной парой двойственных условий* взаимосопряженных задач (2.1) — (2.4) и (2.5) — (2.8). Условия систем (2.4) и (2.6)

$$\sum_{i=1}^m a_{ij}y_i \geq c_j, \quad x_j \geq 0 \quad (2.17)$$

при фиксированном значении индекса столбца j прямой задачи называют *столбцовой парой двойственных условий*.

Условие задачи называют *свободным*, если существует решение, при котором данное условие выполняется как строгое неравенство. Условие задачи называют *закрепленным*, если на всех решениях данной задачи оно удовлетворяется как равенство.

Вторая теорема двойственности формулируется следующим образом: *если взаимосопряженные задачи разрешимы, то в каждой паре их двойственных условий (столбцовых или строчных) одно условие свободное, другое закрепленное.*

Другими словами, решения $X^* = [x_1^*, x_2^*, \dots, x_n^*]$ и $Y^* = [y_1^*, y_2^*, \dots, y_m^*]$ пары двойственных задач удовлетворяют условиям

$$y_i^* \left(\sum_{j=1}^n a_{ij}x_j^* - b_i \right) = 0, \quad i = 1, \dots, m, \quad (2.18)$$

$$x_j^* \left(\sum_{i=1}^m a_{ij}y_i^* - c_j \right) = 0, \quad j = 1, \dots, n. \quad (2.19)$$

Условия (2.18) и (2.19) называются условиями *дополняющей нежесткости*, так как они обеспечивают выполнение неравенства («нежесткости») хотя бы в одном из двух условий системы (2.16) или (2.17). Для задачи в канонической форме записи (1.5) и двойственной к ней задачи (2.9), (2.10) условий (2.16) не имеется, и условия дополняющей нежесткости состоят только из системы (2.19). Эти условия связывают между собой столбцовые пары условий (2.17) в оп-

тимальной точке. Условия (2.19) означают, что в оптимальной точке

$$\sum_{i=1}^m a_{ij}y_i = c_j \quad \text{для } x_j \neq 0$$

или, в векторной форме,

$$(YA_j) = c_j \quad \text{для } x_j \neq 0. \quad (2.20)$$

Вернемся теперь к записи канонической задачи линейного программирования через системы равенств и неравенств (2.12) — (2.15). Нетрудно показать, что равенство (2.15) является следствием условий (2.20). Отсюда — решение задачи линейного программирования эквивалентно нахождению частного решения системы (2.12) — (2.14), (2.20).

3. Четыре системы линейных равенств и неравенств (2.12) — (2.15) (или (2.12) — (2.14), (2.20)) определяют четыре группы методов линейного программирования. В каждой группе методов предполагается движение к оптимуму по точкам X , Y , удовлетворяющим трем системам равенств и неравенств из данных четырех систем. Оставшаяся система используется в качестве критерия оптимальности. При выполнении и этой четвертой системы будет получен оптимум.

В группе методов, объединяемых под названием *симплекс-метода* (или *метода последовательного улучшения плана*) используется движение к оптимуму по точкам, удовлетворяющим условиям (2.12), (2.13), (2.20). Условия системы (2.14) служат критерием оптимальности.

В *двойственном симплекс-методе* (или *методе последовательного уточнения оценок*) движение к оптимуму происходит по точкам, удовлетворяющим условиям (2.12), (2.14), (2.20). Критерием оптимальности служат условия (2.13).

В *методе одновременного решения прямой и двойственной задачи* (или *методе последовательного сокращения невязок*) используются точки, удовлетворяющие системам условий (2.13), (2.14), (2.20). Условия (2.12) выполняют роль критерия оптимальности.

И, наконец, в методах четвертой группы (*метод одновременного улучшения планов прямой и двойственной задачи*, *метод двусторонних оценок*) условия (2.12) — (2.14) выполняются в процессе решения, а условия (2.15) используются в качестве критерия оптимальности.

Рассмотрим более подробно различные формулировки указанного критерия оптимальности для симплекс-метода. Сделаем предварительно некоторые замечания. Мы уже отмечали, что при использовании симплекс-метода для решения задачи (1.6) — (1.8) происходит движение к оптимуму по точкам, удовлетворяющим ограничениям данной прямой задачи.

Одно из свойств задач линейного программирования заключается в следующем: целевая функция задачи достигает своего оптимума в крайней точке выпуклого многогранного множества, образованного ограничениями. Поэтому при нахождении решения можно просматривать не все точки многогранного множества, а только крайние точки. Это хорошо иллюстрируется геометрически для двумерной задачи (см., например, рис. 3.1 в главе III).

Из определений, приведенных в предыдущем параграфе, следует, что каждой крайней точке соответствует некоторый опорный план задачи, причем с данным опорным планом связывается некоторый базис — система m линейно независимых векторов из n векторов A_j . Отсюда видно, что в системе (2.20) (которой также должно удовлетворять решение прямой и двойственной задачи) будет в невырожденном случае m столбцов A_j . По сути дела, решение задачи заключается именно в выборе m векторов из общего числа n векторов A_j таких, что соответствующие значения переменных доставляют оптимум целевой функции. Критерий оптимальности нужен для того, чтобы вести этот выбор целенаправленно, следя за монотонным возрастанием (или монотонным убыванием) целевой функции.

Итак, пусть мы имеем некоторую систему (2.20) из m векторов A_j . Эта система представляет собой систему m линейных алгебраических уравнений с m неизвестными. Переменные, полученные при решении данной системы, могут не удовлетворять ограничениям (2.14) двойственной задачи. Поэтому данные переменные не образуют плана двойственной задачи и обозначаются через λ_i . Условие

$\sum_{i=1}^m a_{ij} \lambda_i - c_j$ обозначим через Δ_j . В оптимальной точке усло-

вия (2.14) удовлетворяются (все $\Delta_j \geq 0$) и переменные λ_i совпадают с оптимальными значениями переменных двойственной задачи $\lambda_i^* = y_i^*$. Таким образом, для m базисных пере-

менных x_j , по аналогии с (2.20), имеем

$$\Delta_j = 0 \text{ или } \sum_{i=1}^m a_{ij} \lambda_i = c_j. \quad (2.21)$$

Критерием оптимальности служит знак оценок Δ_j для $n - m$ внебазисных переменных x_j :

$$\Delta_j = \sum_{i=1}^m a_{ij} \lambda_i - c_j \geq 0. \quad (2.22)$$

Если все неравенства системы (2.22) имеют знак ≥ 0 , значит, получен оптимум. В противном случае переменная x_j , имеющая минимальную отрицательную оценку Δ_j , заменяет некоторую базисную переменную x_i , и вновь проверяются знаки оценок (2.22). Переход от одного опорного плана к другому вызывает возрастание целевой функции (1.6) (в вырожденном случае целевая функция не убывает).

Оценки $\Delta = [\Delta_1, \Delta_2, \dots, \Delta_n]$ называют *оценками векторов условий* A_j относительно данного базиса. Переменные $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_m]$ называют *оценками условий* задачи относительно данного базиса. (Переменные λ_i также иногда называют *симплекс-множителями, двойственными оценками* и др.)

Обозначим через A_b матрицу векторов базиса: $A_b = [A_{b_1}, A_{b_2}, \dots, A_{b_m}]$. Через A_v обозначим матрицу остальных внебазисных векторов: $A_v = [A_{v_1}, A_{v_2}, \dots, A_{v_{n-m}}]$. Аналогично обозначим через C_b и C_v векторы коэффициентов целевой функции, соответственно для базисных и внебазисных векторов: $C_b = [c_{b_1}, c_{b_2}, \dots, c_{b_m}]$; $C_v = [c_{v_1}, c_{v_2}, \dots, c_{v_{n-m}}]$. Тогда в матричных обозначениях оценка (2.22) примет вид

$$\Delta_v = \Lambda A_v - C_v \geq 0, \quad (2.23)$$

где $\Delta_v = (\Delta_{v_1}, \Delta_{v_2}, \dots, \Delta_{v_{n-m}})$, а $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_m]$ — вектор-строка переменных, полученных из решения системы (2.21), которая в матричном виде запишется как

$$\Lambda A_b = C_b. \quad (2.24)$$

Выражения (2.23), (2.24) представляют собой одну форму критерия оптимальности в симплекс-методе. Эта форма

используется в алгоритме симплекс-метода, называемом *методом обратной матрицы* (или *модифицированным симплекс-методом*). Название становится понятным, если учесть, что в данном случае используется обратная базисная матрица, в частности, для вычисления Λ из (2.24):

$$\Lambda = C_6 A_6^{-1}. \quad (2.25)$$

Часто используется также другая форма записи критерия оптимальности. Эта форма легко может быть получена, если подставить значения Λ из (2.25) в (2.23)

$$\Delta_B = C_6 A_6^{-1} A_B - C_B$$

и учесть, что

$$A_6^{-1} A_B = A'_B, \quad (2.26)$$

где A'_B — матрица, составленная из внебазисных векторов, выраженных через векторы данного базиса. Итак, другая форма признака оптимальности записывается как

$$\Delta_B = C_6 A'_B - C_B \geq 0 \quad (2.27)$$

или, отказавшись от матричных обозначений, и учтя, что для базисных столбцов $\Delta_j = 0$, получим

$$\Delta_j = \sum_{i=1}^m c_{6i} a'_{ij} - c_j \geq 0, \quad j = 1, \dots, n. \quad (2.28)$$

Критерий оптимальности (2.28) был предложен еще в первоначальном варианте симплекс-метода. Алгоритм с использованием оценок (2.28) называют иногда *первым алгоритмом* симплекс-метода, а алгоритм с использованием оценок (2.22) — *вторым алгоритмом* симплекс-метода. Алгоритмы, использующие формулы (2.28) или (2.22), могут быть записаны также для других групп методов линейного программирования.

Итак, в данном параграфе мы рассмотрели основы формирования различных методов линейного программирования. Для этого сначала были приведены правила составления двойственной задачи. Затем, записав первую теорему двойственности, мы пришли к тому, что решение прямой и двойственной задачи эквивалентно нахождению частного решения

системы (2.12) — (2.15) линейных равенств и неравенств, в которую входят ограничения прямой и двойственной задач, а также условие равенства значений целевых функций прямой и двойственной задач. Далее, записав вторую теорему двойственности, мы убедились, что последнее равенство в указанной системе можно не рассматривать, если выполняются условия дополняющей нежесткости. Теперь мы имели все необходимое для классификации методов линейного программирования. Методы решения задачи линейного программирования в канонической форме записи разделились на четыре группы в соответствии с тем, какая из четырех подсистем условий системы (2.12) — (2.15) не выполняется в процессе движения к оптимуму. Эта подсистема условий используется в качестве критерия оптимальности. В заключение параграфа рассматриваются различные формы записи критерия оптимальности при использовании симплекс-метода.

Заметим, что порядок изложения материала данного параграфа в некоторой степени отстает от обычно принятого порядка изложения рассматриваемых теоретических вопросов. Поэтому, например, мы и получаем сначала критерий оптимальности для модифицированного симплекс-метода, а потом уже для обычного симплекс-метода. Такая методика изложения нам представляется более стройной, так как классификация как методов, так и алгоритмов решения задач линейного программирования в данном случае исходит из одного и того же факта представления задачи линейного программирования в виде системы линейных равенств и неравенств.

§ 3. Вычислительные схемы

1. Изложение вычислительных схем решения задач линейного программирования начнем с рассмотрения алгоритмов симплекс-метода. Для простоты будем вести изложение на основе задачи линейного программирования в канонической форме записи (1.6) — (1.8), так как любая задача линейного программирования может быть сведена к данной форме.

Как отмечалось в предыдущем параграфе, для решения задачи (1.6) — (1.8) необходимо выбрать оптимальную

комбинацию m векторов A_j из общего числа n векторов. При этом сначала нужно выбрать комбинацию векторов A_j , при которой переменные x_j составляют опорный план задачи, а затем уже организовать движение по опорным планам к оптимальному плану.

Рассмотрим сначала *вопрос получения опорного плана*. Пусть мы имеем некоторую комбинацию m векторов A_j (и пусть σ — множество индексов этих векторов). Из определения опорного плана следует: для того чтобы данная комбинация векторов A_j соответствовала некоторому опорному плану задачи (1.6) — (1.8), нужно, чтобы все переменные x_j с индексами $j \notin \sigma$ были нулевыми, а значения переменных x_j , $j \in \sigma$ были неотрицательными при решении системы

$$\sum_{j \in \sigma} a_{ij} x_j = b_i, \quad i = 1, \dots, m. \quad (3.1)$$

Удобнее всего, если векторы A_j , $j \in \sigma$ являются единичными. Решение системы (3.1) тогда записывается сразу: $x_{\sigma_j} = b_i$, где $j = i$. Это очень удобно также для осуществления перехода от одного опорного плана к другому (в этом мы убедимся в дальнейшем). Если же векторы A_j не единичные, то все равно при решении системы (3.1) их пришлось бы преобразовать в единичные.

Итак, в симплекс-методе, мы все время оперируем с базисом, составленным из единичных векторов. При отыскании исходного опорного плана единичный базис составляется либо из векторов A_j матрицы ограничений задачи, либо из специально добавленных «искусственных» единичных векторов (*метод искусственного базиса*).

Для определения исходного опорного плана может решаться отдельная вспомогательная задача, в которой осуществляется минимизация значений искусственных переменных (т. е. переменных x_j , $j > n$, добавленных в задачу для создания исходного единичного базиса). Существует много различных способов нахождения исходного опорного плана.

Мы рассмотрим один из наиболее применимых на практике способов, в котором этап нахождения исходного опорного плана совмещается с этапом отыскания оптимального плана. Пусть решается задача линейного программирования

Пусть для решения задачи используется первый алгоритм симплекс-метода. Оценки Δ_j векторов условий относительно исходного базиса записываются по формуле (2.28):

$$\Delta_j = \sum_{i=1}^m c_{n-m+i} a_{ij} - c_j, \quad j = 1, \dots, n.$$

В данном случае $c_{6i} = c_{n-m+i}$, $a'_{ij} = a_{ij}$. Очевидно, что $\Delta_{n-m+i} = 0$, $i = 1, \dots, m$. Представим теперь всю исходную информацию в виде таблицы П.1. Если имеются оценки $\Delta_j < 0$, значит, данный опорный план не является оптимальным и осуществляется переход к другому опорному плану. Для этого в число базисных векторов включается вектор $A_v = [a_{iv}]$, имеющий минимальную отрицательную оценку Δ_j , а из базиса выводится вектор A_{6_r} , для которого

$$\theta_r = \min_{a_{iv} > 0} \frac{b_i}{a_{iv}}. \quad (3.6)$$

Если для столбца A_j с оценкой $\Delta_j < 0$ не имеется элементов $a_{iv} > 0$, значит, целевая функция задачи неограничена на рассматриваемом множестве планов. Вектор A_v после данной итерации становится единичным вектором базиса, а все внебазисные векторы выражаются через векторы базиса. Матрица таблицы П.1 при этом преобразуется следующим образом:

$$\left. \begin{aligned} a'_{ij} &= a_{ij} - \frac{a_{iv} a_{rj}}{a_{rv}}, & i \neq r, & j \neq v, \\ a'_{rj} &= \frac{a_{rj}}{a_{rv}}, & j \neq v, & a'_{iv} = 0, & i \neq r, & a'_{rv} = 1, \\ b'_i &= b_i - \frac{a_{iv} b_r}{a_{rv}}, & i \neq r, & b'_r &= \frac{b_r}{a_{rv}}, \\ \Delta'_j &= \Delta_j - \frac{\Delta_v a_{rj}}{a_{rv}}, & j \neq v, & \Delta'_v &= 0, & L' &= L - \frac{\Delta_v b_r}{a_{rv}}, \end{aligned} \right\} \quad (3.7)$$

где v — индекс разрешающего столбца ($j=v$); r — индекс разрешающей строки ($i=r$). Например, если в базис вводится вектор A_2 , а выводится из базиса вектор A_{n-m+1} , то матрица преобразуется к виду, изображенному в таблице П.2. Разрешающим элементом a_{rv} в данном случае является элемент a_{12} (r — позиция вектора A_{6_r} в единичной матрице базиса до проведения итерации). После перехода к новому

Таблица П.1
Табличное представление исходной информации при решении задачи (3.5) первым алгоритмом симплекс-метода

№ строки	X_6	x_1	x_2	x_3	...	x_{n-m}	x_{n-m+1}	x_{n-m+2}	x_{n-m+3}	...	x_n	B
		A_1	A_2	A_3	...	A_{n-m}	A_{n-m+1}	A_{n-m+2}	A_{n-m+3}	...	A_n	
1	x_{n-m+1}	a_{11}	a_{12}	a_{13}	...	$a_{1, n-m}$	1	0	0	...	0	b_1
2	x_{n-m+2}	a_{21}	a_{22}	a_{23}	...	$a_{2, n-m}$	0	1	0	...	0	b_2
3	x_{n-m+3}	a_{31}	a_{32}	a_{33}	...	$a_{3, n-m}$	0	0	1	...	0	b_3
.	.	.	.	.	...	.	.	.	.	...	.	.
.	.	.	.	.	...	.	.	.	.	...	.	.
m	x_n	a_{m1}	a_{m2}	a_{m3}	...	$a_{m, n-m}$	0	0	0	...	1	b_m
$m+1$		Δ_1	Δ_2	Δ_3	...	Δ_{n-m}	0	0	0	...	0	L

Первая итерация решения

Таблица П.2

№№ строк	X6	$x_1 \quad x_2 \quad x_3 \quad \dots \quad x_{n-m}$					$x_{n-m+1} \quad x_{n-m+2} \quad x_{n-m+3} \quad \dots \quad x_n$					B'
		A'_1	A'_2	A'_3	$\dots$	A'_{n-m}	A'_{n-m+1}	A'_{n-m+2}	A'_{n-m+3}	$\dots$	A'_n	
1	x_2	a'_{11}	1	a'_{13}	$\dots$	$a'_{1, n-m}$	$\frac{1}{a_{12}}$	0	0	$\dots$	0	b'_1
2	x_{n-m+2}	a'_{21}	0	a'_{23}	$\dots$	$a'_{2, n-m}$	$-\frac{a_{22}}{a_{12}}$	1	0	$\dots$	0	b'_2
3	x_{n-m+3}	a'_{31}	0	a'_{33}	$\dots$	$a'_{3, n-m}$	$-\frac{a_{32}}{a_{12}}$	0	1	$\dots$	0	b'_3
.	.	.	.	.	$\dots$	.	.	.	.	$\dots$	.	.
.	.	.	.	.	$\dots$	.	.	.	.	$\dots$	.	.
.	.	.	.	.	$\dots$	.	.	.	.	$\dots$	.	.
m	x_n	a'_{m1}	0	a'_{m3}	$\dots$	$a'_{m, n-m}$	$-\frac{a_{m2}}{a_{12}}$	0	0	$\dots$	1	b'_m
$m+1$		Δ'_1	0	Δ'_3	$\dots$	Δ'_{n-m}	$-\frac{\Delta_2}{a_{12}}$					L'

базису в качестве базисных переменных будут переменные $x_2, x_{n-m+2}, x_{n-m+3}, \dots, x_n$. Значения базисных переменных будут $x_{b_r} = b'_r$, т. е. $x_2 = b'_1, x_{n-m+2} = b'_2, x_{n-m+3} = b'_3, \dots, x_n = b'_m$.

Оптимальность полученного опорного плана вновь проверяется по знаку оценок Δ'_j векторов условия. Если имеются отрицательные оценки Δ'_j , то переходим к следующему

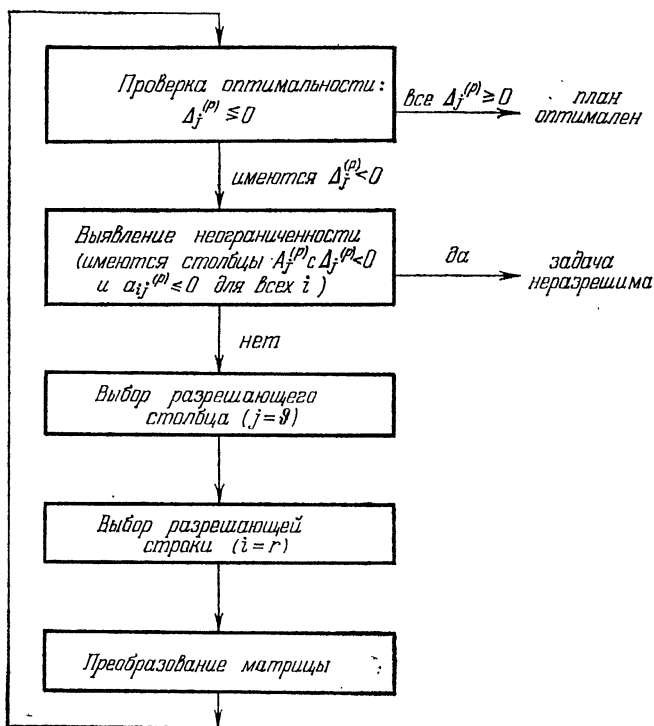


Рис. П.1*). Блок-схема первого алгоритма симплекс-метода.

опорному плану и так далее. Процедура перехода на каждой итерации аналогична процедуре, описанной выше. Только на p -й итерации вместо элементов a'_{ij} , b'_i , Δ'_j будут элементы

*) См. исправления рис. П.1, П.2, П.3 на стр. 344.

$a_{ij}^{(p+1)}$, $b_i^{(p+1)}$, $\Delta_j^{(p+1)}$, а вместо элементов a_{ij} , b_i , Δ_j будут элементы $a_{ij}^{(p)}$, $b_i^{(p)}$, $\Delta_j^{(p)}$.

Если на некоторой итерации окажется, что среди элементов $a_{ij}^{(p)}$ вводимого в базис вектора не имеется положительных (или все $a_{ij}^{(p)} \leq 0$ для столбца $A_j^{(p)}$ с оценкой $\Delta_j^{(p)} < 0$), то это будет означать неограниченность сверху целевой функции задачи на множестве ее планов. Блок-схема первого алгоритма симплекс-метода приведена на рис. П.1.

3. При решении задачи можно не записывать матрицу единичного базиса. Тогда становится компактнее форма записи таблицы симплекс-метода. В этом случае вектор

Таблица П.3

Табличное представление исходной информации при решении задачи (3.5) первым алгоритмом симплекс-метода с использованием модифицированных жордановых исключений

№№ строк	X_B X_6	$x_1 \quad x_2 \quad x_3 \quad \dots \quad x_{n-m}$					B
1	x_{n-m+1}	a_{11}	a_{12}	a_{13}	$\dots$	$a_{1, n-m}$	b_1
2	x_{n-m+2}	a_{21}	a_{22}	a_{23}	$\dots$	$a_{2, n-m}$	b_2
3	x_{n-m+3}	a_{31}	a_{32}	a_{33}	$\dots$	$a_{3, n-m}$	b_3
.	.	.	.	.	$\dots$	.	.
.	.	.	.	.	$\dots$	.	.
.	.	.	.	.	$\dots$	.	.
m	x_n	a_{m1}	a_{m2}	a_{m3}	$\dots$	$a_{m, n-m}$	b_m
$m+1$		Δ_1	Δ_2	Δ_3	$\dots$	Δ_{n-m}	L

$A_{\bar{r}}$, выведенный из базиса, записывается на месте вектора A_v , введенного в базис на данной итерации. Соответственно оценка Δ_j выведенного из базиса вектора записывается на месте оценки Δ_v вектора, введенного в базис. Элементы матрицы преобразуются на каждой итерации по формулам (1.3) модифицированных жордановых исключений [39]. Элементы вектора B и оценки Δ_j преобразуются аналогично — можно просто считать, что эти элементы входят в состав матрицы, преобразуемой по формулам (1.3). Исходная

таблица симплекс-метода имеет в этом случае вид, представленный таблицей П.3. Слева от матрицы по вертикали записываются базисные переменные, вверху матрицы по горизонтали записываются небазисные переменные. На каждой итерации одна из базисных переменных меняется местами с небазисной переменной. При этом, небазисная переменная x_v , вводимая в число базисных переменных, определяет разрешающий столбец, а базисная переменная x_b , выводимая из числа базисных, определяет разрешающую строку. Разрешающий элемент a_{rv} находится на пересечении разрешающих строки и столбца.

Т а б л и ц а П.4

Первая итерация решения при использовании модифицированных жордановых исключений

№№ строк	X_B X_6	X_N $x_1 \ x_{n-m+1} \ x_3 \ \dots \ x_{n-m}$					B'
		x_1	x_{n-m+1}	x_3	$\dots$	x_{n-m}	
1	x_2	a'_{11}	a'_{12}	a'_{13}	$\dots$	$a'_{1, n-m}$	b'_1
2	x_{n-m+2}	a'_{21}	a'_{22}	a'_{23}	$\dots$	$a'_{2, n-m}$	b'_2
3	x_{n-m+3}	a'_{31}	a'_{32}	a'_{33}	$\dots$	$a'_{3, n-m}$	b'_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	x_n	a'_{m1}	a'_{m2}	a'_{m3}	$\dots$	$a'_{m, n-m}$	b'_m
$m+1$		Δ'_1	Δ'_2	Δ'_3	$\dots$	Δ'_{n-m}	L'

Например, если так же, как при составлении таблицы П.2, считать, что в базис вводится вектор A_2 , а выводится из базиса вектор A_{n-m+1} , то матрица преобразуется к виду, изображенному в таблице П.4. Аналогия между таблицами П.4 и П.2 очевидна, если учесть, что из формул (1.3)

$$a'_{12} = \frac{1}{a_{12}}, \quad a'_{22} = -\frac{a_{22}}{a_{12}}, \quad \dots, \quad a'_{m2} = -\frac{a_{m2}}{a_{12}}, \quad \Delta'_2 = -\frac{\Delta_2}{a_{12}}.$$

Заметим, что на p -й итерации в качестве элементов a'_{ij} ,

b'_i , Δ'_j будут элементы $a_{ij}^{(p+1)}$, $b_i^{(p+1)}$, $\Delta_j^{(p+1)}$, а в качестве элементов a_{ij} , b_i , Δ_j будут элементы $a_{ij}^{(p)}$, $b_i^{(p)}$, $\Delta_j^{(p)}$.

4. Рассмотрим теперь процедуру решения задачи при использовании *модифицированного симплекс-метода* (второй алгоритм симплекс-метода).

В этом случае оценки Δ_j векторов условий находятся по формуле (2.22), где значение $\Lambda = [\lambda_i]$ определяется из формулы (2.25) через обратную базисную матрицу A_6^{-1} . Нетрудно убедиться в том, что матрица A_6^{-1} в таблицах симплекс-метода получается на каждой итерации на месте единичной матрицы исходного базиса. В таблицах П.1 и П.2 для иллюстрации матрица A_6^{-1} заключена в жирную рамку.

Отсюда уже ясно, что вместо того, чтобы преобразовывать на каждой итерации всю матрицу таблицы симплекс-метода, можно использовать преобразование только части матрицы, а именно той части, которая заключена в жирную рамку в таблицах П.1, П.2. Имея матрицу A_6^{-1} , мы найдем вектор Λ (см. (2.25)), а отсюда, по формуле (2.22), найдем оценки Δ_j . Заметим, что значения элементов λ_i на каждой итерации могут быть получены непосредственно при преобразовании матрицы A_6^{-1} в $(m+1)$ -й строке таблицы. Для этого достаточно записать на первой итерации под единичной матрицей A_6 $(m+1)$ -ю строку, состоящую из элементов $\lambda_i = c_{6_i}$ (см. (2.25)).

Заметим также, что при преобразовании только указанной части матрицы мы не будем иметь на каждой p -й итерации коэффициентов $a_{ij}^{(p)}$ выражения небазисных векторов через векторы базиса. Однако нам нужно иметь это выражение для вектора A_v , вводимого в базис, так как отсюда определяется вектор A_{6_r} , выводимый из базиса:

$$\Theta_r = \min_{a_{iv}^{(p)} > 0} \frac{b_i^{(p)}}{a_{iv}^{(p)}}.$$

Выражение вектора A_v через векторы базиса вычисляется следующим образом (см. (2.26)):

$$A_v^{(p)} = A_6^{-1} A_v.$$

Таким образом, в модифицированном симплекс-методе вместо преобразования всей матрицы, изображенной на таблице П.1, мы преобразуем только подматрицу A_6^{-1} , вектора

Λ , B , и на каждой итерации добавляем эту таблицу соответствующим вектором $A_v^{(p)}$ (см. таблицу П.5).

Очевидно, что модифицированный симплекс-метод имеет определенные преимущества перед первым алгоритмом симплекс-метода в том случае, когда уменьшение преобразуемой

Таблица П.5

Исходная запись преобразуемой части матрицы при решении задачи (3.5) модифицированным симплекс-методом

№№ строк	X_6	$A_v^{(1)}$	1 2 3 ... m	B
1	x_{n-m+1}	a_{1v}	1 0 0 ... 0	b_1
2	x_{n-m+2}	a_{2v}	0 1 0 ... 0	b_2
3	x_{n-m+3}	a_{3v}	0 0 1 ... 0	b_3
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
m	x_n	a_{mv}	0 0 0 ... 1	b_m
$m+1$		Δ_v	$\lambda_1 \lambda_2 \lambda_3 \dots \lambda_m$	L

части матрицы дает существенный эффект. А это будет тогда, когда число столбцов матрицы значительно больше числа строк ($n \gg m$).

Для иллюстрации мы привели в таблице П.5 исходную запись преобразуемой части матрицы при использовании модифицированного симплекс-метода. Вместо матрицы небазисных векторов $A_1, A_2, \dots, A_{n-m}$ (см. таблицу П.1) здесь записывается только один вектор $A_{v(1)}$, который вводится в базис на первой итерации. Если, например, мы вводим в базис вектор A_2 и выводим из базиса вектор A_{n-m+1} , то преобразуемая часть матрицы примет вид, изображенный в таблице П.6 (ср. с таблицей П.2). Блок-схема модифицированного симплекс-метода изображена на рис. П.2.

Заметим следующее. Из таблиц П.5 и П.6 видно, что при использовании модифицированного симплекс-метода преобразуемая часть матрицы может содержать большое число

единичных векторов. Поэтому часто практикуется такое представление матрицы A_6^{-1} , при котором записываются только неединичные векторы (*мультипликативное представление обратной матрицы*).

Таблица П.6

Первая итерация решения задачи
модифицированным симплекс-методом

№№ строк	X_6	$A'_{v(2)}$	1	2	3	...	m	B'
1	x_2	a'_{1v}	$\frac{1}{a_{12}}$	0	0	...	0	b'_1
2	x_{n-m+3}	a'_{2v}	$-\frac{a_{22}}{a_{12}}$	1	0	...	0	b'_2
3	x_{n-m+3}	a'_{3v}	$-\frac{a_{32}}{a_{12}}$	0	1	...	0	b'_3
.	.	.	.	.	.	...	.	.
.	.	.	.	.	.	...	.	.
m	x_n	a'_{mv}	$-\frac{a_{m2}}{a_{12}}$	0	0	...	1	b'_m
$m+1$		Δ'_v	λ'_1	λ'_2	λ'_3	...	λ'_m	L'

5. Рассмотрим теперь *двойственный симплекс-метод*. Как следует из классификации методов решения задачи линейного программирования, приведенной в предыдущем параграфе, при использовании двойственного симплекс-метода происходит движение к оптимуму по точкам, при которых значения $\Delta_j \geq 0$, однако значения x_j могут быть отрицательными. В оптимальной точке условия $x_j \geq 0$ выполняются — эти условия в данном случае являются критерием оптимальности.

Процедуру решения задачи двойственным симплекс-методом, так же как и в случае использования прямого симплекс-метода, можно разбить на два этапа — этап нахождения исходного опорного плана и этап нахождения оптимального опорного плана. Только в двойственном симплекс-методе требуется нахождение опорного плана

двойственной задачи (тогда удовлетворяются условия $\Delta_j \geq 0$). Двойственный симплекс-метод имеет определенные вычислительные преимущества перед прямым симплекс-методом именно в том случае, когда оказывается проще найти исходный опорный план двойственной задачи, нежели исходный опорный план прямой задачи. Мы не будем здесь останавливаться на различных способах нахождения исходного опорного плана двойственной задачи. Заметим только, что этот план легко записывается сразу, например, в том случае, если в задаче (3.2) все коэффициенты целевой функции неположительны: $c_j \leq 0$. Ограничения задачи, двойственной к (3.2), тогда имеют вид

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

и опорный план двойственной задачи будет

$$\begin{aligned} Y &= [y_i] = 0, & i &= 1, \dots, m, \\ Y^A &= [y_j^A] = -c_j, & j &= 1, \dots, n, \end{aligned}$$

где y_j^A — дополнительная переменная, обращающая в равенство j -е ограничение двойственной задачи.

Итак, пусть мы имеем исходный опорный план двойственной задачи. В таблице симплекс-метода в этом случае в строке Δ_j (т. е. в $(m+1)$ -й строке) все элементы будут неотрицательны. Если при этом столбец $X_6 = B = [b_i]$ также содержит все неотрицательные элементы, значит, данный опорный план является оптимальным. Если же имеются элементы $b_i < 0$, то мы либо убеждаемся в неразрешимости задачи либо осуществляем переход к другому опорному плану. Таким образом, если в симплекс-методе процедура нахождения оптимального плана заключалась в устранении отрицательных значений элементов строки $\Delta = [\Delta_j]$ при сохранении неотрицательности элементов столбца $B = [b_i]$, то в двойственном симплекс-методе данная процедура заключается в устранении отрицательных значений элементов столбца B при сохранении в то же время неотрицательности элементов строки Δ .

Ход вычислений при переходе от одного опорного плана двойственной задачи к другому опорному плану на p -й итерации будет следующим. Среди элементов $b_i^{(p)}$ столбца $B^{(p)}$ отыскивается элемент, имеющий минимальное отрица-

тельное значение $b_i^{(p)}$:

$$b_r^{(p)} = \min_{b_i^{(p)} < 0} b_i^{(p)}.$$

Этот элемент определяет разрешающую строку: $i=r$, т. е. определяет вектор $A_{\sigma_r}^{(p)}$, выводимый из базиса на данной

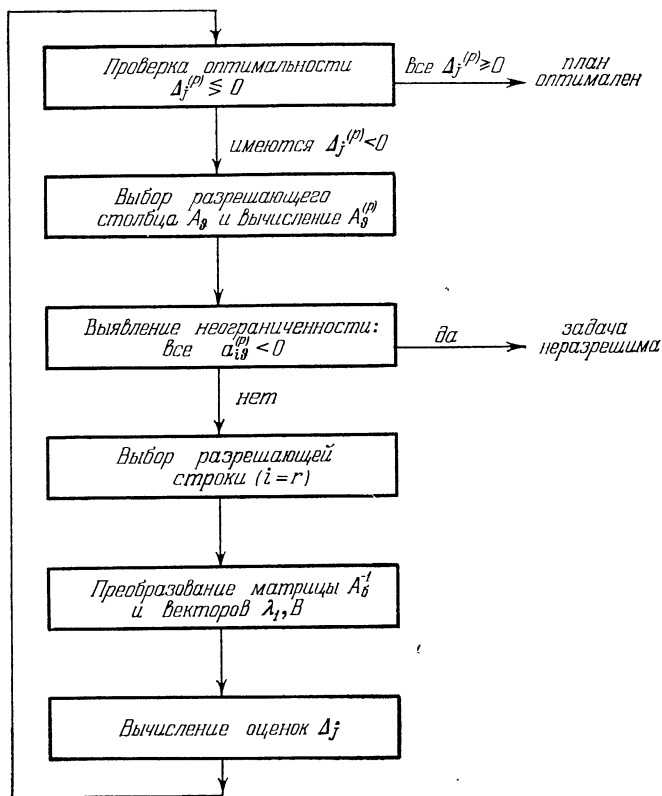


Рис. П.2. Блок-схема модифицированного симплекс-метода.

итерации. Вектор $A_{\sigma_r}^{(p)}$, вводимый в базис (т. е. разрешающий столбец) определяется из соотношения

$$\theta_v = \min_{a_{rj}^{(p)} < 0} -\frac{\Delta_j^{(p)}}{a_{rj}^{(p)}}.$$

Если в строке, имеющей $b_i^{(p)} < 0$, все элементы $a_{ij}^{(p)} \geq 0$,

то это будет означать неограниченность снизу целевой функции двойственной задачи на множестве ее планов (и, следовательно, противоречивость условий прямой задачи — см. § 2 данного приложения). После определения направляющей строки и направляющего столбца осуществляется преобразование матрицы по рекуррентным соотношениям типа (3.7)

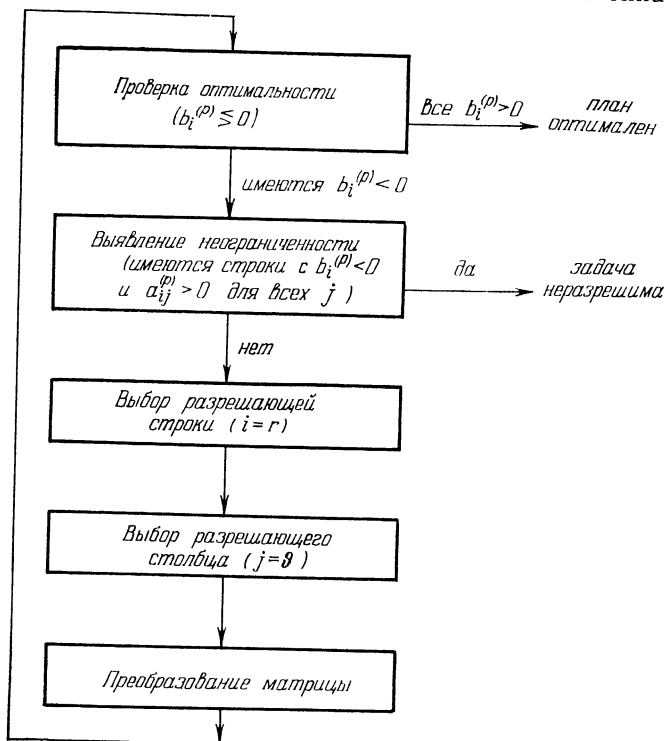


Рис. П.3. Блок-схема первого алгоритма двойственного симплекс-метода.

или (1.3) (в зависимости от формы записи таблицы: при записи в виде таблиц П.1, П.2 — формулы (3.7), при записи в виде таблиц П.3, П.4 — формулы (1.3)). И начинается следующая итерация алгоритма. Блок-схема двойственного симплекс-метода приведена на рис. П.3.

По аналогии с модифицированным симплекс-методом (второй алгоритм симплекс-метода) нетрудно построить второй алгоритм двойственного симплекс-метода,

ЛИТЕРАТУРА

1. Арбузова Н. И., Данилов В. Л., Об одной задаче стохастического линейного программирования и ее устойчивости. ДАН СССР, т. 162, № 1, 1965.
2. Атабеков Г. И., Теоретические основы электротехники, Госэнергоиздат, 1962.
3. Беллман Р., Динамическое программирование, ИЛ, 1960.
4. Беллман Р., Дрейфус С., Прикладные задачи динамического программирования, Изд-во «Наука», 1964.
5. Блекуэлл Д., Динамическое программирование с затухающим действием. «Математика» (Сборник переводов), т. 11 : 4, 1967.
6. Букан Дж., Кенигсберг Э., Научное управление запасами, Изд-во «Наука», 1967.
7. Вазов В., Форсайт Дж., Разностные методы решения дифференциальных уравнений в частных производных, ИЛ, 1963.
8. Варшавский В. И., Коллективное поведение автоматов. Экономика и математические методы, № 2, 1970.
9. Васильев В. В., Клепикова А. И., Тимошенко А. Г., Решение задач оптимального планирования на электрических моделях, «Наукова думка», Киев, 1966.
10. Вахитов Г. Г., Эффективные способы решения задач разработки нефтеводоносных пластов методом конечных разностей, Гостехиздат, 1963.
11. Верховский Б. С., Литвак Б. Л., Алгоритмы оптимизации ирригационных систем с водохранилищем. Применение математических методов и вычислительной техники в прогнозировании производственно-технического развития. Тезисы симпозиума, Кишинев, 1970.
12. Верховский Б. С., Спивак В. А., Оптимальное проектирование водохозяйственных систем и управляемые случайные процессы. В сб. «Применение ЭВМ и математических методов в ирригации», М., 1971.
13. Вильчевский Н. О., Разумихин Б. С., Механическая модель и метод решения общей задачи линейного программирования. «Автоматика и телемеханика», № 4, 1966, стр. 88—103.
14. Висков О. В., Ширяев А. Н., Об управлениях, приводящих к оптимальным стационарным решениям. Труды Матем. ин-та АН СССР им. В. А. Стеклова, т. XXI (1964), стр. 35—45.
15. Витенберг И. М., Задачи исследования операций и основные направления их решения средствами аналоговой вычислительной техники. В сб. «Аналого-вычислительная техника в организации производства и исследовании больших систем», М., 1970, стр. 4—13.

16. Воеводин В. В., Развитие методов решения задач алгебры в вычислительном центре университета. Вестник МГУ, сер. «Математика, механика», № 2, 1970.
17. Волконский В. А., Оптимальное планирование в условиях большой размерности (итеративные методы и принцип декомпозиции). Экономика и математические методы, т. I, вып. 2, 1965.
18. Вольф Р., Данциг Дж., Марковские цепи и линейное программирование. «Кибернетический сборник», новая серия, вып. 4, Изд-во «Мир», 1967.
19. Гантмахер Ф. Р., Теория матриц, Изд-во «Наука», 1966.
20. Гасс С., Линейное программирование, Физматгиз, 1961.
21. Гейл Д., Теория линейных экономических моделей, ИЛ, 1963.
22. Гольштейн Е. Г., О возможности расширения применимости частных методов линейного программирования. В сб. «Планирование и экономико-математические методы». (К семидесятилетию академика В. С. Немчинова), Изд-во «Наука», 1964, стр. 409—423.
23. Гольштейн Е. Г., Об одном общем подходе к решению задач линейного программирования блочной структуры. ДАН СССР, т. 166, вып. 5, 1966.
24. Гольштейн Е. Г., Теория двойственности в математическом программировании и ее приложения, «Наука», 1971.
25. Гольштейн Е. Г., Мовшович С. М., О некоторых современных направлениях в математическом программировании. Экономика и математические методы, т. III, вып. 5, 1967.
26. Гольштейн Е. Г., Юдин Д. Б., Новые направления в линейном программировании, Изд-во «Советское радио», 1966.
27. Грубов В. И., Решение задач линейного программирования большой размерности. Вторая зимняя школа по математическому программированию в Дрогобыче, М., 1969 (ротапринт).
28. Гурвиц Л., Программирование в линейных пространствах. В сб. «Исследования по линейному и нелинейному программированию», ИЛ, 1962, стр. 65—155.
29. Гутенмахер Л. И., Электрические модели, Изд-во АН СССР, 1949.
30. Гутенмахер Л. И., Новые аналоговые модели, Труды IV Всесоюзного совещания по автоматическому управлению (технической кибернетике), Рефераты докладов, т. I, 1968.
31. Данилов-Данильян В. И., Задачи большой размерности и итеративные методы оптимального планирования. Сборник алгоритмов и программ для решения на ЭЦВМ ЦСУ СССР, Изд-во «Статистика», 1967.
32. Данциг Дж., Линейное программирование, его обобщения и применения. Изд-во «Прогресс», 1966.
33. Даффин Р. Дж., Бесконечные программы. В сб. «Линейные неравенства и смежные вопросы», ИЛ, 1959, стр. 263—276.
34. Демидович Б. П., Марон И. А., Шувалова Э. З., Численные методы анализа, Физматгиз, 1963.
35. Деннис Дж. Б., Математическое программирование и электрические цепи, ИЛ, 1961.
36. Джевёлл В. С., Управляемые полумарковские процессы. «Кибернетический сборник», (новая серия) № 4, Изд-во «Мир», 1967.

37. З в я г и н а Р. А., Задача линейного программирования с блочно-диагональными матрицами. В сб. «Оптимальное планирование», II, Новосибирск, 1964.
38. З о й т е н д е й к Г., Методы возможных направлений, ИЛ, 1963.
39. З у х о в и ц к и й С. И., А в д е е в а Л. И., Линейное выпуклое программирование, Изд-во «Наука», 1967.
40. К а р л и н С., Математические методы в теории игр, программировании и экономике, Изд-во «Мир», 1964.
41. К и м К. В., Об использовании специфики условий задачи в методе улучшения плана. Экономика и математические методы, т. I, вып. I, 1965.
42. К о л л а т ц Л., Функциональный анализ и вычислительная математика. Изд-во «Мир», 1969.
43. К о р б у т А. А., Ф и н к е л ь ш т е й н Ю. Ю., Дискретное программирование, Изд-во «Наука», 1969.
44. К о р н а и И., Л и п т а к Т., Планирование на двух уровнях. В сб. «Применение математики в экономических исследованиях», т. 3, Изд-во «Мысль», 1965.
45. К о с с о в В. В., Межотраслевой баланс, Изд-во «Экономика», 1966.
46. К р ы л о в А. П. и др., Проектирование разработки нефтяных месторождений, Гостоптехиздат, 1962.
47. К р ы л о в А. П., Л у т о ш к и н С. С., Изучение гидравлических сопротивлений и удельного веса смеси при работе воздушных подъемников. Труды ВНИИ, вып. 13, Гостоптехиздат, 1958.
48. К у л и б а н о в В. Н., М е е р о в М. В., К проблеме оптимального управления нефтедобывающим предприятием. «Автоматика и телемеханика», № 6, 1970, стр. 124—133.
49. Л и т в а к Б. Л., Статическая оптимизация сложных многосвязных систем. Автореферат диссертации, МИНХ и ГП, 1969.
50. Л и т в а к Б. Л., К вопросу оптимизации одного класса объектов многосвязного регулирования. Труды IV Всесоюзного совещания по автоматическому управлению (технической кибернетике). Сб. [рефератов, Изд-во «Наука», 1968.
51. Л и т в а к Б. Л., О решении задач линейного программирования большой размерности. Технический сборник ВНИИ Нефть «Применение математических методов для решения задач разработки нефтяных месторождений», М., 1968.
52. Л и т в а к Б. Л., О решении одного специального класса задач линейного программирования. Труды МИНХ и ГП, вып. 84, 1969.
53. М а к с и м о в и ч И. Г., Линейные электрические цепи и их преобразования, Госэнергоиздат, 1961.
54. М а н д е л ь А. Б., Анализ решения задач линейного программирования. Экономика и математические методы, т. III, вып. 6, 1967.
55. М е е р о в М. В., Системы многосвязного регулирования. Изв. АН СССР, «Энергетика и автоматика», № 6, 1961.
56. М е е р о в М. В., Системы многосвязного регулирования. Изд-во «Наука», 1965.
57. М е е р о в М. В., Особенности систем многосвязного регулирования и методы их оптимизации. Труды I Международного симпозиума по многосвязному регулированию, Дюссельдорф, 1968.

58. Мееров М. В., Литвак Б. Л., О решении некоторых задач оптимизации многосвязных объектов большой размерности. Изв. АН СССР, «Техническая кибернетика», № 6, 1969.
59. Мееров М. В., Литвак Б. Л., К вопросу оптимизации сложных многосвязных систем большой размерности. ДАН СССР, 1969, т. 187, № 5.
60. Мееров М. В., Литвак Б. Л., Статическая оптимизация многосвязных систем большой размерности I, II, III. «Автоматика и телемеханика», № 3, 4, 5, 1970.
61. Мееров М. В., Салимжанов Э. С., Некоторые вопросы теории управления нефтяными скважинами. Изв. АН СССР, «Техническая кибернетика», № 3, 1963.
62. Мееров М. В., Фридман В. Г., К теории управления нефтяными скважинами. Изв. АН СССР. «Техническая кибернетика», № 3, 1965.
63. Михалевич В. С., Ермольев Ю. М., Шкурба В. В., Шор Н. З., Сложные системы и решение экстремальных задач. «Кибернетика», Киев, № 5, 1967.
64. Мовшович С. М., Питтель Б. Г., Магистральные свойства моделей замкнутой экономики и динамические процессы принятия решений. Экономика и математические методы, т. VI, № 2, 1970.
65. Морозовский В. Т., Многосвязные системы автоматического регулирования, Изд-во «Энергия», 1970.
66. Мухачева Э. А., Транспортная задача на сети с дополнительными ограничениями. Экономика и математические методы, т. I, вып. 4, 1965.
67. Немчинов В. С., Экономико-математические методы и модели, Соцэкгиз, 1962.
68. Первозванская Т. Н., Первозванский А. А., Алгоритм поиска оптимального распределения централизованных ресурсов, Изд-во АН СССР, «Техническая кибернетика», № 3, 1966.
69. Питтель Б. Г., Об асимптотических оценках в динамической задаче принятия решений. Теория вероятностей и ее применение, т. 14, вып. 2, 1969.
70. Прохоров Ю. В., Розанов Ю. А., Теория вероятностей, Изд-во «Наука», 1967.
71. Пшеничный Б. Н., Необходимые условия экстремума, Изд-во «Наука», 1969.
72. Разумихин Б. С., Физические модели и алгоритмы решения некоторых задач математического программирования и математической экономики. V Всесоюзное совещание по проблемам управления. Рефераты докладов, Изд-во «Наука», 1971.
73. Робертс С., Динамическое программирование в процессах химической технологии и методы управления, Изд-во «Мир», 1965.
74. Романовский И. В., Существование оптимального стационарного управления в марковском процессе решения. Теория вероятностей и ее применения, т. 10, № 1, 1965.
75. Романовский И. В., Магистральные теоремы для полумарковских процессов принятия решений. Труды Матем. ин-та АН СССР им. В. А. Стеклова, т. СХI, Ленинград, 1970.

76. Рыбашов М. В., Дубников С. Е., Градиентные методы решения линейных равенств, неравенств и задач линейного программирования на АВМ, Изд-во «Советское радио», 1970.
- 77—78. Сборник «Проблемы повышения эффективности работы транспорта», АН СССР, 1949.
79. Справочник по добыче нефти, под ред. проф. И. М. Муравьева, т. 3, Гостоптехиздат, 1958.
80. Стефанюк В. Л., Цетлин М. Л., О регулировке мощности в коллективе радиостанций. Проблемы передачи информации, т. 3, вып. 4, 1967.
81. Тетельбаум И. М., Электрическое моделирование, Физматгиз, 1959.
82. Фаддеев Д. К., Фаддеева В. Н., Вычислительные методы линейной алгебры, Физматгиз, 1963.
83. Феллер В., Введение в теорию вероятностей и ее приложения, Изд-во «Мир», 1964.
84. Фридман В. Г., Некоторые вопросы теории линейного программирования в пространстве Гильберта и одно его приложение. Труды МИНХ и ГП, вып. 59, Изд-во «Недра», 1965.
85. Хедли Дж., Линейная алгебра, Изд-во «Высшая школа», 1966.
86. Хедли Дж., Нелинейное и динамическое программирование, Изд-во «Мир», 1967.
87. Ховард Р. А., Динамическое программирование и марковские процессы. Изд-во «Советское радио», 1964.
88. Хранович И. Л., Транспортная задача выпуклого программирования и возможности ее решения на АВМ. В сб. «Аналоговая и аналого-цифровая вычислительная техника», вып. 1, Изд-во «Советское радио», 1968, стр. 116—122.
89. Чарный И. А., Подземная гидрогазодинамика, Гостоптехиздат, 1963.
90. Шор Н. З., Применение обобщенного градиентного спуска в блочном программировании. «Кибернетика», № 3, Киев, 1967.
91. Щелкачев В. Н., Разработка нефтеводоносных пластов при упругом режиме, Гостоптехиздат, 1959.
92. Юдин Д. С., Гольштейн Е. Г., Об одном методе количественного анализа упрощенных экономических моделей. Сб. статей под ред. В. С. Немчинова. Применение математики в экономических исследованиях, т. II, Соцэкономиздат, 1962.
93. Юдин Д. С., Гольштейн Е. Г., Задачи и методы линейного программирования, Изд-во «Советское радио», 1964.
94. Dantzig G. B., Wolfe Ph., Decomposition principle for Linear Programms. Oper. Res., v. 8, № 1, 1960, p. 101—112.
95. Denarso Eric V., On linear programming in a Markov decision problem, Manag. Sci., 16, № 5, 1970, 281—288.
96. Derman C., On Sequential Decisions and Markov Chains, Manag. Sci., 9, 1962, 16—24.
97. d'Épenoux F., Sur un Probleme de Production et de Stockage dans L'Aleatoire. Revue Francaise de Recherche Operationnelle (Société Francaise de Recherche Operationnelle), 1960, 3—16.
98. Leontief W. W., The structure of the American Economy, 1919—1920, Oxford, University Press. New York, 1931.

99. M a n n e A. S., Linear Programming and Sequential Decisions. Management Sci., № 3, 259—267, 1960.
100. М е е р о в М. В., Л и т в а к В. Л., Static Optimization of High — Dimensionality Multivariable Systems. IFAC Kyoto Symposium on Systems Engineering Approach to Computer Control, August 1970, Kyoto, Japan.
101. V e r k h o v s k i B. S., Optimal complex Use of Controlled Water Resources of a Basin. Proceedings International Symposium on Mathematical Models in Hydrology, Warsaw, July 1971.

ИСПРАВЛЕНИЯ В РИСУНКАХ

На рис. В.1, В.2, В.3, 1.3, 1.4 компоненты векторов входных и выходных параметров должны быть обозначены строчными (малыми) буквами x_i, y_i .

На рис. П.1, П.2, П.3 всюду вместо Φ должно быть v . На рис. П.2 вместо $a_{i\Phi}^{(p)} < 0$ должно быть $a_{iv}^{(p)} \leq 0$ и вместо λ_1, B должно быть $\Lambda^{(p)}, B^{(p)}$. На рис. П.3 всюду вместо знака $>$ должен быть знак $\geq$.

