

ЭКОНОМИКО-  
МАТЕМАТИЧЕСКАЯ  
БИБЛИОТЕКА

---

В. С. МИХАЛЕВИЧ  
В. А. ТРУБИН  
Н. З. ШОР

ОПТИМИЗАЦИОННЫЕ  
ЗАДАЧИ  
ПРОИЗВОДСТВЕННО-  
ТРАНСПОРТНОГО  
ПЛАНИРОВАНИЯ

В. С. МИХАЛЕВИЧ, В. А. ТРУБИН, Н. З. ШОР

ОПТИМИЗАЦИОННЫЕ ЗАДАЧИ  
ПРОИЗВОДСТВЕННО-  
ТРАНСПОРТНОГО  
ПЛАНИРОВАНИЯ  
МОДЕЛИ, МЕТОДЫ, АЛГОРИТМЫ

5  
11  
11  
16  
23  
26  
29  
31  
31  
33  
46  
61  
69  
90  
108  
108  
109  
114  
117  
130  
37  
40  
46  
46  
52  
57  
63  
69  
69  
78  
3



МОСКВА «НАУКА»  
ГЛАВНАЯ РЕДАКЦИЯ  
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ  
1986

ББК 22.18  
М69  
УДК 519.6

Михалевич В. С., Трубин В. А., Шор Н. З. **Оптимизационные задачи производственно-транспортного планирования: Модели, методы, алгоритмы.**— М.: Наука. Гл. ред. физ.-мат. лит., 1986.— 264 с.

Оптимизационные задачи производственно-транспортного планирования относятся к наиболее распространенному типу задач, возникающих при анализе вопросов долгосрочного и текущего планирования.

Книга посвящена описанию моделей, методов и алгоритмов решения обычных и многоиндексных распределительных задач в линейной, нелинейной и динамической постановках, задач размещения производства и технических средств, синтеза сетей и т. д.

Для специалистов в области прикладной математики и экономики, а также для студентов и аспирантов соответствующих специальностей.

Табл. 10. Ил. 12. Библиогр. 181 назв.

Рецензент

доктор физико-математических наук *Е. Г. Гольштейн*

1702070000—134  
М—————37-86  
053(02)-86

© Издательство «Наука».  
Главная редакция  
физико-математической  
литературы, 1986

Предисловие . . . . .	5
<b>Глава 1. Основные сведения из выпуклого анализа . . . . .</b>	<b>11</b>
§ 1. Выпуклые функции и субградиентные множества . . . . .	11
§ 2. Необходимые условия экстремума и методы негладких штрафных функций . . . . .	16
§ 3. Схемы декомпозиции по переменным . . . . .	23
§ 4. Схемы декомпозиции по ограничениям . . . . .	26
§ 5. Важные классы недифференцируемых функций. Обобщенные градиентные множества . . . . .	29
<b>Глава 2. Алгоритмы недифференцируемой оптимизации . . . . .</b>	<b>31</b>
§ 1. Введение . . . . .	31
§ 2. Обобщенный градиентный спуск . . . . .	33
§ 3. Методы обобщенного градиентного спуска с растяжением пространства . . . . .	46
§ 4. Субградиентные методы, использующие операцию растяжения пространства в направлении разности двух последовательных субградиентов . . . . .	61
§ 5. Методы эллипсоидов . . . . .	69
§ 6. Вычислительная сложность решения задач математического программирования и метод эллипсоидов . . . . .	90
<b>Глава 3. Линейные и выпуклые задачи производственно-транспортного планирования . . . . .</b>	<b>108</b>
§ 1. Введение . . . . .	108
§ 2. Транспортные задачи в матричной и сетевой формах . . . . .	109
§ 3. Многопродуктовые сетевые транспортные задачи . . . . .	114
§ 4. Динамические распределительные задачи . . . . .	117
§ 5. Применение $\epsilon$ -квадратичного сглаживания при решении задачи линейного программирования путем декомпозиции по ограничениям . . . . .	130
§ 6. Задача об оптимальном планировании производства и распределения труб . . . . .	137
§ 7. Стохастические распределительные задачи . . . . .	140
<b>Глава 4. Задачи синтеза сетей и метод решения релаксационной задачи . . . . .</b>	<b>146</b>
§ 1. Задачи синтеза сетей. Модели и свойства . . . . .	146
§ 2. Задачи линейного программирования с переменными верхними границами и блокирующие множества . . . . .	152
§ 3. Динамическая декомпозиция задачи (4.18)—(4.20) и метод ее решения . . . . .	157
§ 4. Динамическая декомпозиция и сетевая транспортная задача . . . . .	163
<b>Глава 5. Задачи размещения производства и синтеза сетей и методы их решения . . . . .</b>	<b>169</b>
§ 1. Простейшая задача размещения производства: свойства и построение оценок . . . . .	169
§ 2. Метод решения простейшей задачи размещения . . . . .	178



§ 3. Некоторые обобщения простейшей задачи размещения . . . . .	189
§ 4. Варианты простейшей задачи размещения . . . . .	198
§ 5. Задача синтеза сетей с одним источником . . . . .	206
§ 6. Задача синтеза надежных сетей . . . . .	216
<b>Глава 6. Полиномиально разрешимые задачи, связанные с размещением производства и синтезом сетей . . . . .</b>	<b>221</b>
§ 1. Градиентные алгоритмы и кратчайшее ветвление в орграфах . . . . .	221
§ 2. Простейшая задача размещения на древовидных сетях . . . . .	227
§ 3. Подкласс задач Штейнера на плоскости с прямоугольной метрикой . . . . .	235
§ 4. Задача Вебера в пространстве с прямоугольной метрикой . . . . .	241
<b>Приложение. Пакет прикладных программ для решения задач произ- водственно-транспортного планирования большой размерности (ПЛАНЕР)</b>	<b>248</b>
<b>Список литературы . . . . .</b>	<b>253</b>

При решении важных задач планирования и управления народным хозяйством, пресектирования сложных объектов и систем на первый план выдвигаются вопросы качества принимаемых решений, в связи с чем возрастает роль методов и алгоритмов решения оптимизационных задач в математическом обеспечении автоматизированных систем различного уровня и назначения. Среди приложений математического программирования большое место занимают проблемы производственно-транспортного типа. Основным источником такого рода задач связан с планированием на отраслевом уровне: распределение заказов между предприятиями с учетом затрат на производство и транспортных издержек, размещение производства, складов и технических средств, решение вопросов развития отрасли (строительство и реконструкция предприятий, техническая модернизация производственных и транспортных отраслей). Близкие по типу задачи возникают при проектировании сложных сетей и объектов: дорог, продуктопроводов, тепловых и электрических сетей, сетей связи и т. д., при решении вопросов рационального распределения грузопотоков между различными видами транспорта.

Вычислительные сложности, возникающие при решении производственно-транспортных задач, зачастую связаны с их большой размерностью, многоэкстремальностью и наличием дискретных переменных. В то же время эти задачи обладают специфической структурой, позволяющей разрабатывать специальные методы их решения, которые обычно более эффективны, чем общие методы линейного, нелинейного и дискретного программирования. Как правило, рассматриваемые модели обладают рядом свойств, близких к транспортной задаче и ее обобщениям, что позволяет использовать эти свойства при построении методов решения. В частности, эти свойства используются при построении схем декомпозиции, которые в сочетании с методами негладкой оптимизации позволяют получить максимальный вычислительный эффект.

С другой стороны, известные алгоритмы, хорошо проявившие себя при решении транспортных и простых распределительных задач (например, метод потенциалов), при переходе на более общие модели теряют свою простоту и становятся менее эффективными. Практическое использование стандартных средств линейного и линейного целочисленного программирования, основанных, как правило, на симплекс-методе и методе ветвей и границ, при решении реальных

задач производственно-транспортного типа большой размерности не приводит к успеху. Нет также данных об успешном применении для реальных задач производственно-транспортного планирования большой размерности известных декомпозиционных алгоритмов типа Данцига — Вулфа, Розена и др.

В Институте кибернетики имени В. М. Глушкова АН УССР начиная с 1960 г. ведутся теоретические исследования моделей, свойств и методов решения различных классов задач производственно-транспортного планирования и накоплен серьезный опыт их практического использования. В настоящей книге в основном изложены результаты, полученные авторами и их учениками, которые принимали непосредственное участие в этой работе. Книгу можно условно разбить на две части. В первых трех главах изложен материал, относящийся к решению одноэкстремальных задач с непрерывными переменными. Здесь речь идет в основном о решении задач линейного и выпуклого программирования большой размерности. Фундамент теоретических рассуждений составляет выпуклый анализ, в основе алгоритмов лежат схемы декомпозиции и методы недифференцируемой оптимизации, приложения связаны с задачами отраслевого планирования, материально-технического снабжения и рационализации перевозок. Во второй части (гл. 4—6) рассматриваются задачи минимизации вогнутой функции на выпуклых многогранных множествах транспортного типа или их обобщениях. Задачи этого вида возникают при проектировании сетей и размещении производства. Они могут быть сформулированы в терминах задач частично целочисленного программирования специального вида и решены соответствующими методами с учетом их специфических свойств и особенностей.

В первой главе приводятся основные факты выпуклого анализа в конечномерном евклидовом пространстве и схемы декомпозиции, приводящие к задачам недифференцируемой оптимизации. Так как основные вопросы выпуклого анализа весьма подробно изложены в ряде известных монографий и учебных пособий [28, 61, 62], то некоторые достаточно элементарные результаты приводятся без доказательств. Большое внимание уделено тем понятиям и вопросам, которые непосредственно используются в последующем изложении: свойствам субградиентных множеств, обобщенной теореме Куна — Таккера, исследованию координирующих задач в схемах декомпозиции по переменным и ограничениям, технике вычисления субградиентов при решении координирующих задач, методу негладких штрафных функций. Известные алгоритмические схемы решения задач квазиблочного линейного программирования, такие как метод Данцига — Вулфа генерации столбцов или метод релаксации Розена, нами не рассматривались, так как интересующийся этими вопросами читатель найдет описание этих методов в монографиях [12, 24]. Отметим, что близкий к нашему подход к реализации схем декомпозиции рассмотрен в работах [10, 143, 162].

В гл. 1 показано, что реализация схем декомпозиции по переменным и по ограничениям приводит к координирующим задачам,

представляющим собой задачи нахождения экстремума негладкой функции при простых ограничениях. Описанию методов решения такого рода задач посвящена гл. 2. Основное содержание этой главы связано с исследованиями субградиентных методов минимизации выпуклых функций, проводившимися в Институте кибернетики АН УССР начиная с 1961 г. Наиболее полно эти исследования были представлены в монографии [97], однако эта книга вышла небольшим тиражом и труднодоступна. Кроме того, в последние годы тематика недифференцируемой оптимизации бурно развивалась как в части теоретического исследования вычислительной сложности задач линейного программирования и информационной сложности выпуклого программирования (без требования гладкости входящих в модель функций), так и в части разработки новых методов негладкой оптимизации, таких как  $\varepsilon$ -субградиентные методы, методы эллипсоидов и их обобщения. Поэтому во второй главе обзор современного состояния методов субградиентного типа дан достаточно подробно. Изложена теория методов обобщенного градиентного спуска (ОГС), субградиентных методов с растяжением пространства в направлении градиента (ОГСРП). Детально описаны методы субградиентного типа с растяжением пространства в направлении разности двух последовательных градиентов, так называемые  $r$ -алгоритмы, предложенные в [89]. Эти методы являются в настоящее время наиболее практичными эффективными средствами недифференцируемой выпуклой минимизации, причем их эффективность особенно проявляется при решении задач производственно-транспортного планирования большой размерности. Многочисленные экспериментальные и практические расчеты с помощью некоторых вариантов  $r$ -алгоритмов показывают устойчивую сходимость последовательности «рекордов» по функционалу со скоростью геометрической прогрессии — за  $n$  итераций ( $n$  — размерность пространства аргументов минимизируемой функции) относительная точность улучшается в 3—10 раз. В то же время теоретические оценки для скорости сходимости  $r$ -алгоритмов не получены. С другой стороны, начиная с 1976 г. [95] пристальное внимание многих исследователей привлечено к методу эллипсоидов, который получается из метода ОГСРП при определенных значениях его параметров [96]. Хотя этот метод практически уступает  $r$ -алгоритму по скорости сходимости примерно в  $n$  раз, теоретически он хорошо обоснован и гарантирует сходимость рекордов по функционалу со скоростью геометрической прогрессии со знаменателем  $q \approx 1 - 1/(2n^2)$ . Метод эллипсоидов оказался весьма удобным средством анализа сложности различных задач математического программирования. Близость  $q$  к единице стимулировала поиск различных модификаций метода эллипсоидов с целью ускорения его сходимости и приближения его эффективности к эффективности  $r$ -алгоритмов. О некоторых результатах в этом направлении, полученных в ИК АН УССР [20, 21, 98], сообщается в § 5 гл. 2. И хотя окончательный успех в этом направлении не достигнут, все же определенные модификации метода эллипсоидов имеют и практическое значение. В частности, они могут использоваться в сочетании с  $r$ -алгоритмом на последней

стадии процесса минимизации, когда требуется получить гарантированную точность по функционалу, а путем анализа полученного решения точность оценить затруднительно.

Последний параграф гл. 2 как бы перекидывает мостик от проблем выпуклой минимизации к дискретным экстремальным задачам. Здесь речь идет об анализе сложности различных экстремальных задач с использованием метода эллипсоидов, в частности приводятся интересные результаты Л. Г. Хачияна, впервые обосновавшего полиномиальные алгоритмы линейного и квадратичного программирования [77, 78], а также результаты известной работы [141], в которой аппарат метода эллипсоидов используется для анализа сложности дискретных задач и показывается, что для многих дискретных задач, для которых получены алгоритмы полиномиальной сложности, легко строится полиномиальный алгоритм, основанный на методе эллипсоидов. На практике метод эллипсоидов из-за своей медленной сходимости (требуется порядка  $4,6n^2$  итераций, чтобы гарантировать улучшение точности на один десятичный знак;  $n$  — размерность пространства) оказался малоприменимым.

Кроме метода эллипсоидов в линейном программировании известен также полиномиальный алгоритм Булатова — Ямницкого — Левина [2, 102], в котором роль тел, локализирующих оптимум, выполняют  $n$ -мерные симплексы, а также разработанный Кармаркаром [144] оригинальный алгоритм, основанный на проективных преобразованиях пространства. Практическая эффективность последнего алгоритма пока мало исследована.

В гл. 3 приводятся математические модели наиболее часто встречающихся на практике задач производственно-транспортного планирования с непрерывными переменными. Показывается, что при большой размерности задач такого типа наиболее эффективными методами их решения являются схемы декомпозиции в сочетании с алгоритмом субградиентного типа с растяжением пространства в направлении разности последовательных субградиентов ( $r$ -алгоритм). Большинство рассмотренных в гл. 3 моделей встечались в практике работы отдела экономической кибернетики ИК АН УССР, по ним были проведены многочисленные реальные расчеты, показавшие надежность и эффективность соответствующего математического и программного обеспечения. Часть функциональных модулей, предназначенных для решения конкретных задач, вошла в пакет прикладных программ ПЛАНЕР, краткое описание которого содержится в Приложении.

При разработке планов развития большинства отраслей народного хозяйства страны возникают задачи оптимального размещения вновь строящихся и реконструкции существующих предприятий, а также задачи оптимального проектирования коммуникационных сетей различного назначения: дорожных, связи, энергетических и др.

Актуальность исследования свойств и методов решения таких задач связана с их важностью и сложностью решения. Они относятся к задачам минимизации вогнутой функции на транспортных многогранниках и их обобщениях. Как правило, большинство таких задач являются универсальными ( $NP$ -трудными) задачами. Поэтому мало-



вероятно, что для них могут быть созданы методы столь же эффективные, как методы линейной и выпуклой оптимизации. В связи с этим при исследовании таких задач необходимо ориентироваться на разработку методов, которые позволяют за приемлемое время построить приближенное решение с заданной относительной погрешностью. Необходимым элементом исследования таких классов задач и методов их решения является в связи со сказанным их экспериментальное исследование с целью выявления зависимостей времени решения от задаваемой погрешности.

Наиболее распространенным способом решения  $NP$ -трудных задач являются методы типа ветвей и границ, в которых нижние оценки функционала (для задачи минимизации) получаются релаксацией исходной задачи путем линеаризации ее целевой функции либо сведением исходной задачи к задаче целочисленного программирования с последующей ее заменой на задачу линейного программирования. В последнем случае соответствующая оценочная (релаксированная) задача может обладать специфическими особенностями, которые целесообразно учитывать при разработке методов ее решения.

В гл. 4 рассмотрена общая постановка задач синтеза сетей и размещения производства, приведена классификация этих задач и описан ряд свойств таких задач и способ их сведения к задачам целочисленного линейного программирования специального вида. Основная часть этой главы посвящена исследованию метода динамической декомпозиции для задач линейного программирования следующего вида:

$$\min \{cX + fY : AX = b, X \leq DY, X \geq 0, Y \geq 0\}, \quad (*)$$

где  $c, f, b$  — заданные векторы,  $A$  — произвольная матрица,  $D$  — матрица с неотрицательными коэффициентами. Частными случаями данной задачи являются релаксированные задачи размещения производства и синтеза сетей. Предложенный метод декомпозиции сводит решение этой задачи к решению последовательности задач меньшей размерности вида

$$\max \{\varphi : AX = b\varphi, X \leq D\bar{Y}, X \geq 0\}, \quad (**)$$

где  $\varphi$  — скалярная переменная,  $\bar{Y}$  — фиксированное значение вектора  $Y$ . Задача (\*\*) является обобщением для произвольных  $A, b$  задачи о максимальном потоке в сети. Она превращается в последнюю, если  $A$  — матрица инцидентий орграфа, а вектор  $b$  имеет ровно две ненулевые компоненты:  $b_s = -1, b_t = +1$  для заданных индексов  $s, t$ . В этом случае задача (\*\*) может быть эффективно решена методами с полиномиальной трудоемкостью [3]. В случае произвольного вектора  $b$  решение задачи (\*\*) сводится к решению последовательности задач о максимальном потоке в графе. Общая схема решения декомпозиционной задачи является схемой симплексного типа. В ней ищется решение двойственной задачи с матрицей, строки которой соответствуют переменным  $Y$ . Столбец, вводимый в базис, определяется путем решения задачи (\*\*).

Реальные задачи синтеза сетей на графах со средним числом вершин  $n$  порождают задачи целочисленного программирования с большим числом переменных, порядка  $n^2 - n^4$  (в зависимости от типа задачи). Поэтому трудоемкость точного решения таких задач лучшим из известных алгоритмов оценивается громадной величиной:  $O(n^7) - O(n^{14})$ . В связи с этим точное их решение становится бесперспективным. Естественной альтернативой становится приближенное решение оценочных задач с трудоемкостью, существенно меньшей, чем указанная. В гл. 5 приводятся алгоритмы приближенного решения задачи, получаемой из (\*) методом динамической декомпозиции, для ряда классов задач размещения производства и синтеза сетей. Здесь же приведены схемы метода ветвей и границ, реализованные для решения этих задач и использующие их специфические особенности. Приведены результаты численного решения ряда практических и тестовых задач рассматриваемых классов, позволяющие составить представление об эффективности разработанного программного обеспечения.

Гл. 6 посвящена исследованию полиномиально разрешимых подклассов задач синтеза сетей. Такими задачами являются: задача о кратчайшем ветвлении в орграфах, простейшая задача на сети в форме дерева, задача Штейнера на плоскости с прямоугольной метрикой с дополнительными ограничениями на структуру дерева Штейнера и задача Вебера в пространстве с прямоугольной метрикой. Исследование задач с полиномиальной оценкой трудоемкости позволяет выделить те специфические особенности задач, наличие или отсутствие которых определяет грань между «хорошими» и «плохими» задачами. Кроме того, знание «хороших» алгоритмов решения задач из некоторых подклассов позволяет конструировать приближенные и эвристические алгоритмы решения задач из более широких классов «плохих» задач.

При написании книги авторы, считая, что к настоящему времени имеется достаточно обширная литература по теории графов, традиционным методам целочисленного и линейного программирования, в том числе по методам решения транспортных задач, сочли целесообразным включение этих материалов в книгу. Для подробного ознакомления с ними мы отсылаем читателя к монографиям [3, 25].

## ОСНОВНЫЕ СВЕДЕНИЯ ИЗ ВЫПУКЛОГО АНАЛИЗА

## § 1. Выпуклые функции и субградиентные множества

Мы ограничимся в основном рассмотрением функций, определенных на подмножествах  $n$ -мерного евклидова пространства  $E^n$ . Очень часто эти функции будут выпуклыми или вогнутыми. Напомним читателю основные определения и свойства таких функций [28, 62].

Пусть задана действительная функция  $f(x)$ , определенная на множестве  $X \subset E^n$ . Графиком функции  $f(x)$  называется подмножество  $\Gamma[X, f]$   $(n+1)$ -мерного пространства  $E^{n+1}$ , состоящее из точек вида  $\{x, f(x)\}$  ( $x \in X$ ). Надграфиком функции назовем подмножество  $\Gamma^+[X, f] \subset E^{n+1}$  вида  $\{x, u\}$ , где  $u \geq f(x)$  ( $x \in X$ ). Функция  $f(x)$  называется выпуклой, если ее надграфик является выпуклым множеством. Из этого определения сразу вытекает следующая теорема.

**Теорема 1.1.** Для того чтобы функция  $f(x)$ , определенная на множестве  $X \subset E^n$ , была выпуклой, необходимо и достаточно, чтобы:

- а) множество  $X$  было выпуклым;
- б) для любых точек  $x_1, x_2 \in X$  и числа  $\alpha$  ( $0 \leq \alpha \leq 1$ ) выполнялось соотношение

$$\alpha f(x_1) + (1 - \alpha) f(x_2) \geq f(\alpha x_1 + (1 - \alpha) x_2). \quad (1.1)$$

Свойство (1.1) обобщается на произвольное конечное семейство точек  $x_1, x_2, \dots, x_k \in X$  и набор чисел  $\alpha_1, \alpha_2, \dots, \alpha_k \geq 0$  таких, что

$$\sum_{i=1}^k \alpha_i = 1:$$

$$\sum_{i=1}^k \alpha_i f(x_i) \geq f\left(\sum_{i=1}^k \alpha_i x_i\right). \quad (1.2)$$

Выпуклые функции  $f(x)$  обладают также следующими свойствами.

I. Во всех внутренних точках области определения  $X$   $f(x)$  непрерывна [62]. Это значит, что если  $f(x)$  определена на всем пространстве  $E^n$ , то она непрерывна.

II.  $f(x)$  почти всюду (за исключением точек меры нуль) дифференцируема, причем если в некоторой точке  $f(x)$  дифференцируема, то она непрерывно дифференцируема в этой точке.

III. Через любую точку  $\{y, f(y)\} \in \Gamma[X, f]$  можно провести опорную гиперплоскость к надграфу  $\Gamma^+[X, f]$ . Если точка  $y$  является внутренней в  $X$ , то уравнение этой опорной гиперплоскости

имеет вид

$$(p, x - y) = u - f(y);$$

при этом для всех точек надграфика  $\{x, z\}$  должно выполняться соотношение

$$z - f(y) \geq (p, x - y).$$

Если  $z = f(x)$ , то для всех  $x \in X$

$$f(x) - f(y) \geq (p, x - y). \quad (1.3)$$

Произвольный вектор  $p$ , удовлетворяющий неравенству (1.3) для всех  $x \in X$ , называется *субградиентом* (обобщенным градиентом) функции  $f(x)$  в точке  $y$ .

Рассмотрим *лебегово множество*  $L(y) = \{x \in X: f(x) \leq f(y)\}$ . Это множество является выпуклым, и в силу неравенства (1.3) справедливо неравенство  $(p, y - x) \geq 0$  для всех  $x \in L(y)$ . Если вектор  $p$  ненулевой, то его можно интерпретировать как вектор внешней нормали к опорной гиперплоскости к множеству  $L(y)$ , ограниченному гиперповерхностью  $f(x) = f(y)$ .

Назовем *субградиентным множеством* (множеством обобщенных градиентов) выпуклой функции  $f(x)$  в точке  $y$  множество векторов  $p$ , удовлетворяющих соотношению (1.3), и будем обозначать его  $G_f(y)$ , а его представителей —  $g_f(y)$ .

**Теорема 1.2.** В любой внутренней точке  $y$  области определения  $X$  выпуклой функции  $f(x)$  существует непустое выпуклое замкнутое и ограниченное множество субградиентов. Если  $G_f(y)$  состоит из единственной точки  $g_f(y)$ , то  $g_f(y)$  является градиентом функции  $f(x)$  в точке  $y$ ; в противном случае в точке  $y$   $f(x)$  недифференцируема.

IV. Из (1.3) сразу следует, что если внутренняя точка  $y$  является точкой минимума  $f(x)$  на  $X$ , то  $0 \in G_f(y)$ . Справедливо и обратное: если  $0 \in G_f(y)$ , то  $y$  является точкой минимума. Далее, любой локальный минимум выпуклой функции является глобальным.

V. Пусть для некоторых  $y \in X$ ,  $t > 0$  и  $e \in E^n$  ( $e \neq 0$ ) выполняется соотношение  $y + te \in X$ . Рассмотрим

$$\lim_{t \rightarrow +0} \frac{f(y + te) - f(y)}{t} = f'_e(y). \quad (1.4)$$

Если предел (1.4) существует, то  $f'_e(y)$  называется *производной по направлению  $e$*  функции  $f(x)$  в точке  $y$ .

**Теорема 1.3.** В любой внутренней точке  $y$  области определения  $X$  выпуклой функции  $f(x)$  существует производная  $f'_e(y)$  по любому направлению  $e \in E^n$ ; при этом  $f'_e(y)$  связана с субградиентным множеством  $G_f(y)$  соотношением

$$f'_e(y) = \max_{g \in G_f(y)} (g, e).$$

Рассмотрим параметрическое семейство точек (луч)  $W_e(t) = y + te$ ,  $t \geq 0$ ,  $y + te \in X$ . Тогда производная по направлению  $f'_e(W_e(t))$  является монотонно неубывающей функцией от  $t$ .

VI. Назовем направлением наискорейшего спуска в точке  $y \in X$  направление  $\eta^*$ , на котором достигается минимум  $f_{\eta^*}(y)$  при  $\|\eta\| = 1$ .

Теорема 1.4. Если точка  $y$  является внутренней и  $G_f(y)$  не содержит 0 (т. е. точка  $y$  не является точкой минимума), то вектор  $\eta^* = -g^*(y)/\|g^*(y)\|$ , где  $g^*(y) \in G_f(y)$  — ближайшая к 0 точка из субградиентного множества, является единственным направлением наискорейшего спуска.

Если в точке  $y$  функция дифференцируема, то направление наискорейшего спуска, естественно, совпадает с нормированным антиградиентом.

VII. Пусть задано конечное семейство выпуклых функций  $f_1(x)$ ,  $f_2(x)$ , ...,  $f_k(x)$ , определенных на выпуклом множестве  $X$ . С помощью операций взятия линейных комбинаций с неотрицательными коэффициентами и взятия максимума (поточечного) мы можем получать новые выпуклые функции.

Теорема 1.5. Если  $\{f_i(x)\}_{i=1}^k$  — семейство выпуклых функций, определенных на  $X$ ,  $c_1, \dots, c_k$  — отрицательные числа, то функция

$$f(x) = \sum_{i=1}^k c_i f_i(x)$$

также является выпуклой функцией с областью определения  $X$ .

Субградиентное множество  $G_f(y)$  функции  $f(x)$  во внутренней точке  $y \in X$  строится следующим образом:

$$G_f(y) = \sum_{i=1}^k c_i G_{f_i}(y), \quad (1.5)$$

т. е. любой вектор  $g_f(y) \in G_f(y)$  может быть представлен в виде

$$g_f(y) = \sum_{i=1}^k c_i g_{f_i}(y), \quad g_{f_i}(y) \in G_{f_i}(y), \quad i = 1, \dots, k, \quad (1.6)$$

и обратно, произвольный вектор вида (1.6) является субградиентом  $f(x)$  в точке  $y$ .

Теорема 1.6. Функция  $\varphi(x) = \max_{1 \leq i \leq k} f_i(x)$  является выпуклой функцией, определенной на  $X$ ; при этом субградиентное множество  $G_\varphi(y)$  ( $y$  — внутренняя точка множества  $X$ ) является выпуклой оболочкой объединения субградиентных множеств  $G_{f_i}(y)$  ( $i \in I(y)$ ), где  $I(y)$  — подмножество индексов функций  $f_i$ , на которых в точке  $y$  достигается максимум. Это означает, что любой вектор  $g_\varphi(y) \in G_\varphi(y)$  представим в виде

$$g_\varphi(y) = \sum_{i \in I(y)} \alpha_i g_{f_i}(y), \quad \alpha_i \geq 0, \quad \sum_{i \in I(y)} \alpha_i = 1, \quad (1.7)$$

и наоборот, любой вектор, представимый в виде (1.7), является субградиентом функции  $\varphi$  в точке  $y$ .



VIII. Функция  $f_1(x)$ , для которой  $-f_1(x)$  выпукла, называется *вогнутой*. Для вогнутых функций «с точностью до перемены знаков» сохраняются свойства выпуклых функций. В частности, любой локальный максимум вогнутой функции является глобальным. Аналогом субградиента в точке  $y$  для вогнутых функций является *суперградиент*  $g_{f_1}(y) = -g_f(y)$ , где  $f(x) = -f_1(x)$ . Суперградиент (который мы также будем называть *обобщенным градиентом*) удовлетворяет неравенству

$$f(x) - f(y) \leq (g_{f_1}(y), x - y) \quad \forall x \in X.$$

*Направление наискорейшего подъема* для вогнутой функции  $f_1(x)$  совпадает с направлением наискорейшего спуска для  $f(x) = -f_1(x)$ . Операция взятия (поточечного) минимума на семействе вогнутых функций дает в результате вогнутую функцию и т. д.

IX. При использовании функций Лагранжа и их модификаций мы будем иметь дело с выпукло-вогнутыми функциями. Пусть  $F(z) = F(x, y)$  — функция двух векторных переменных,  $x \in X \subset E^n$ ,  $y \in Y \subset E^m$ ,  $z = \{x, y\}$  принадлежит  $X \times Y$  — области определения  $F$ . Функция  $F(x, y)$  называется *выпукло-вогнутой*, если при любом фиксированном  $\bar{y} \in Y$   $F(x, \bar{y})$  выпукла на  $X$  и при любом фиксированном  $\bar{x} \in X$   $F(\bar{x}, y)$  вогнута на  $Y$ . Заметим, что отсюда следует, что  $X$  и  $Y$  — выпуклые множества в соответствующих пространствах. Для выпукло-вогнутых функций характерным является существование седловой точки  $\{x^*, y^*\}$ , которая определяется следующим соотношением:

$$F(x^*, y) \leq F(x^*, y^*) \leq F(x, y^*) \quad \forall y \in Y, x \in X.$$

В теории игр [11] были доказаны весьма общие теоремы о существовании седловых точек.

В частности, если  $X$  и  $Y$  — компактные множества,  $F(x, y)$  — непрерывная на  $X \times Y$  выпукло-вогнутая функция, то

$$\max_{y \in Y} \min_{x \in X} F(x, y) = \min_{x \in X} \max_{y \in Y} F(x, y).$$

Обозначим:

$$\varphi(y) = \min_{x \in X} F(x, y), \quad \psi(x) = \max_{y \in Y} F(x, y),$$

$$\alpha = \max_{y \in Y} \varphi(y) = \min_{x \in X} \psi(x),$$

$$Y(x) = \{y \in Y : F(x, y) = \psi(x)\}, \quad Y^* = \{y \in Y : \varphi(y) = \alpha\},$$

$$X(y) = \{x \in X : F(x, y) = \varphi(y)\}, \quad X^* = \{x \in X : \psi(x) = \alpha\}.$$

Покажем, что если  $x^* \in X^*$ ,  $y^* \in Y^*$ , то  $\{x^*, y^*\}$  — седловая точка. Если  $y^* \notin Y(x^*)$ , то  $F(x^*, y^*) > \varphi(y^*) = \alpha$ . С другой стороны,  $F(x^*, y^*) \leq \varphi(y^*) = \alpha$ . Полученное противоречие показывает, что  $y^* \in Y(x^*)$ . Аналогично показывается, что  $x^* \in X(y^*)$ . Таким образом, точка  $\{x^*, y^*\}$  является седловой, так как в ней при фиксированном  $y = y^*$  достигается минимум по  $x$  и при фиксированном  $x = x^*$  достигается максимум по  $y$ .

При разработке методов недифференцируемой минимизации выпуклых функций используется понятие  $\varepsilon$ -субградиента. Пусть  $f(x)$  — определенная на  $E^n$  выпуклая функция,  $y \in E^n$ ,  $\varepsilon \geq 0$ ;  $\varepsilon$ -субградиентом  $g_f^\varepsilon(y)$  функции  $f(x)$  в точке  $y$  называется произвольный вектор  $g$ , удовлетворяющий при любом  $x$  неравенству

$$f(x) - f(y) \geq (g, x - y) - \varepsilon. \quad (1.8)$$

Совокупность  $\varepsilon$ -субградиентов функции  $f$  в точке  $y$  образует  $\varepsilon$ -субградиентное множество  $G_f^\varepsilon(y)$ . При  $\varepsilon = 0$   $\varepsilon$ -субградиентное множество совпадает с субградиентным. При  $\varepsilon_1 > \varepsilon_2$  справедливо включение  $G_f^{\varepsilon_1}(y) \supset G_f^{\varepsilon_2}(y)$ .

Теорема 1.7. Для любых  $y \in E^n$  и  $\varepsilon \geq 0$  множество  $G_f^\varepsilon(y)$  непусто, замкнуто, выпукло и ограничено.

Назовем  $\varepsilon$ -производной по направлению  $\eta$  в точке  $y$  величину

$$\frac{\partial_\eta f(y)}{\partial \eta} = \max_{v \in G_f^\varepsilon(y)} (v, \eta).$$

Справедливо соотношение

$$\frac{\partial_\eta f(y)}{\partial \eta} = \inf_{\alpha > 0} \frac{f(y + \alpha \eta) - f(y) + \varepsilon}{\alpha}.$$

Пусть  $\varepsilon \geq 0$ . Точку  $\bar{x} \in E^n$  будем называть  $\varepsilon$ -стационарной, если  $0 \in G_f^\varepsilon(\bar{x})$ . Из (1.8) вытекает, что  $f(\bar{x}) - f^* \leq \varepsilon$ , где  $f^* = \inf_{x \in E^n} f(x)$ . Если

точка  $y$  не является  $\varepsilon$ -стационарной, то можно определить направление  $\varepsilon$ -наискорейшего спуска  $\eta^*$  как направление, дающее минимальное значение  $\varepsilon$ -производной среди направлений  $\eta$ ,  $\|\eta\| = 1$ . Это направление может быть получено тем же путем, что и направление наискорейшего спуска, только роль  $G_f(y)$  играет  $G_f^\varepsilon(y)$ .

Напомним, что расстоянием между двумя множествами  $A$  и  $B$ ,  $A \subset E^n$ ,  $B \subset E^n$ , в метрике Хаусдорфа называется число

$$\rho(A, B) = \sup_{a \in A} \inf_{b \in B} \|a - b\| + \sup_{b \in B} \inf_{a \in A} \|a - b\|.$$

Точечно-множественное отображение  $x \mapsto C(x)$  называется непрерывным по Хаусдорфу в точке  $y$ , если

$$\lim_{x \rightarrow y} \rho(C(x), C(y)) = 0.$$

Если  $f(x)$  — выпуклая функция, определенная на  $E^n$ , то отображение  $x \mapsto G_f^\varepsilon(x)$ ,  $\varepsilon > 0$ , непрерывно по Хаусдорфу [28, 54]. Свойство непрерывности  $\varepsilon$ -субградиентного отображения играет основную роль при обосновании сходимости к  $\varepsilon$ -стационарной точке монотонных процессов  $\varepsilon$ -субградиентного типа [28, 153 — 155, 177].

## § 2. Необходимые условия экстремума и методы негладких штрафных функций

Среди теорем о седловых точках важное значение имеют теоремы о седловых точках функции Лагранжа.

Рассмотрим задачу математического программирования следующего вида: определить

$$\inf f_0(x) \quad (1.9)$$

при ограничениях

$$x \in X \subset E^n; \quad f_i(x) \leq 0, \quad i = 1, \dots, m. \quad (1.10)$$

Возьмем  $m$ -мерный вектор  $u = \{u_1, \dots, u_m\}$  и составим функцию Лагранжа:

$$\mathcal{L}(x, u) = f_0(x) + \sum_{i=1}^m u_i f_i(x).$$

Пусть оптимальное значение целевой функции равно  $f_0^*$  (если система ограничений несовместна, будем полагать  $f_0^* = +\infty$ ). При  $u \geq 0$  для любого допустимого вектора  $x$   $\mathcal{L}(x, u) \leq f_0(x)$ . Отсюда

$$\psi(u) = \inf_{x \in X} \mathcal{L}(x, u) \leq f_0^*, \quad u \geq 0,$$

т. е.  $\psi(u)$  дает оценку снизу для  $f_0^*$  при всех  $u \geq 0$ .

Для того чтобы получить более точную оценку, нужно решать задачу отыскания

$$\psi^* = \sup_{u \geq 0} \psi(u).$$

Теоремы о седловой точке функции Лагранжа дают условия, при которых  $\psi^* = f_0^*$ . Для того чтобы получить такие теоремы, необходимо конкретизировать задачу (1.9), (1.10).

В общем случае указанная выше схема лежит в основе реализации процессов, использующих декомпозицию по ограничениям. При довольно общих предположениях  $\psi(u)$  является вогнутой функцией, обобщенный градиент которой вычисляется сравнительно просто.

**Л е м м а 1.1.** Если функции  $f_0(x)$ ,  $f_i(x)$  ( $i = 1, \dots, m$ ) непрерывны на компактном множестве  $X$ , то  $\psi(u)$  определена при всех  $u \geq 0$  и вогнута.

Пусть  $\bar{x}(\bar{u})$  — точка минимума функции  $\mathcal{L}(x, \bar{u})$  по  $x \in X$ . Тогда вектор  $\{f_i(\bar{x}(\bar{u}))\}_{i=1}^m$  является суперградиентом функции  $\psi(u)$  в точке  $\bar{u}$ .

**Д о к а з а т е л ь с т в о.** При указанных выше предположениях функция  $\mathcal{L}(x, u)$  является непрерывной на  $X$  при любом  $u$ , поэтому существует точка  $x(u) \in X$  (не обязательно единственная), на которой достигается

$$\min_{x \in X} f(x, u) = f(x(u), u) = \psi(u).$$

Пусть  $u_1, u_2 \in E^m$ ,  $\alpha_1, \alpha_2 \geq 0$ ,  $\alpha_1 + \alpha_2 = 1$ . Положим  $u_3 = \alpha_1 u_1 + \alpha_2 u_2$ ;

тогда

$$\begin{aligned}\psi(u_3) &= \psi(\alpha_1 u_1 + \alpha_2 u_2) = \alpha_1 \mathcal{L}(x(u_3), u_1) + \alpha_2 \mathcal{L}(x(u_3), u_2) \geq \\ &\geq \alpha_1 \mathcal{L}(x(u_1), u_1) + \alpha_2 \mathcal{L}(x(u_2), u_2) = \alpha_1 \psi(u_1) + \alpha_2 \psi(u_2).\end{aligned}$$

Отсюда вытекает вогнутость функции  $\psi(u)$ . Далее,

$$\begin{aligned}\psi(u) - \psi(\bar{u}) &= \mathcal{L}(x(u), u) - \mathcal{L}(x(\bar{u}), \bar{u}) \leq \mathcal{L}(x(\bar{u}), u) - \mathcal{L}(x(\bar{u}), \bar{u}) = \\ &= \sum_{i=1}^m (u_i - \bar{u}_i) f_i(x(\bar{u})) = (u - \bar{u}, f(x(\bar{u}))), \quad f(x(\bar{u})) = \{f_i(x(\bar{u}))\}_{i=1}^m.\end{aligned}$$

Отсюда видно, что вектор невязок в ограничениях  $f(x(u))$  является суперградиентом функции  $\psi(u)$  в точке  $\bar{u}$ . Доказательство леммы закончено.

Подведем некоторые итоги. Использование функции Лагранжа позволяет на время исключить ограничения  $f_i(x) \leq 0$  ( $i = 1, \dots, m$ ). Если задача минимизации  $\mathcal{L}(x, u)$  по  $x \in X$  решается эффективно, то мы получим  $\psi(u)$  и значение суперградиента в этой точке. Эта информация является достаточной для применения обобщенных градиентных методов негладкой оптимизации (о которых речь будет идти ниже) с целью приближения к  $\psi^*$ . Заметим, что функция  $\psi(u)$  непрерывно дифференцируема в точке  $\bar{u}$ , если  $x(\bar{u})$  определяется однозначно. Если же при некоторых  $u$   $x(u)$  определяется неоднозначно, то функция  $\psi(u)$  является, вообще говоря, недифференцируемой.

Используя обобщенные градиентные процессы, мы можем приблизиться к  $\psi^*$  и получить оценку снизу для  $f_0^*$ , близкую к наилучшей в данном классе оценок. Эти оценки можно применять в сочетании с другими методами, например со схемами ветвления или отсечения, для получения точного или приближенного решения первоначальной задачи; при этом для приближенного допустимого решения мы получаем оценку отклонения от оптимального значения функционала. Если даже априори известно, что  $\psi(u^*) = \max_{u \geq 0} \psi(u) = f_0^*$ , то получение  $u^*$  и соответствующего  $x(u^*)$  не всегда дает решение первоначальной задачи, так как  $x(u^*)$  может оказаться недопустимым. Если же пара  $(x(u^*), u^*)$  является седловой точкой функции Лагранжа, то отсюда следует, что  $x(u^*)$  — оптимальное решение задачи (1.9), (1.10).

В самом деле, пусть  $\{x(u^*), u^*\}$  удовлетворяет неравенству

$$\mathcal{L}(x(u^*), u^*) \geq \mathcal{L}(x(u^*), u) \quad \forall u \geq 0.$$

Тогда, если  $u_i^* > 0$ , то  $f_i(x(u^*)) \leq 0$ , а если  $f_i(x(u^*)) < 0$ , то  $u_i^* = 0$ , так как в противном случае за счет вариации  $u_i$  мы можем увеличить  $\mathcal{L}(x(u^*), u^*)$ . Таким образом, в этом случае  $x(u^*)$  будет допустимым решением и оптимальным, так как

$$f_0(x(u^*)) = \mathcal{L}(x(u^*), u^*) = \psi(u^*) = f_0^*.$$

Существование седловой точки функции Лагранжа доказано для задач выпуклого программирования в форме теоремы Куна—Таккера. Учитывая важность этой теоремы для дальнейшего изложения, приведем ее доказательство.

**Теорема 1.8 (Куна—Таккера).** Пусть имеется задача выпуклого программирования: найти

$$\min f_0(x) \quad (1.11)$$

при ограничениях

$$x \in X \subset E^n; \quad f_i(x) \leq 0, \quad i = 1, \dots, m, \quad (1.12)$$

где  $X$  — замкнутое выпуклое множество,  $f_\nu(x)$  ( $\nu = 0, 1, \dots, m$ ) — выпуклые функции. Пусть  $u = \{u_1, \dots, u_m\}$ ,

$$\mathcal{L}(x, u) = f_0(x) + \sum_{i=1}^m u_i f_i(x)$$

— функция Лагранжа и пусть выполняется условие Слейтера: существует такая точка  $\bar{x} \in X$ , что  $f_i(\bar{x}) < 0$  для  $i = 1, \dots, m$ .

Тогда для того, чтобы  $x^*$  было оптимальным решением задачи (1.11), (1.12), необходимо и достаточно существование такого  $u^*$ , что  $\{x^*, u^*\}$  — седловая точка функции Лагранжа на множестве  $X \times \{u : u \geq 0\}$ ; при этом выполняется условие  $u_i^* f_i(x^*) = 0$  ( $i = 1, \dots, m$ ).

**Доказательство.** Достаточность доказывается просто — при помощи рассуждений, проведенных выше для общего случая. Докажем необходимость, следуя [38]. Рассмотрим в  $E^{m+1}$  подмножество  $A$  векторов  $a = \{a_0, a_1, \dots, a_m\}$  таких, что система неравенств

$$f_0(x) - f_0(x^*) \leq a_0, \quad f_i(x) \leq a_i, \quad i = 1, \dots, m, \quad (1.13)$$

выполняется при некотором  $x \in X$ . Множество  $A$  выпукло. В самом деле, выберем  $a^1, a^2 \in A$ ,  $\alpha_1, \alpha_2 \geq 0$  и  $\alpha_1 + \alpha_2 = 1$ . Пусть  $a^3 = \alpha_1 a^1 + \alpha_2 a^2$ . Покажем, что  $a^3 \in A$ .

Так как  $a^1, a^2 \in A$ , то системы неравенств  $f_0(x) - f_0(x^*) \leq a_j^1$ ,  $f_i(x) \leq a_i^1$  выполняются при некоторых  $x^1 \in X$  ( $j = 1, 2$ ). Рассмотрим  $x^3 = \alpha_1 x^1 + \alpha_2 x^2$ . В силу выпуклости  $X$   $x^3 \in X$ . Далее, так как  $f_\nu$  ( $\nu = 0, 1, \dots, m$ ) выпуклы, имеем

$$f_0(x^3) - f_0(x^*) \leq \alpha_1 f_0(x^1) + \alpha_2 f_0(x^2) \leq \alpha_1 a_0^1 + \alpha_2 a_0^2 = a_0^3.$$

Аналогично  $f_i(x^3) \leq a_i^3$  ( $i = 1, \dots, m$ ), откуда следует, что  $a^3 \in A$ , т. е. множество  $A$  выпукло. Если  $a_\nu \geq 0$  ( $\nu = 0, 1, \dots, m$ ), то при  $x = x^*$  неравенства (1.13) выполняются, т. е. множество точек с положительными координатами содержится в  $A$ . С другой стороны, в силу определения  $x^*$  точка 0 лежит на границе  $A$ , и благодаря условию Слейтера среди точек множества  $A$  имеются такие, у которых  $a_i < 0$  ( $i = 1, \dots, m$ ).



Так как точка 0 лежит на границе множества  $A$ , то в соответствии с теоремой об опорной гиперплоскости [28] через нее можно провести опорную гиперплоскость к  $A$ , т. е. найдется такой вектор  $v = \{v_0, v_1, \dots, v_m\} \neq 0$ , что

$$v_0 a_0 + \sum_{i=1}^m v_i a_i \geq 0 \quad \forall a \in A. \quad (1.14)$$

Если выбрать точку  $a \in A$  так, чтобы  $a_i < 0$  ( $i = 1, \dots, m$ ), то из (1.14) будет вытекать, что  $v_0 > 0$ . Далее, если среди  $v_i$  ( $i = 1, \dots, m$ ) найдется  $v_i < 0$ , то, взяв вектор (принадлежащий  $A$ ), у которого все координаты равны 0, кроме  $i$ -й координаты, равной 1, мы придем к противоречию с (1.14). Таким образом,  $v_i \geq 0$  ( $i = 1, \dots, m$ ).

Положив  $u_i^* = v_i/v_0$  ( $i = 1, \dots, m$ ), из (1.14) получаем

$$a_0 + \sum_{i=1}^m u_i^* a_i \geq 0 \quad \forall a \in A; \quad (1.15)$$

при этом  $u_i \geq 0$  ( $i = 1, \dots, m$ ). Возьмем произвольное  $x \in X$  и рассмотрим вектор  $a = \{f_0(x) - f_0(x^*), f_1(x), \dots, f_m(x)\}$ . Поскольку  $a \in A$ , имеем

$$f_0(x) - f_0(x^*) + \sum_{i=1}^m u_i^* f_i(x) \geq 0 \quad \forall x \in X,$$

или

$$f_0(x^*) \leq f_0(x) + \sum_{i=1}^m u_i^* f_i(x) \quad \forall x \in X.$$

Далее,  $\sum_{i=1}^m u_i^* f_i(x^*) \geq 0$ . Так как  $u_i^* \geq 0$ ,  $f_i(x^*) \leq 0$  ( $i = 1, \dots, m$ ), то отсюда вытекает  $u_i^* f_i(x^*) = 0$  ( $i = 1, \dots, m$ ).

Таким образом, мы показали существование неотрицательного вектора  $u^* = \{u_1^*, \dots, u_m^*\}$  такого, что

$$f_0(x^*) + \sum_{i=1}^m u_i f_i(x^*) \leq f_0(x^*) + \sum_{i=1}^m u_i^* f_i(x^*) \leq f_0(x) + \sum_{i=1}^m f_i(x) u_i^*,$$

или, в краткой записи,

$$\mathcal{L}(x^*, u) \leq \mathcal{L}(x^*, u^*) \leq \mathcal{L}(x, u^*) \quad \forall x \in X, u \geq 0.$$

Теорема доказана.

Так как при  $u \geq 0$   $\mathcal{L}(x, u)$  представляет собой выпуклую по  $x$  функцию (как линейная комбинация выпуклых функций с неотрицательными коэффициентами), а по  $u$  — линейную, то теорему Куна—Таккера можно рассматривать как вариант теоремы о седловой точке для выпукло-вогнутых функций.

Добавим к ограничениям (1.12) систему линейных уравнений в форме

$$Ax - b = 0, \quad (1.16)$$

где  $A$  — матрица размера  $p \times n$ . При этом условие Слейтера нарушается, однако модифицированная теорема Куна—Таккера остается в силе.

В самом деле, пусть  $\bar{X} = X \cap L$ , где  $L$  — линейное многообразие, высекаемое ограничениями (1.16). Если найдется точка  $x \in \bar{X}$  такая, что  $f_i(x) < 0$  ( $i = 1, \dots, m$ ), то теорема Куна—Таккера примет следующую форму:

$$\mathcal{L}(x^*, u) \leq \mathcal{L}(x^*, u^*) \leq \mathcal{L}(x, u^*), \quad x \in \bar{X}, \quad u \geq 0. \quad (1.17)$$

Из (1.17) видно, что проекция некоторого субградиента  $g_{\mathcal{L}}^x$  по  $x$  на множество  $\bar{X}$  от функции  $\mathcal{L}(x, u^*)$  в точке  $x^*$  равна 0, откуда следует существование  $p$ -мерного вектора  $v = v^*$  такого, что  $g_{\mathcal{L}}^x(x^*) + A^*v^* = 0$ .

Рассмотрим  $\tilde{\mathcal{L}}(x, u, v) = \mathcal{L}(x, u) + (v, Ax - b)$ . Тогда условие (1.17) переписется следующим образом:

$$\tilde{\mathcal{L}}(x^*, u, v) \leq \tilde{\mathcal{L}}(x^*, u^*, v^*) \leq \mathcal{L}(x, u^*, v^*), \quad u \geq 0, \quad x \in X.$$

Таким образом, получаем модифицированную теорему Куна—Таккера при наличии линейных ограничений в форме равенств.

**Теорема 1.9.** Пусть для задачи выпуклого программирования: найти

$$\min_{x \in X} \{f_0(x) : f_i(x) \leq 0, \quad i = 1, \dots, m, \quad Ax = b\}$$

выполняется следующее условие:  $\exists \bar{x} \in X \cap L$ ,  $L = \{x : Ax = b\}$ , такое, что  $f_i(\bar{x}) < 0$ . Тогда, для того чтобы  $x^*$  была оптимальной точкой этой задачи, необходимо и достаточно существование  $p$ -мерного вектора  $v^*$  и  $m$ -мерного вектора  $u^* \geq 0$  таких, что

$$\tilde{\mathcal{L}}(x^*, u, v) \leq \tilde{\mathcal{L}}(x^*, u^*, v^*) \leq \tilde{\mathcal{L}}(x, u^*, v^*), \quad x \in X, \quad u \geq 0,$$

где

$$\tilde{\mathcal{L}}(x, u, v) = f_0(x) + \sum_{i=1}^m u_i f_i(x) + (v, Ax - b).$$

Рассмотрим параметрическое семейство задач выпуклого программирования: найти

$$\inf \{f_0(x) : x \in X, \quad f_i(x) \leq b_i, \quad i = 1, \dots, m\}, \quad (1.18)$$

где  $b = \{b_1, \dots, b_m\}$  — вектор параметров правой части ограничений. Пусть  $\Phi(\bar{b})$  — оптимальное значение целевой функции,  $x(\bar{b})$  — некоторый оптимальный вектор задачи (1.18), соответствующие значению параметра  $b = \bar{b}$ . Пусть, далее,  $X$  — компактное множество,

$f_i(x)$  выпуклы и непрерывны на  $X$ . Тогда, если задача (1.18) имеет допустимое решение, то она имеет и оптимальное решение  $x(b) \in X$ . Пусть  $B \subset E^m$  — множество векторов  $b$ , для которых существует допустимое решение  $x(b)$ . Покажем, что это множество является выпуклым, а функция  $\Phi(b)$  — выпуклая функция на множестве  $B$ .

Пусть  $b_1, b_2 \in B$  и  $x_1, x_2$  — допустимые решения задачи (1.18) при  $b = b_1$  и  $b = b_2$  соответственно;  $\alpha_1, \alpha_2 \geq 0$ ,  $\alpha_1 + \alpha_2 = 1$ . Покажем, что  $x_3 = \alpha_1 x_1 + \alpha_2 x_2$  — допустимое решение задачи (1.18) при  $b_3 = \alpha_1 b_1 + \alpha_2 b_2$ . Действительно,

$$f_i(x_3) = f_i(\alpha_1 x_1 + \alpha_2 x_2) \leq \alpha_1 f_i(x_1) + \alpha_2 f_i(x_2) \leq \alpha_1 b_{i1} + \alpha_2 b_{i2} = b_{i3}, \quad i = 1, \dots, m.$$

Таким образом,  $x_3$  — допустимое решение при  $b = b_3$ , и множество  $B$  является выпуклым.

Далее,

$$\Phi(b_3) = f_0(x(b_3)) \leq f_0(\alpha_1 x(b_1) + \alpha_2 x(b_2)) \leq \alpha_1 f_0(x(b_1)) + \alpha_2 f_0(x(b_2)) = \alpha_1 \Phi(b_1) + \alpha_2 \Phi(b_2),$$

т.е.  $\Phi(b)$  — выпуклая функция на  $B$ .

При доказательстве теоремы Куна—Таккера строилось множество  $A$ . Это множество можно рассматривать как надграфик функции  $\Phi(b) - \Phi(0)$ ,  $x^* = x(0)$ .

Как мы знаем, проекция на подпространство аргументов вектора внешней нормали опорной гиперплоскости к надграфику выпуклой функции  $f(x)$ , проведенной через точку  $\{\bar{x}, f(\bar{x})\}$ , который нормирован таким образом, что проекция на ось значений функции равна  $-1$ , дает субградиент  $f(x)$  в точке  $\bar{x}$ .

Из (1.15) следует, что если при  $b = 0$  в задаче (1.18) выполняется условие Куна—Таккера, то внешняя нормаль опорной гиперплоскости к множеству  $A$ , проведенная через точку  $0$ , имеет координаты  $\{-u_i\}_{i=1}^m, -1$ . Следовательно, вектор  $\{-u_i\}_{i=1}^m$  является субградиентом функции  $\Phi(b)$  в точке  $0$ . Заменяя вектор  $f(x) = \{f_i(x)\}_{i=1}^m$  на  $\{f_i(x) - b_i\}_{i=1}^m$  и подытоживая сказанное, получаем следующую теорему.

**Теорема 1.10.** Пусть задано параметрическое семейство задач (1.18), где  $X$  — выпуклое компактное множество,  $f_i(x)$  ( $i = 1, \dots, m$ ) — выпуклые функции, непрерывные на  $X$ . Тогда функция  $\Phi(b)$  является выпуклой, причем если при  $b = \bar{b}$  в задаче (1.18) выполняются условия теоремы Куна—Таккера, то субградиент функции  $\Phi(b)$  в точке  $\bar{b}$  имеет вид  $g_\Phi(\bar{b}) = \{-u_i^*(\bar{b})\}_{i=1}^m$ , где  $\{u_i^*(b)\}_{i=1}^m = u^*(b)$  — вектор оптимальных множителей Лагранжа в задаче (1.18) при  $b = \bar{b}$ .

Теоремы о множителях Лагранжа (или о двойственных оценках) типа теоремы Куна—Таккера могут служить основой различных схем декомпозиции, применяемых в математическом программировании. Интересное развитие в [32] получила теория двойственности для

несобственных задач оптимизации (несовместных или с неограниченным значением функционала).

С теоремой Куна—Таккера тесно связаны результаты по точным негладким функциям штрафа, позволяющие свести задачу выпуклого программирования к задаче безусловной минимизации [30, 110]. Рассматривается задача выпуклого программирования: найти

$$\min_{x \in E^n} \{f_0(x) : f_i(x) \leq 0, i = 1, \dots, m\}. \quad (1.19)$$

Пусть множество оптимальных решений  $X^*$  непусто,  $f_0(x^*) = f_0^*$  при  $x^* \in X^*$  и пара векторов  $\{x^*, \lambda^*\}$  образует седловую точку функции Лагранжа

$$\mathcal{L}(x, \lambda) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x), \quad \lambda \geq 0.$$

Образует штрафные функции двух типов:

$$S_1(x) = f_0(x) + \sum_{i=1}^m \psi_i(f_i(x)), \quad (1.20)$$

$$S_2(x) = f_0(x) + \varphi(\max_{1 \leq i \leq m} f_i(x)). \quad (1.21)$$

**Теорема 1.11.** Пусть  $\psi_i(t)$ ,  $\varphi(t)$  — одномерные выпуклые функции, принимающие при  $t \leq 0$  значение 0. Если  $\psi'_i(+0) > \lambda_i^*$  ( $i = 1, \dots, m$ ), то область минимумов задачи минимизации  $S_1(x)$  совпадает с  $X^*$ . Если  $\varphi'(+0) > \sum_{i=1}^m \lambda_i^*$ , то область минимумов задачи минимизации  $S_2(x)$  также совпадает с  $X^*$ .

**Доказательство.** Так как для любой допустимой точки  $x$   $f_0(x) = S_1(x) = S_2(x)$ , то для доказательства теоремы достаточно показать, что если  $x$  не является допустимой точкой, то  $S_1(\bar{x}) > f_0^*$  ( $S_2(\bar{x}) > f_0^*$ ). Пусть  $I^+(\bar{x})$  — непустое множество индексов  $i$ , для которых  $f_i(\bar{x}) > 0$ . Тогда

$$\begin{aligned} f_0^* \leq \mathcal{L}(\bar{x}, \lambda^*) &\leq f_0(\bar{x}) + \sum_{i \in I^+(\bar{x})} \lambda_i^* f_i(\bar{x}) < \\ &< f_0(\bar{x}) + \sum_{i \in I^+(\bar{x})} \psi'_i(+0) f_i(\bar{x}) \leq f_0^* + \sum_{i \in I^+(\bar{x})} \psi_i(f_i(\bar{x})) = S_1(\bar{x}). \end{aligned}$$

Аналогично получаем

$$\begin{aligned} f_0^* \leq \mathcal{L}(\bar{x}, \lambda^*) &= f_0(\bar{x}) + \sum_{i=1}^m \lambda_i^* f_i(\bar{x}) \leq f_0(\bar{x}) + \left( \sum_{i=1}^m \lambda_i^* \right) \max_{1 \leq i \leq m} f_i(\bar{x}) < \\ &< f_0(\bar{x}) + \varphi'(+0) \max_{1 \leq i \leq m} f_i(\bar{x}) \leq f_0(\bar{x}) + \varphi(\max_{1 \leq i \leq m} f_i(\bar{x})) = S_2(\bar{x}). \end{aligned}$$

Таким образом, если точка  $\bar{x}$  не является допустимой, то

$$f_0^* < S_1(\bar{x}), f_0^* < S_2(\bar{x}).$$

Теорема о точных негладких функциях штрафа доказана.

Практическое значение теоремы 1.11 весьма велико. Дело в том, что обычные гладкие функции штрафа, например с квадратичным штрафом, вида

$$S(x) = f_0(x) + \sum_{i=1}^m S_i(\max\{0, f_i(x)\})^2 \quad (1.22)$$

при минимизации при любых конечных положительных  $\{S_i\}_{i=1}^m$  дают лишь приближение к оптимальному решению задачи (1.19). Для получения решения с большой точностью при использовании штрафных функций вида (1.22) нужно работать с большими коэффициентами штрафа, при этом минимизируемая функция становится плохо обусловленной. Для борьбы с этими трудностями предложен подход, основанный на модифицированных функциях Лагранжа, однако при этом задачу безусловной минимизации нужно решать многократно [23].

При использовании точных гладких функций штрафа, как правило, удается, используя априорные оценки множителей Лагранжа, работать с умеренными значениями штрафных параметров. Далее, субградиентные методы негладкой оптимизации с растяжением пространства типа  $r$ -алгоритмов оказались устойчивыми по отношению к плохо обусловленным задачам. Это позволяет практически эффективно использовать метод негладких штрафов при решении сложных задач выпуклого программирования [15, 97].

### § 3. Схемы декомпозиции по переменным

Одна из основных схем декомпозиции — разложение задачи по переменным. Она состоит в том, что часть переменных фиксируется и решается локальная задача оптимизации относительно остальных переменных. Координирующая задача состоит в том, чтобы подобрать оптимальные значения для фиксированных переменных, т. е. такие, присоединив к которым оптимальные значения оставшихся переменных соответствующей локальной задачи, получим оптимальное решение первоначальной задачи.

Рассмотрим этот вопрос применительно к задачам выпуклого программирования следующего вида: найти

$$\min \{f_0(x, y) : f_i(x, y) \leq 0, i = 1, \dots, m\}. \quad (1.23)$$

где  $x \in E^n$ ,  $y \in E^p$  — две группы переменных;  $z = \{x, y\}$  —  $(n+p)$ -мерный вектор всех переменных задачи, функции  $f_\nu(z)$  ( $\nu = 0, 1, \dots, m$ ) выпуклы по совокупности переменных, определены на  $E^{n+p}$  и ограничения вырезают в  $E^{n+p}$  компактное множество.



Зафиксируем  $x = \bar{x}$  и рассмотрим задачу  $T(\bar{x})$  выпуклого программирования относительно  $y$ : найти

$$\min_y \{f(\bar{x}, y) : f_i(\bar{x}, y) \leq 0, i = 1, \dots, m\}. \quad (1.24)$$

Пусть  $\varphi(\bar{x})$  — оптимальное значение целевой функции в задаче  $T(\bar{x})$ . (Если эта задача не имеет решения, положим  $\varphi(\bar{x}) = +\infty$ .) Покажем, что  $\varphi(\bar{x})$  — выпуклая функция. Обозначим через  $y(\bar{x})$  произвольный оптимальный вектор задачи  $T(\bar{x})$ . Пусть  $\alpha_1, \alpha_2 \geq 0, \alpha_1 + \alpha_2 = 1$  и  $x_1, x_2$  — точки, в которых  $\varphi(\bar{x})$  принимает конечные значения. Тогда в силу выпуклости функций  $f_\alpha$  ( $\alpha = 0, 1, \dots$ )

$$\begin{aligned} f_i(\alpha_1 \bar{x}_1 + \alpha_2 \bar{x}_2, \alpha_1 y(\bar{x}_1) + \alpha_2 y(\bar{x}_2)) &\leq \\ &\leq \alpha_1 f_i(\bar{x}_1, y(\bar{x}_1)) + \alpha_2 f_i(\bar{x}_2, y(\bar{x}_2)) \leq 0, \quad i = 1, \dots, m, \end{aligned}$$

т. е. в точке  $\alpha_1 \bar{x}_1 + \alpha_2 \bar{x}_2$   $\varphi(x)$  принимает конечное значение. Далее,

$$\begin{aligned} \varphi(\alpha_1 \bar{x}_1 + \alpha_2 \bar{x}_2) &= \\ &= \min_y \{f_0(\alpha_1 \bar{x}_1 + \alpha_2 \bar{x}_2, y) : f_i(\alpha_1 \bar{x}_1 + \alpha_2 \bar{x}_2, y) \leq 0, i = 1, \dots, m\} \leq \\ &\leq f_0(\alpha_1 \bar{x}_1 + \alpha_2 \bar{x}_2, \alpha_1 y(\bar{x}_1) + \alpha_2 y(\bar{x}_2)) \leq \\ &\leq \alpha_1 f_0(\bar{x}_1, y(\bar{x}_1)) + \alpha_2 f_0(\bar{x}_2, y(\bar{x}_2)) \leq \alpha_1 \varphi(\bar{x}_1) + \alpha_2 \varphi(\bar{x}_2), \end{aligned}$$

откуда следует выпуклость функции  $\varphi(x)$ .

Субградиент функции  $\varphi(x)$  может быть получен в соответствии со следующей теоремой.

**Теорема 1.12.** Пусть для задачи  $T(\bar{x})$  вида (1.24) выполняется условие Слейтера. Обозначим вектор решения через  $y(\bar{x})$ , а вектор оптимальных множителей Лагранжа — через  $u^*(\bar{x}) = \{u_1^*(\bar{x}), \dots, u_m^*(\bar{x})\}$  (существование таких множителей вытекает из теоремы Куна — Таккера). Тогда точка  $\bar{x}$  является внутренней точкой области определения  $\varphi(x)$ , в ней существует субградиентное множество  $G_\varphi(\bar{x})$ , и обобщенный градиент может быть вычислен по формуле

$$g_\varphi(\bar{x}) = \hat{g}_{\mathcal{L}_x^*}(\bar{x}, y(\bar{x})), \quad (1.25)$$

где  $\mathcal{L}_x^*(x, y) = f_0(x, y) + \sum_{i=1}^m u_i^*(\bar{x}) f_i(x, y)$ ,  $\hat{g}_{\mathcal{L}_x^*}(\bar{x}, y(\bar{x}))$  —  $n$ -мерная  $x$ -составляющая такого субградиента функции  $\mathcal{L}^*(x, y)$ , взятого в точке  $\{\bar{x}, y(\bar{x})\}$ ,  $y$ -составляющая равна 0.

Доказательство. Так как в точке  $\bar{x}$  для задачи (1.24) выполняется условие Слейтера, то найдется такая точка  $\bar{y}$ , что  $f_i(\bar{x}, \bar{y}) < 0$  ( $i = 1, \dots, m$ ). Из непрерывности  $f_i$  по  $x, y$  следует существование окрестности точки  $(\bar{x}, \bar{y}) \in E^{n+p}$ , для всех точек которой выполняется  $f_i(x, y) < 0$  ( $i = 1, \dots, m$ ). Поэтому существует окрестность  $S(\bar{x}) \subset E^n$  точки  $\bar{x}$ , для всех точек которой функция  $\varphi(x)$  определена, и задача  $T(x)$  удовлетворяет условию Слейтера. Пусть  $\tilde{x} \in S(\bar{x})$ . Используя теорему Куна — Таккера, получаем следующие соотношения:

$$\begin{aligned} \varphi(\tilde{x}) - \varphi(\bar{x}) &= \mathcal{L}_x^*(\tilde{x}, y(\tilde{x})) - \mathcal{L}_x^*(\bar{x}, y(\bar{x})) \geq \\ &\geq \mathcal{L}_x^*(\tilde{x}, y(\tilde{x})) - \mathcal{L}_x^*(\bar{x}, y(\bar{x})) \geq (g_{\mathcal{L}_x^*}^x(\bar{x}, y(\bar{x})), \tilde{x} - \bar{x}) = \\ &= (g_{\mathcal{L}_x^*}^x(\bar{x}, y(\bar{x})), \tilde{x} - \bar{x}) + (g_{\mathcal{L}_y^*}^y(\bar{x}, y(\bar{x})), \tilde{y} - \bar{y}). \end{aligned}$$

Так как в точке  $y(\bar{x})$  достигается минимальное значение задачи  $T(\bar{x})$ , можно выбрать такой субградиент  $g_{\mathcal{L}_x^*}^x(\bar{x}, y(\bar{x}))$ , у которого  $y$ -составляющая равна 0, т. е.  $g_{\mathcal{L}_x^*}^y(\bar{x}, y(\bar{x})) = 0$ . При этом  $\varphi(\tilde{x}) - \varphi(\bar{x}) \geq (g_{\mathcal{L}_x^*}^x(\bar{x}, y(\bar{x})), \tilde{x} - \bar{x})$  при всех  $\tilde{x} \in S(\bar{x})$ , откуда следует, что  $g_{\mathcal{L}_x^*}^x(\bar{x}, y(\bar{x}))$  является субградиентом функции  $\varphi(x)$  в точке  $\bar{x}$ .

Следствие. Если функции  $f_v(x, y)$  являются непрерывно дифференцируемыми ( $v=0, 1, \dots, m$ ), то в предположениях теоремы 1.12

$$g_\varphi(\bar{x}) = g_{\mathcal{L}_x^*}^x(\bar{x}, y(\bar{x})).$$

Это является следствием того, что  $\mathcal{L}_x^*$  также непрерывно дифференцируема, при этом  $y$ -составляющая ее градиента в точке  $(\bar{x}, y(\bar{x}))$  равна нулю.

Рассмотрим некоторые свойства функции  $\varphi(x)$  в предположении, что функции  $f_v(x, y)$  ( $v = 0, 1, \dots, m$ ) непрерывно дифференцируемы. Если в некоторой точке  $\bar{x}$ , являющейся внутренней точкой области определения, функции  $\varphi(x)$ ,  $y(\bar{x})$  и  $u^*(\bar{x})$  определены однозначно, то в этой точке  $\varphi(x)$  непрерывно дифференцируема. В противном случае в этой точке наблюдается разрыв градиента.

В случае задачи линейного программирования: найти

$$\min ((c, x) + (d, y)) \quad (1.26)$$

при ограничениях

$$Ax + By \leq f, \quad (1.27)$$

или

$$By \leq f - Ax, \quad (1.28)$$

$\varphi(x)$  является кусочно линейной выпуклой функцией. Неоднозначность  $u^*(x)$  наблюдается в точках, которые соответствуют изменению базисных переменных задачи (1.26), (1.28), рассматриваемой как задача минимизации по  $y$  при фиксированном  $x$ .

Схема разложения по переменным обычно применяется в том случае, когда задача минимизации по  $y$  (1.24) решается значительно проще, чем первоначальная задача (1.23). Наиболее типичен случай, когда при фиксированном  $x = \bar{x}$  задача (1.24) распадается на блоки, каждый из которых представляет собой задачу линейного программирования сравнительно небольшой размерности. Примером такой задачи может служить двухэтапная стохастическая задача линейного программирования, которая служит моделью последовательного принятия решений в условиях неопределенности, когда значения части переменных определяются на первом этапе принятия решений, а значения остальных, корректирующих переменных — после прояснения ситуации, когда становится известной реализация случайных компонент системы. Значения корректирующих переменных являются случайными величинами, зависящими от реализации случайных событий. Примеры двухэтапных стохастических задач транспортного типа будут приведены в § 7 гл. 3.

#### § 4. Схемы декомпозиции по ограничениям

Многие задачи линейного и выпуклого программирования, встречающиеся при решении задач производственно-транспортного планирования, обладают структурой, позволяющей использовать декомпозицию по ограничениям. Суть этого приема состоит в том, что ограничения задачи разбиваются на две части, в одной из которых находятся «простые» ограничения, а в другой — «сложные» (конечно, деление ограничений на «простые» и «сложные» в достаточной степени условно и сильно зависит от специфики конкретной задачи). «Сложные» ограничения с неопределенными множителями Лагранжа вводятся в целевую функцию, при фиксированных множителях Лагранжа получаем значительно упрощенную по сравнению с первоначальной «внутреннюю» (локальную) задачу. Координирующая («внешняя») задача состоит в оптимальном подборе множителей Лагранжа.

Перейдем к формальному описанию схемы декомпозиции по ограничениям. Пусть нужно решить задачу выпуклого программирования: найти

$$\min f_0(x) \quad (1.29)$$

при ограничениях

$$f_i(x) \leq 0, \quad i = 1, \dots, m, \quad (1.30)$$

$$x \in X, \quad (1.31)$$

где  $f_v(x)$  — выпуклые функции, определенные на всем  $E^n$ ,  $X$  — выпуклое, замкнутое и (без существенной потери общности) компактное множество.

Введем  $u = \{u_1, \dots, u_m\}$  — вектор множителей Лагранжа и функцию Лагранжа

$$\mathcal{L}(x, u) = f_0(x) + \sum_{i=1}^m u_i f_i(x).$$

Рассмотрим при фиксированном  $u \geq 0$  внутреннюю задачу: найти

$$\min_{x \in X} \mathcal{L}(x, u). \quad (1.32)$$

Эта задача является задачей выпуклого программирования. Оптимальное значение  $x$  в (1.32) обозначим  $x(u)$ ; соответствующее значение функции  $\mathcal{L}(x, u)$  обозначим  $\psi(u)$ , так что  $\psi(u) = \mathcal{L}(x(u), u)$ . Из предыдущего изложения известно, что  $\psi(u)$  — вогнутая функция; если  $x(u^0)$  — единственное оптимальное решение задачи (1.32) при  $u = u^0$ , то  $\psi(u)$  в точке  $u^0$  дифференцируема и

$$g_\psi(u^0) = \{f_i(x(u^0))\}_{i=1}^m.$$

Если же при  $u = u^0$  задача (1.32) имеет неоднозначное решение, то при любом  $x(u^0) \in X^*(u^0)$  ( $X^*(u)$  — множество оптимальных решений) получим суперградиент функции  $\psi(u)$  в точке  $u^0$  в виде

$$g_\psi(u^0) = \{f_i(x(u^0))\}_{i=1}^m. \quad (1.33)$$

Если первоначальная задача удовлетворяет условиям обобщенной теоремы Куна—Таккера, то  $\sup_{u \geq 0} \psi(u) = \psi^*$  совпадает с оптимальным значением  $f_0^*$  целевой функции в задаче (1.29) — (1.31) и достигается на оптимальных значениях вектора множителей Лагранжа  $u = u^*$ . Таким образом, оптимальные множители Лагранжа в задаче (1.29) — (1.31) можно получить, решая задачу отыскания  $\max_{u \geq 0} \psi(u)$

с использованием эффективных методов недифференцируемой оптимизации, описанных в следующей главе. Значение  $x(u^*)$ , если оно единственно, будет давать решение задачи (1.29) — (1.31). Если же имеется множество решений  $X^*(u^*)$ , то не обязательно все они удовлетворяют ограничениям  $f_i(x^*) \leq 0$  ( $i = 1, \dots, m$ ). Подмножество  $X^* \subset X^*(u^*)$ , которое удовлетворяет этим ограничениям, образует множество оптимальных планов задачи (1.29) — (1.31).

Рассмотрим случай, когда задача (1.29) — (1.31) является квазиоблочной задачей линейного программирования со связывающими ограничениями: найти

$$\min \sum_{i=1}^N (c_i, x_i) \quad (1.34)$$

при ограничениях

$$\sum_{i=1}^N A_i x_i \leq a, \quad (1.35)$$

$$B_i x_i \leq b_i, \quad i = 1, \dots, N. \quad (1.36)$$

Здесь  $x_i$  — векторы размерности  $n_i$  ( $i = 1, \dots, N$ ),  $N$  — число блоков,  $A_i$  — матрицы размера  $m \times n_i$ ,  $B_i$  — матрицы размера  $m_i \times n_i$ ,  $a$  —  $m$ -мерный вектор,  $b_i$  —  $m_i$ -мерные векторы ( $i = 1, \dots, N$ ). Пусть  $\lambda = \{\lambda_1, \dots, \lambda_m\}$  — вектор множителей Лагранжа связывающих ограничений (1.35). Применим к задаче (1.34) — (1.36) прием декомпозиции по ограничениям, причем в качестве «простых» ограничений выберем «блочные» ограничения (1.36). Пусть  $X_i$  — множество векторов  $x_i$ , удовлетворяющих (1.36) при фиксированном  $i$  (будем предполагать, что все эти множества непусты и компактны),  $X = X_1 \times \dots \times X_n$ ,  $x$  — вектор, составленный из компонент  $x_1, \dots, x_n$ . Функция Лагранжа принимает следующий вид:

$$\mathcal{L}(x, \lambda) = \sum_{i=1}^N (c_i, x_i) + \left( \lambda, \sum_{i=1}^N A_i x_i - a \right) = \sum_{i=1}^N (c_i + A_i^T \lambda, x_i) - (\lambda, a).$$

«Внутренняя» задача определения  $\psi(\lambda) = \min_{x \in X} \mathcal{L}(x, \lambda)$  распадается на  $N$  подзадач следующего вида: найти

$$\psi_i(\lambda) = \min_{x_i \in X_i} (c_i + A_i^T \lambda, x_i), \quad i = 1, \dots, N, \quad (1.37)$$

так что

$$\psi(\lambda) = \sum_{i=1}^N \psi_i(\lambda) - (\lambda, a).$$

Пусть  $x^*(\lambda) = \{x_1^*(\lambda), \dots, x_N^*(\lambda)\}$  — вектор оптимальных значений задач (1.37). Субградиент функции  $\psi$  в точке  $\lambda$  задается следующей формулой:

$$g_\psi(\lambda) = \sum_{i=1}^N A_i x_i^*(\lambda) - a. \quad (1.38)$$

Для получения оптимальных двойственных переменных  $\lambda^*$  задачи (1.34) — (1.36), соответствующих связывающим ограничениям (1.35), нужно решить задачу нахождения  $\sup_{\lambda \geq 0} \psi(\lambda)$ . Максимизация кусочно-линейной вогнутой функции  $\psi(\lambda)$  может быть эффективно произведена на одном из методов, описанных в следующей главе. При  $\lambda = \lambda^*$ , решая задачи (1.37) симплекс-методом, получаем для каждого блока  $i$   $x_i(\lambda^*)$  и соответствующий вектор двойственных переменных  $\mu_i^*$  ( $i = 1, \dots, N$ ). Вектор  $\Lambda^* = \{\lambda^*, \mu_1^*, \dots, \mu_N^*\}$  является оптимальным вектором двойственных переменных задачи (1.34) — (1.36). Легко заметить, что схема декомпозиции по ограничениям задачи (1.34) — (1.36) эквивалентна схеме декомпозиции по переменным для двойственной к ней задачи.

Для получения решения  $x^*$  задачи (1.34) — (1.36) при известном  $\Lambda^*$  разработано несколько процедур. Во-первых, используя принцип дополняющей нежесткости, можно редуцировать задачу (1.34) — (1.36) к задаче меньшего объема, отбросив те ограничения, у которых оптималь-

ные двойственные оценки равны 0. Далее, нужно приравнять нулю значения тех переменных, у которых при оптимальных значениях двойственных переменных соответствующее двойственное ограничение выполняется с  $\varepsilon$ -запасом при некотором  $\varepsilon > 0$ . Выбор  $\varepsilon$  зависит от точности решения двойственной задачи. Если после исключения указанных нулевых переменных в  $i$ -м блоке остались только базисные переменные, то эти переменные можно исключить, придав им те значения, которые получены при решении задачи (1.37) при  $\lambda = \lambda^*$ ; при этом связывающие ограничения нужно соответствующим образом подкорректировать.

Редуцированная задача, как правило, оказывается задачей гораздо меньшего размера, чем первоначальная. Для ее решения можно: 1) использовать стандартную процедуру симплекс-метода; в невырожденном случае процедура исключения переменных и ограничений, описанная выше, оставит лишь ограничения в форме равенств, причем число переменных будет равняться числу ограничений, так что дело сведется к решению системы линейных уравнений; 2) в более сложных случаях, когда редуцированная задача достаточно велика и оптимальные двойственные переменные  $\Lambda^*$  определены с большой точностью, а также в условиях, когда выгодно использовать оптимальный план с минимальным числом ненулевых компонент (например, в задачах загрузки оборудования), применяется комбинация итеративной схемы получения двойственных переменных в сочетании со схемой декомпозиции Розена [24] на последней стадии решения задачи. Такой способ использован в ППП ПЛАНЕР при разработке модуля решения квазиблочной задачи линейного программирования [47]. Другой способ получения решения прямой задачи по решению двойственной описан в работе [7] (схема усреднения). Для решения многих задач производственно-транспортного планирования эффективным оказался метод  $\varepsilon$ -квадратичного сглаживания, описанный в гл. 3, § 5.

## § 5. Важные классы недифференцируемых функций. Обобщенные градиентные множества

При рассмотрении широкого круга задач производственно-транспортного планирования модели выпуклого и линейного программирования занимают видное место, однако на практике возникают и более сложные, часто многоэкстремальные задачи, в которых целевые функции и ограничения выражаются с помощью невыпуклых и негладких функций. С другой стороны, многие алгоритмы минимизации негладких выпуклых функций могут быть применены для нахождения локальных экстремумов более общих классов функций. В связи с этим возникает потребность в исследовании достаточно широких классов функций, которые охватывали бы выпуклые и вогнутые функции, кусочно гладкие и др., и в то же время достаточно узких, чтобы для них можно было естественно ввести понятие обобщенного градиента и построить аналоги градиентных процессов для нахождения локального минимума.

Функция  $f(x)$ , определенная на  $n$ -мерном евклидовом пространстве  $E^n$ , называется *почти дифференцируемой*, если:

а) в любой ограниченной области  $f(x)$  удовлетворяет условию Липшица;

б)  $f(x)$  почти везде дифференцируема;

в) градиент  $f(x)$  непрерывен на том множестве  $M$ , где он определен.

Отметим, что условие б) не является независимым, а является следствием условия а).

*Почти-градиентом* функции  $f(x)$  в точке  $x$  называется вектор, являющийся предельной точкой некоторой последовательности градиентов  $g(x_1), g(x_2), \dots, g(x_k)$ , где  $\{x_k\}_{k=1}^{\infty}$  — последовательность точек, сходящихся к  $x$ , такая, что во всех точках этой последовательности  $f(x)$  дифференцируема.

Эти понятия были введены в работе [92]. Там же показано, что множество  $G_f(x)$  почти-градиентов почти дифференцируемой функции  $f(x)$  в точке  $x$  является непустым, ограниченным и замкнутым. В точке, где  $f(x)$  дифференцируема, почти-градиент совпадает с градиентом; для выпуклой функции почти-градиент является субградиентом и наоборот, крайние точки субградиентного множества являются почти-градиентами; для кусочно гладкой функции почти-градиент совпадает с градиентом к одному из примыкающих к данной точке кусков. Таким образом, в большинстве случаев при решении практических задач вычисление почти-градиента не вызывает особых трудностей.

*Обобщенным градиентным множеством*  $G_f(x)$  почти дифференцируемой функции  $f$  в точке  $x$  называется выпуклое замыкание субградиентного множества  $G_f(x)$ .

Для выпуклой (вогнутой) функции обобщенное градиентное множество совпадает с субградиентным (суперградиентным) множеством. В последние годы в литературе по оптимизации большой популярностью пользуется введенный Ф. Кларком [116] класс липшицевых функций, который несколько обобщает класс почти дифференцируемых функций и отличается от него тем, что не требуется выполнение свойства в). Обобщенное градиентное множество в точке  $x$   $\hat{G}_f(x)$  определяется для этого класса подобно тому, как оно было определено для почти дифференцируемых функций. Другие обобщения понятия субградиентного множества даны в работах Б. Н. Пшеничного [61], Р. Миффлина [161], В. Ф. Демьянова, А. М. Рубинова [29] и др.

**АЛГОРИТМЫ НЕДИФФЕРЕНЦИРУЕМОЙ ОПТИМИЗАЦИИ**

---

**§ 1. Введение**

Владение методами недифференцируемой оптимизации упрощает решение многих вопросов, связанных с оптимизацией сложных систем.

Во-первых, это дает возможность гибкого использования различных схем декомпозиции (по переменным, по ограничениям, по ресурсам и т. п.), получения двойственных оценок в задачах дискретного и смешанного дискретно-непрерывного программирования для некоторых классов многоэкстремальных задач.

Во-вторых, появляется возможность свободного использования точных негладких функций штрафа, позволяющих при конечных значениях штрафных параметров получать задачу безусловной минимизации, полностью эквивалентную первоначальной задаче выпуклого программирования. Использование штрафных функций в форме функций максимума особенно полезно при решении задач с большим числом ограничений, так как при этом на каждой итерации требуется не более двух вычислений субградиента (см. (1.21)).

В-третьих, это важно при построении достаточно адекватных моделей реальных систем, когда появляется необходимость использовать негладкие критериальные функции или негладкие ограничения. Так, например, в задачах распознавания и идентификации, в статистическом оценивании большую роль играют так называемые робастные оценки, обладающие определенными преимуществами по сравнению с оценками наименьших квадратов в том случае, когда вероятностные распределения помех могут изменяться в широких пределах. В этом случае удобно использовать для оценки параметров критерий минимума суммы модулей невязок. При решении многокритериальных задач, когда используется прием равномерных относительных уступок, возникают критерии типа функции максимума.

Далее, технико-экономические характеристики оптимизируемых объектов часто хорошо аппроксимируются кусочно гладкими функциями от неизвестных параметров, что опять порождает задачи, целевые функции и (или) ограничения которых выражаются с помощью негладких функций. Отсутствие эффективных методов негладкой оптимизации затрудняло в свое время решение указанных классов задач и заставляло либо изменять постановку задачи в ущерб адекватности модели реальности, либо использовать различные приемы сглаживания, что не всегда приводит к успеху. Дело в том, что при-



менение сглаживающих функций с малым параметром вызывает плохую обусловленность минимизируемой функции, и это ухудшает вычислительную устойчивость таких методов, как квазиньютоновские или методы сопряженных градиентов.

Вообще следует сказать, что эффективные методы негладкой оптимизации оказались на практике вполне конкурентоспособными с наиболее эффективными методами решения гладких плохо обусловленных задач (см. [57, 89, 97, 156]). Таким образом, область приложений методов негладкой оптимизации весьма широка; отсюда и большое внимание, которое уделяется в последние годы разработке вычислительных методов негладкой оптимизации.

Задачи производственно-транспортного планирования явились благодатной почвой для использования методов негладкой оптимизации. Специфика этих задач позволяет в большинстве случаев путем использования схем декомпозиции и применения негладких штрафных функций сводить их к задачам безусловной минимизации негладких выпуклых функций умеренной размерности и достаточно эффективно их решать.

Среди методов негладкой оптимизации, которые в той или иной мере нашли практическое применение, можно выделить следующие классы.

1. Обобщенный градиентный спуск. Это класс немонотонных субградиентных процессов минимизации выпуклых функций, основанных на движении в направлении, которое дает уменьшение расстояния до точки минимума, если шаговый множитель достаточно мал. К преимуществам этих методов следует отнести простоту их реализации, к недостаткам — медленную, как правило, сходимость.

2. Обобщенные градиентные методы с растяжением пространства, в которых для ускорения сходимости используются преобразование метрики пространства. Это наиболее применяемый в настоящее время класс методов, показавший свою высокую эффективность при решении, в частности, задач производственно-транспортного планирования.

3. Использование секущих гиперплоскостей для аппроксимации графика функции или для последовательного уменьшения объема области локализации экстремума. К числу методов этого типа относятся и получивший широкую известность метод эллипсоидов и его модификации. С другой стороны, метод эллипсоидов можно отнести к методам обобщенного градиентного спуска с растяжением пространства.

4. Монотонные  $\varepsilon$ -субградиентные процессы, основанные на выборе устойчивого направления спуска за счет аппроксимации  $\varepsilon$ -субградиентного множества в текущей точке путем вычисления субградиентов в окрестности этой точки. Это широкий класс методов, теория которого находится в состоянии быстрого развития, однако его практическая эффективность пока исследована слабо [154, 155, 177]. Подробный обзор монотонных методов недифференцируемой оптимизации содержится в монографиях [28, 147], поэтому далее мы их рассматривать не будем.

5. Стохастические субградиентные процессы, которые обладают по сравнению с детерминированными более медленной сходимостью. Их область эффективного применения связана с задачами стохастического программирования, в которых вычисление стохастического субградиента требует гораздо меньше времени по сравнению с вычислением субградиента.

Перейдем к изложению основных результатов исследования указанных классов методов.

## § 2. Обобщенный градиентный спуск

Пусть на евклидовом пространстве  $E^n$  задана выпуклая функция  $f(x)$  и ставится задача определения  $\inf f(x)$ . Методом обобщенного градиентного спуска (ОГС) назовем процедуру построения минимизирующей последовательности  $\{x_k\}_{k=0}^{\infty}$ , где  $x_0$  — начальное приближение, а  $x_k$  ( $k = 1, 2, \dots$ ) строятся последовательно по следующей формуле:

$$x_{k+1} = x_k - h_k(x_k) g_f(x_k), \quad k = 0, 1, 2, \dots; \quad (2.1)$$

здесь  $g_f(x_k)$  — произвольный субградиент функции  $f(x)$  в точке  $x_k$ ,  $h_k(x_k)$  — шаговый множитель, зависящий, вообще говоря, от  $x_k$ .

Рассмотрим ОГС с постоянным шагом, когда

$$h_{k+1}(x_k) = h / \|g_f(x_k)\|, \quad h > 0.$$

**Теорема 2.1.** Пусть  $f(x)$  — выпуклая функция с непустым множеством минимумов  $M^*$ . Тогда для любых  $\varepsilon > 0$  и  $x^* \in M^*$  найдутся такие  $k = k^*$  и  $\bar{x}$ , что при использовании ОГС с постоянным шагом  $h$  будет выполняться следующее свойство:  $f(\bar{x}) = f(x_{k^*})$ , причем  $\|\bar{x} - x^*\| < h(1 + \varepsilon)/2$ .

**Доказательство.** Если для некоторого  $k^*$   $g_f(x_{k^*}) = 0$ , то  $f(x_{k^*}) = f(x^*)$  и в качестве  $\bar{x}$  можно взять  $x^*$ . Если для всех  $k = 0, 1, \dots$   $g_f(x_k) \neq 0$ , то

$$x_{k+1} = x_k - \frac{hg_f(x_k)}{\|g_f(x_k)\|}, \quad k = 0, 1, 2, \dots$$

Для любого  $x_k \in E^n$

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|x_k - x^* - hg_f(x_k) / \|g_f(x_k)\|\|^{-1}\|^2 = \\ &= \|x_k - x^*\|^2 + h^2 - 2h \left( x_k - x^*, \frac{g_f(x_k)}{\|g_f(x_k)\|} \right). \end{aligned} \quad (2.2)$$

Оценим выражение  $a_k(x^*) = \left( x_k - x^*, \frac{g_f(x_k)}{\|g_f(x_k)\|} \right)$  — расстояние от  $x^*$  до опорной гиперплоскости  $L_k = \{x: (g_f(x_k), x - x_k) = 0\}$ . Обозначим  $b_k(x^*) = \rho(x^*, U_k)$ , где  $U_k = \{x: f(x) = f(x_k)\}$ . Так как множество  $U$  и точка  $x^*$  находятся по одну сторону от  $L_k$  и отрезок, проведенный из  $x^*$  в произвольную точку, принадлежащую  $L_k$ , пересекает  $U_k$ , то

$a_k(x^*) \geq b_k(x^*)$ . Из (2.2) получаем

$$\|x_{k+1} - x^*\| \leq \|x_k - x^*\|^2 + h^2 - 2hb_k(x^*). \quad (2.3)$$

Если для всех  $k = 0, 1, 2, \dots$   $b_k(x^*) \geq h(1 + \varepsilon)/2$ , то

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \varepsilon h^2 \leq \|x_0 - x^*\|^2 - \varepsilon(k+1)h^2$$

при  $k = 0, 1, 2, \dots$  и  $\|x_{k+1} - x^*\|^2 \xrightarrow[k \rightarrow \infty]{} -\infty$ . Но  $\|x_{k+1} - x^*\|^2 \geq 0$ .

Значит, найдется такое  $k^*$ , что

$$b_{k^*}(x^*) = \rho(x^*, U_{k^*}) \leq h(1 + \varepsilon)/2.$$

Если теперь положить  $\bar{x} = \operatorname{argmin}_{x \in U_{k^*}} \|x - x^*\|$ , то все утверждения

теоремы выполняются.

Рассмотрим некоторые следствия доказанной теоремы.

**Следствие 1.** В условиях теоремы 2.1 для произвольного  $\delta > 0$  существует такое  $h_\delta > 0$ , что при применении метода ОГС с постоянным шагом  $h = h_\delta$  при любом  $x_0 \in E^n$  найдется либо такое  $k^*$ , что  $x_{k^*} \in M^*$ , либо такая подпоследовательность  $k_1 < k_2 < k_3 < \dots$ , что  $f(x_{k_i}) - f^* < \delta$ , где  $f^* = \min_{x \in E^n} f(x)$ .

**Следствие 2.** Если для функции  $f(x)$  множество  $M^*$  содержит сферу радиуса  $r > h/2 > 0$ , то при использовании метода ОГС с постоянным шагом  $h$  найдется такое  $k^*$ , что  $x_{k^*} \in M^*$ .

Исследуем сходимость нескольких основных модификаций метода ОГС. Из доказательства теоремы 2.1 видно, что уменьшение расстояния до точек области минимумов на данном шаге гарантируется только в том случае, если мы находимся за пределами некоторой окрестности области минимумов, зависящей от шага  $h$ . Поэтому, чтобы получить обычные теоремы сходимости, нужно требовать, чтобы  $h_k$  стремилось к нулю не слишком быстро. В частности, если ряд

$\sum_{k=1}^{\infty} h_k$  будет сходящимся, то последовательность  $\{x_k\}_{k=0}^{\infty}$  будет сходя-

диться, но не обязательно к точке, принадлежащей  $M^*$ . Так мы приходим к ставшим уже классическими условиям:  $h_k > 0$ ;  $h_k \xrightarrow[k \rightarrow \infty]{} 0$ ;

$\sum_{k=1}^{\infty} h_k = +\infty$ .

Имеется несколько вариантов доказательства теоремы о сходимости метода ОГС [33, 55]. Все они основаны на изучении поведения последовательности  $\{\rho_k\}_{k=0}^{\infty}$ , где  $\rho_k = \min_{y \in M^*} \|x_k - y\|$ . Наиболее общий

результат (для случая выпуклых функций, определенных в гильбертовом пространстве, когда минимизация производится при наличии ограничений) получен Б. Т. Поляком [55]. Ю. М. Ермольев [33] проводит доказательство теоремы о сходимости для конечномерного случая. Приведем конструктивный вариант доказательства теоремы о сходимости ОГС.

**Теорема 2.2.** Пусть  $f(x)$  — выпуклая функция, определенная на  $E^n$ , с ограниченной областью минимумов  $M^*$ ,  $\{h_k\}$  ( $k = 1, 2, \dots$ ) — последовательность положительных чисел, обладающая свойствами:

$$\lim_{k \rightarrow \infty} h_k = 0; \quad \sum_{k=1}^{\infty} h_k = +\infty.$$

Тогда последовательность  $\{x_k\}$  ( $k=0, 1, \dots$ ), образованная по формуле

$$x_{k+1} = x_k - h_{k+1} \frac{g_f(x_k)}{\|g_f(x_k)\|}, \quad (2.4)$$

при произвольном  $x_0 \in E^n$  обладает одним из следующих свойств: либо найдется такое  $k = \bar{k}$ , что  $x_{\bar{k}} \in M^*$ , либо

$$\lim_{k \rightarrow \infty} \min_{y \in M^*} \|x_k - y\| = 0, \quad \lim_{k \rightarrow \infty} f(x_k) = \min_{x \in E^n} f(x) = f^*.$$

**Доказательство.** Пусть  $x^* \in M^*$ . Как и при доказательстве теоремы 2.1, получаем при  $x_k \notin M^*$

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 + h_{k+1}^2 - 2h_{k+1}b_k(x^*). \quad (2.5)$$

Рассмотрим множество  $\{x: f(x) \leq f^* + a\}$  и его границу  $U_{f^*+a}$  ( $a > 0$ ). Из условия теоремы следует, что  $M^*$  и  $U_{f^*+a}$  — ограниченные замкнутые множества, причем  $M^* \cap U_{f^*+a} = \emptyset$ . В силу этого существует  $\min_{x \in U_{f^*+a}; y \in M^*} \|x - y\| = b(a) > 0$ . Так как  $\lim_{k \rightarrow \infty} h_k = 0$ , то найдется такое  $N_\varepsilon$ , что при  $k > N_\varepsilon$   $h_k < \varepsilon$ . Примем  $\varepsilon = b(a)$ . Для  $k > N_{b(a)}$  имеем  $h_k < b(a)$ . Если  $f(x_k) \geq f^* + a$ , то в силу (2.5)

$$\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2 \geq b(a)h_{k+1}.$$

Так как  $\sum_{k=1}^{\infty} h_k = \infty$ , то найдется  $N_a^{(1)} > N_{b(a)}$  такое, что  $f(x_{N_a^{(1)}}) \leq f^* + a$ .

Введем обозначение

$$d(a) = \max_{y \in U_{f^*+a}} \min_{x \in M^*} \|y - x\|.$$

Если  $f(x_k) \leq f^* + a$ , то  $\min_{x \in M^*} \|x_k - x\| \leq d(a)$  и  $\min_{x \in M^*} \|x_{k+1} - x\| \leq d(a) + h_{k+1}$ . Если  $f(x_k) > f^* + a$  и  $k > N_{b(a)}$ , то

$$\min_{x \in M^*} \|x_{k+1} - x\| \leq \min_{x \in M^*} \|x_k - x\|.$$

Отсюда для  $k > N_a^{(1)}$

$$\min_{x \in M^*} \|x_k - x\| \leq d(a) + h(N_a^{(1)}),$$

где  $h(N_a^{(1)}) = \max_{k > N_a^{(1)}} h_k$ . Так как  $d(a) \rightarrow 0$  при  $a \rightarrow 0$ , то для произвольного  $\delta > 0$  можно выбрать такое  $a(\delta)$ , что  $d(a(\delta)) \leq \delta/2$ , а затем — такой номер  $N_\delta$ , что  $f(x_{N_\delta}) \leq f^* + a(\delta)$  и для всех  $k > N_\delta$   $h_k \leq \delta/2$ . Тогда для всех  $k > N_\delta$

$$\min_{y \in M^*} \|x_k - y\| \leq \delta.$$

Это и доказывает соотношение  $\lim_{k \rightarrow \infty} \min_{y \in M^*} \|x_k - y\| = 0$ .

Из непрерывности функции  $f(x)$  следует, что

$$\lim_{k \rightarrow \infty} f(x_k) = f^*.$$

Аналогичная теорема может быть доказана и для другого способа регулировки шага (без нормирования градиента).

**Теорема 2.3.** Если в условиях теоремы 2.2 определить последовательность  $\{x_k\}_{k=0}^\infty$  формулой

$$x_{k+1} = x_k - h_{k+1} g_f(x_k), \quad k = 0, 1, \dots, \quad (2.6)$$

где  $x_0 \in E^n$  — некоторая начальная точка,  $h_k > 0$ ,  $\lim_{k \rightarrow \infty} h_k = 0$ ,  $\sum_{k=1}^\infty h_k = +\infty$ , то возможны такие случаи:

а) последовательность  $\{g_f(x_k)\}_{k=0}^\infty$  неограничена, сходимость отсутствует;

б) последовательность  $\{g_f(x_k)\}_{k=0}^\infty$  ограничена; тогда

$$\lim_{k \rightarrow \infty} \min_{x \in M^*} \|x_k - x\| = 0, \quad \lim_{k \rightarrow \infty} f(x_k) = f^*.$$

**Доказательство.** Приведем простой пример, когда реализуется случай а). Примем  $f(x) = x^4$ ,  $h_k = 1/k$ ,  $x_0 = 1$ . Получим  $g_f(x_0) = 4$ ,  $x_1 = -3$ ,  $g_f(x_1) = -108$ ,  $x_2 = 51$  и т. д. Легко проверить, что  $\lim_{k \rightarrow \infty} |g_f(x_k)| = +\infty$ .

Рассмотрим случай б). Если для некоторого  $k^*$   $g_f(x_{k^*}) = 0$ , то  $x_{k+1} = x_k$  для  $k \geq k^*$  и  $x_{k^*} \in M^*$ . Допустим, что  $g_f(x_k) \neq 0$  ( $k = 0, 1, \dots$ ). Рассмотрим  $\tilde{h}_k = h_k \|g_f(x_k)\|$ . Так как  $\{\|g_f(x_k)\|\}$  ограничена, то  $\lim_{k \rightarrow \infty} \tilde{h}_k = \lim_{k \rightarrow \infty} h_k = 0$ .

Если  $\sum_{k=0}^\infty \tilde{h}_k = +\infty$ , то, заменив (2.6) выражением  $x_{k+1} = x_k - h_{k+1} \frac{g_f(x_k)}{\|g_f(x_k)\|}$  и применив теорему 2.2, получаем требуемый результат.

Если  $\sum_{k=0}^\infty \tilde{h}_k < \infty$ , то последовательность  $\{x_k\}_{k=0}^\infty$  сходится к некоторой точке  $x^*$ . Покажем, что  $g_f(x^*) = 0$ . Предположим противное.

Тогда  $x^* \notin M^*$ , а это значит, что, начиная с некоторого  $\bar{k}$ ,  $\|g_f(x_k)\| \geq \varepsilon > 0$ . Но тогда  $\sum_{k=\bar{k}}^{\infty} \tilde{h}_k \geq \varepsilon \sum_{k=\bar{k}}^{\infty} h_k = +\infty$ , и мы пришли к противоречию.

Таким образом,  $g_f(x^*) = 0$ , т. е.  $x^* \in M^*$ . Теорема доказана.

Заметим, что если  $f(x)$  — кусочно линейная функция с конечным числом кусков, то последовательность  $\{g_f(x_k)\}_{k=0}^{\infty}$  всегда ограничена, т. е. выполняется случай б). Отметим также, что для заданного  $x_0$  всегда можно подобрать такое  $\delta > 0$ , что если  $\max_{k \geq 1} h_k \leq \delta$ , то будет

выполняться случай б). В самом деле, рассмотрим для некоторого  $a > 0$  множества  $U(a) = \{x: f(x) = f(x_0) + a\}$ ,  $S(a) = \{x: f(x) \leq f(x_0) + a\}$ . Так как  $f(x)$  определена на всем  $E^n$  и  $M^*$  ограничено, то  $U(a)$  и  $S(a)$  — компактные множества. Рассмотрим минимальный замкнутый шар  $S^*$  с центром в точке  $x^* \in M^*$ , содержащий  $S(a)$ , и пусть  $c = \max_{x \in S(a)} \|g_f(x)\|$ ,  $\rho_1 = \rho(U(a), U(a/2))$ ,  $\rho_2 = \rho(x^*, U(a/2))$ . Вы-

берем  $\delta = \min\{\rho_1/c, \rho_2/c\}$ . Если  $h_k \leq \delta$ , то  $\tilde{h}_k = h_k \|g_f(x_k)\| \leq \min\{\rho_1, \rho_2\}$  при  $x_k \in S^*$ . Покажем, что  $x_k \in S^*$  при всех  $k = 0, 1, \dots$ . В самом деле,  $x_0 \in S^*$ . Если  $x_k \in S(a) \setminus S(a/2)$ , то из (2.5) получаем  $\rho(x_{k+1}, x^*) \leq \rho(x_k, x^*)$  и  $x_{k+1} \in S^*$ . Если  $x_k \in S(a/2)$ , то, так как  $\|x_{k+1} - x_k\| \leq \rho_1$ ,  $x_{k+1} \in S(a) \subseteq S^*$ . Таким образом, при всех  $k$   $x_k \in S^*$ . Но тогда  $\|g_f(x_k)\| < c$ , и мы имеем случай б). Приведенные выше рассуждения позволяют модифицировать теорему 2.3 следующим образом.

**Теорема 2.4.** Если в условиях теоремы 2.2 определить последовательность  $\{x_k\}_{k=0}^{\infty}$  формулой

$$x_{k+1} = \begin{cases} x_k - h_{k+1} g_f(x_k) & \text{при } h_{k+1} \|g_f(x_k)\| \leq c, \\ x_0 & \text{в противном случае,} \end{cases}$$

где  $c > 0$  — некоторая константа,  $\lim_{k \rightarrow \infty} h_k = 0$ ,  $\sum_{k=1}^{\infty} h_k = +\infty$ , то для любого начального приближения  $x_0$

$$\lim_{k \rightarrow \infty} \min_{x \in M^*} \|x_k - x\| = 0, \quad \lim_{k \rightarrow \infty} f(x_k) = f^*.$$

Для доказательства следует лишь отметить, что так как  $h_k \rightarrow 0$ , то, начиная с некоторого  $\bar{k}$ , условие  $h_{k+1} \|g_f(x_k)\| \leq c$  будет выполняться для всех последующих  $k$ .

**Теорема 2.5.** Если множество минимумов  $M^*$  функции  $f(x)$  содержит сферу  $S_r$  радиуса  $r > 0$ ,  $h_k > 0$ ,  $\sum_{k=1}^{\infty} h_k = +\infty$  и  $\text{vrai sup } h_k <$

$< 2r$ , то для произвольного  $x_0 \in E^n$  при применении ОГС в форме (2.4) найдется  $k(x_0)$  такое, что  $x_{k(x_0)} \in M^*$ .

**Доказательство.** Пусть  $x^* \in M^*$  — центр сферы  $S_r$ . Тогда если  $x_k \notin M^*$  при  $k = 0, 1, \dots$ , то  $b_k(x^*) > r$ . Используя (2.5), получаем

при  $k \geq m$

$$0 \leq \|x_{k+1} - x^*\|^2 \leq \|x_m - x^*\|^2 - \sum_{i=m}^k h_{i+1} (2b_i(x^*) - h_{i+1}). \quad (2.6)$$

Так как  $\forall i \sup h_k < 2r$ , то найдутся  $k'$  и  $\delta > 0$  такие, что  $2b_i(x^*) - h_{i+1} \geq 2r - h_{i+1} \geq \delta > 0$  при  $i > k'$ . Из (2.6) получаем

$$0 \leq \|x_{k'} - x^*\|^2 - \delta \sum_{i=k'}^{\infty} h_i.$$

Приходим к противоречию, так как по предположению  $\sum_{i=k'}^{\infty} h_i = +\infty$ .

Таким образом, найдется  $x_{k(x_0)} \in M^*$ . Теорема доказана.

Б. Т. Поляк [55] доказал следующую теорему о сходимости метода проекции обобщенного градиента для случая минимизации выпуклых функционалов, определенных в гильбертовом пространстве, при ограничениях довольно общего вида.

Пусть  $f(x)$  — выпуклый функционал, определенный на выпуклом множестве  $Q$  гильбертова пространства  $E$ ,  $Q = Q_1 \cap Q_2$ , где  $Q_1 = \{x: \varphi(x) \leq 0\}$ ,  $\varphi(x)$  — выпуклый на  $E$  функционал,  $Q_1$  имеет внутренние точки (множество внутренних точек обозначим  $Q_1^0$ ),  $Q_2$  — выпуклое замкнутое множество, причем  $Q_1^0 \cap Q_2 \neq \emptyset$ . Обозначим через  $P_{Q_2}(x)$  проекцию точки  $x$  на  $Q_2$ , т. е.

$$\|x - P_{Q_2}(x)\| = \inf_{y \in Q_2} \|x - y\|.$$

Метод обобщенного градиентного спуска будет заключаться в построении последовательности  $\{x_k\}_{k=1}^{\infty}$  по формуле

$$x_{k+1} = P_{Q_2}(x_k + \Delta(x_k)), \quad (2.7)$$

где  $\Delta(x_k)$  — произвольный опорный функционал к множеству  $\{x: f(x) \leq f(x_k)\}$ , если  $x_k \in Q_1$ , и к множеству  $\{x: \varphi(x) \leq \varphi(x_k)\}$ , если  $x_k \notin Q_1$  (линейный функционал называется опорным к  $R \subset E$  в точке  $x \in R$ , если  $(c, x) \leq (c, y)$  для всех  $y \in R$ ).

Теорема 2.6. Если  $\lim_{k \rightarrow \infty} \Delta(x_k) = 0$ ,  $\sum_{k=0}^{\infty} \|\Delta(x_k)\| = +\infty$ , то при любом  $x_0$  в методе (2.7) найдется подпоследовательность  $\{x_{n_k}\}$  такая, что  $x_{n_k} \in Q$ ,  $\lim_{k \rightarrow \infty} f(x_{n_k}) = f^* = \inf_{x \in Q} f(x)$ .

Доказательство. Выберем произвольное  $\alpha > f^*$  и обозначим  $S_\alpha = \{x: x \in Q_1; f(x) \leq \alpha\}$ ,  $S_\alpha^0$  — множество внутренних точек  $S_\alpha$ . Поскольку  $Q_1^0 \cap Q_2 \neq \emptyset$ , существует  $\tilde{x} \in Q_2 \cap S_\alpha^0$ . Выберем  $\rho > 0$  так, чтобы  $x \in S_\alpha$  для всех  $\|x - \tilde{x}\| \leq \rho$ . Предположим, что  $x_k \notin S_\alpha$  для всех  $k$ . Тогда если  $x_k \notin Q_1$ , то  $S_\alpha \subset Q_1 \subset \{x: \varphi(x) \leq \varphi(x_k)\}$ ; если же

$x_k \in Q_1$ , то  $S_\alpha \subset \{x: f(x) \leq f(x_k)\}$ , так что во всех случаях  $(\Delta(x_k), x_k) \leq (\Delta(x_k), \tilde{x} - \rho \Delta(x_k) / \|\Delta(x_k)\|)$ . Отсюда

$$\begin{aligned} \|x_{k+1} - \tilde{x}\|^2 &= \|P_{Q_2}(x_k + \Delta(x_k)) - \tilde{x}\|^2 \leq \|x_k + \Delta(x_k) - \tilde{x}\|^2 = \\ &= \|x_k - \tilde{x}\|^2 + \|\Delta(x_k)\|^2 + 2\left(\Delta(x_k), x_k - \tilde{x} + \rho \frac{\Delta(x_k)}{\|\Delta(x_k)\|}\right) - \\ &\quad - 2\rho \|\Delta(x_k)\| \leq \|x_k - \tilde{x}\|^2 + \|\Delta(x_k)\|^2 - 2\rho \|\Delta(x_k)\|. \end{aligned}$$

Выберем  $N$  так, чтобы  $\|\Delta(x_k)\| \leq \rho$  для всех  $k \geq N$ , и просуммируем полученные выше неравенства от  $k = N$  до  $k = N + m$ . Получим

$$\begin{aligned} 0 \leq \|x_{N+m+1} - \tilde{x}\|^2 &\leq \|x_N - \tilde{x}\|^2 + \sum_{k=N}^{N+m} \|\Delta(x_k)\| (\|\Delta(x_k)\| - 2\rho) \leq \\ &\leq \|x_N - \tilde{x}\|^2 - \rho \sum_{k=N}^{N+m} \|\Delta(x_k)\|, \quad m = 1, 2, \dots, \end{aligned}$$

что противоречит расходимости  $\sum_{k=1}^{\infty} \|\Delta(x_k)\|$ . Значит, для любого  $\alpha > f^*$

найдется  $x_k \in S_\alpha$ , т. е. существует подпоследовательность  $\{x_{n_k}\}_{k=1}^{\infty}$  такая, что  $\lim_{k \rightarrow \infty} f(x_{n_k}) = f^*$ . Теорема доказана.

При определенных дополнительных предположениях удастся получить варианты ОГС, сходящиеся со скоростью геометрической прогрессии.

**Теорема 2.7.** Пусть  $f(x)$  — выпуклая функция, определенная на  $E^n$ , и для всех  $x \in E^n$  при некотором  $\varphi$  ( $0 \leq \varphi < \pi/2$ ) выполняется неравенство

$$(g_f(x), x - x^*(x)) \geq \cos \varphi \|g_f(x)\| \cdot \|x - x^*(x)\|, \quad (2.8)$$

где  $x^*(x)$  — точка, принадлежащая множеству минимумов функции  $f(x)$  и лежащая на кратчайшем расстоянии от  $x$ . Тогда если при заданном  $x_0$  выбрать величину  $h_1$ , удовлетворяющую неравенству

$$h_1 \geq \begin{cases} \|x^*(x_0) - x_0\| \cos \varphi, & \pi/4 \leq \varphi < \pi/2, \\ \|x^*(x_0) - x_0\| / (2 \cos \varphi), & 0 \leq \varphi < \pi/4, \end{cases} \quad (2.9)$$

определить  $\{h_k\}_{k=1}^{\infty}$  в соответствии с рекуррентной формулой

$$h_{k+1} = h_k r(\varphi), \quad k = 1, 2, \dots, \quad (2.11)$$

где

$$r(\varphi) = \begin{cases} \sin \varphi, & \pi/4 \leq \varphi < \pi/2, \\ 1/(2 \cos \varphi), & 0 \leq \varphi < \pi/4, \end{cases} \quad (2.12)$$

$$(2.13)$$

и вычислить  $\{x_k\}_{k=1}^{\infty}$  по формуле

$$x_{k+1} = x_k - h_{k+1} g_f(x_k) / \|g_f(x_k)\|,$$



то либо при некотором  $k^*$   $g_f(x_{k^*}) = 0$  и  $x_{k^*}$  принадлежит области минимумов, либо при всех  $k = 0, 1, 2, \dots$  выполняется неравенство

$$\|x_k - x^*(x_k)\| \leq \begin{cases} h_{k+1}/\cos \varphi, & \pi/4 \leq \varphi < \pi/2, \\ 2 \cos \varphi \cdot h_{k+1}, & 0 \leq \varphi < \pi/4. \end{cases} \quad (2.14)$$

**Доказательство.** Рассмотрим сначала случай  $\pi/4 \leq \varphi < \pi/2$ . Для  $k = 0$  неравенство (2.14) выполняется. Пусть оно выполняется для  $k = p$  и  $g_f(x_p) \neq 0$ . Докажем его для  $k = p + 1$ :

$$\begin{aligned} \|x_{p+1} - x^*(x_{p+1})\|^2 &\leq \|x_{p+1} - x^*(x_p)\|^2 = \|x_p - h_{p+1} \frac{g_f(x_p)}{\|g_f(x_p)\|} - x^*(x_p)\|^2 = \\ &= \|x_p - x^*(x_p)\|^2 - \frac{2h_{p+1}}{\|g_f(x_p)\|} (x_p - x^*(x_p), g_f(x_p)) + h_{p+1}^2. \end{aligned}$$

Воспользовавшись последовательно неравенствами (2.8), (2.14) и учитывая, что при  $\pi/4 \leq \varphi < \pi/2$   $1 - 2 \cos^2 \varphi \geq 0$ , получаем

$$\begin{aligned} \|x_{p+1} - x^*(x_{p+1})\|^2 &\leq \|x_p - x^*(x_p)\|^2 (1 - 2 \cos^2 \varphi) + h_{p+1}^2 \leq \\ &\leq \frac{h_{p+1}^2}{\cos^2 \varphi} (1 - 2 \cos^2 \varphi) + h_{p+1}^2 = h_{p+1}^2 \left( \frac{\sin \varphi}{\cos \varphi} \right)^2. \end{aligned}$$

Принимая во внимание (2.11), (2.12), получаем

$$\|x_{p+1} - x^*(x_{p+1})\|^2 \leq h_{p+2}^2 / \cos^2 \varphi.$$

Таким образом, для  $\pi/4 \leq \varphi < \pi/2$  теорема справедлива.

Рассмотрим случай  $0 \leq \varphi < \pi/4$ . Для  $k = 0$  (2.14) выполняется. Предположим, что оно выполняется для  $k = p$ ,  $g_f(x_p) \neq 0$ , и докажем его для  $k = p + 1$ .

Воспользовавшись (2.8), (2.14), (2.13), получаем

$$\begin{aligned} \|x_{p+1} - x^*(x_{p+1})\|^2 &\leq \\ &\leq \|x_p - x^*(x_p)\|^2 - 2h_{p+1} \cos \varphi \|x_p - x^*(x_p)\| + h_{p+1}^2 \leq \\ &\leq \|x_p - x^*(x_p)\|^2 - \|x_p - x^*(x_p)\|^2 + h_{p+1}^2 = \\ &= h_{p+1}^2 = (2 \cos \varphi h_{p+1})^2. \end{aligned}$$

Теорема доказана.

**З а м е ч а н и е.** Из доказательства очевидно, что для справедливости теоремы достаточно выполнения условия (2.8) лишь для точек последовательности  $\{x_k\}_{k=0}^{\infty}$ .

Таким образом, если угол  $\varphi$  заранее известен, то, регулируя шаг по формулам (2.11), (2.12), (2.13), можно получить сходимость к минимуму со скоростью геометрической прогрессии со знаменателем  $q = r(\varphi)$ .

В формуле (2.8)  $\cos \varphi$  характеризует степень вытянутости поверхности уровня функции  $f(x)$ . Если в некоторой окрестности минимума функции  $f(x)$  не существует  $\varphi < \pi/2$  такого, что для любого  $x$  из этой окрестности выполняется (2.8), то такую функцию будем называть *существенно овражной*. При минимизации существенно овражной

ных функций приведенный в теореме способ регулировки шаговых множителей неприменим. В этом случае нужно использовать универсальный способ выбора шаговых множителей, указанный в теореме 2.2.

Сформулируем теорему, аналогичную теореме 2.7, непосредственно в терминах, характеризующих степень «вытянутости» поверхностей уровня.

**Теорема 2.8.** Пусть выпуклая функция  $f(x)$  определена на  $E^n$ ,  $x^*$  — единственная точка минимума  $f$  и заданы начальное приближение  $x_0$  и числа  $\sigma$  и  $h_1$ , причем  $\sigma \geq \sqrt{2}$ ,  $h_1 \geq \|x_0 - x^*\|/\sigma$ . Рассмотрим множество  $Y = \{y : \|y - x^*\| \leq \sigma h_1\}$ . Если для любой пары точек  $x, z \in Y$  такой, что  $f(x) = f(z) \neq f(x^*)$ , выполняется условие

$$\|x - x^*\| \leq \sigma \|z - x^*\|, \quad (2.15)$$

то последовательность  $\{x_k\}_{k=0}^\infty$ , образованная с помощью рекуррентных формул

$$x_{k+1} = x_k - h_{k+1} g_f(x_k) / \|g_f(x_k)\|,$$

где  $h_{k+1} = h_k \sqrt{\sigma^2 - 1}/\sigma$ , сходится к  $x^*$  со скоростью геометрической прогрессии:

$$\|x_k - x^*\| \leq h_{k+1} \sigma, \quad (2.16)$$

за исключением случая, когда для некоторого  $k = \bar{k}$   $g_f(x_{\bar{k}}) = 0$ , т. е.  $x_{\bar{k}} = x^*$ .

**Доказательство.** При  $k = 0$  неравенство (2.16) выполняется,  $x_0 \in Y$ . Допустим, что (2.16) справедливо для  $k = p$ , и докажем его справедливость для  $k = p + 1$ . Предположим, что  $x_p \neq x^*$  и, так как  $x_p \in Y$ , выполняется соотношение (2.15). Далее,

$$(g_f(x_k), x_k - x^*) = a_k(x^*) \|g_f(x_k)\|,$$

где  $a_k(x^*)$  — проекция  $x_k - x^*$  на  $g_f(x_k)$ . Из доказательства теоремы 2.2 следует, что  $a_k(x^*) \geq b_k(x^*)$ , где  $b_k(x^*)$  — кратчайшее расстояние от  $x^*$  до гиперповерхности уровня  $f(x) = f(x_k)$ . Таким образом,  $(g_f(x_k), x_k - x^*) \geq b_k(x^*) \|g_f(x_k)\|$ , а из (2.15) вытекает  $b_k(x^*) \geq \|x_k - x^*\|/\sigma$ . Значит,

$$(g_f(x_k), x_k - x^*) \geq \|g_f(x_k)\| \cdot \|x_k - x^*\|/\sigma. \quad (2.17)$$

Введем обозначение  $\cos \varphi = 1/\sigma$ . Тогда условие (2.17) приобретает вид условия (2.8) из теоремы 2.7. Применим эту теорему. Учитывая, что при  $\sigma \geq \sqrt{2}$ ,  $\varphi \geq \pi/4$ ,  $\sin \varphi = \sqrt{\sigma^2 - 1}/\sigma$ , получаем  $\|x_k - x^*\| \leq h_{k+1} \sigma$ , что и требовалось доказать.

**Пример.** Рассмотрим частный случай — минимизацию положительно определенной квадратичной формы  $f(x) = (Ax, x)$ . Пусть  $\lambda$  — наименьшее собственное число оператора  $A$ ,  $\mu$  — наибольшее собственное число. Определим минимальное значение величины

$$\frac{(g_f(x), x)}{\|g_f(x)\| \cdot \|x\|} = \frac{(Ax, x)}{\|Ax\| \cdot \|x\|}, \quad \|x\| \neq 0.$$

Для этого зафиксируем произвольный несобственный вектор  $x_0$  и рассмотрим подпространство  $E(x_0)$  размерности два, порожденное векторами  $x_0$  и  $Ax_0$ . Обозначим через  $\tilde{A}$  матрицу квадратичной формы, определенной на  $E(x_0)$ , такую, что  $(\tilde{A}x, x) = (Ax, x)$  при  $x \in E(x_0)$ . Легко видеть, что оператор  $\tilde{A}$ , определенный на подпространстве  $E(x_0)$ , является положительно определенным оператором и его собственные числа  $\tilde{\lambda}, \tilde{\mu}$  (записанные в порядке неубывания) удовлетворяют неравенствам  $\lambda \leq \tilde{\lambda} \leq \tilde{\mu} < \mu$ . Пусть  $x_1, x_2$  — ортонормированная система собственных векторов оператора  $\tilde{A}$  в  $E(x_0)$  и  $\|x_0\| = 1$ . Представим  $x_0$  в виде  $x_0 = \alpha_1 x_1 + \alpha_2 x_2$ , причем  $\alpha_1^2 + \alpha_2^2 = 1$ . Тогда

$$\frac{(Ax_0, x_0)}{\|Ax_0\| \cdot \|x_0\|} = \frac{(\tilde{A}x_0, x_0)}{\|\tilde{A}x_0\| \cdot \|x_0\|} = \frac{\alpha_1^2 \tilde{\lambda} + \alpha_2^2 \tilde{\mu}}{\sqrt{\alpha_1^2 \tilde{\lambda}^2 + \alpha_2^2 \tilde{\mu}^2}}. \quad (2.18)$$

Минимизируя правую часть (2.18) при условии  $\alpha_1^2 + \alpha_2^2 = 1$ , получаем неравенство

$$\frac{(Ax_0, x_0)}{\|Ax_0\| \cdot \|x_0\|} \geq \frac{2\tilde{\lambda}\tilde{\mu}}{\tilde{\lambda} + \tilde{\mu}},$$

которое превращается в равенство при  $\alpha_1^2 = \tilde{\mu}/(\tilde{\lambda} + \tilde{\mu})$ ,  $\alpha_2^2 = \tilde{\lambda}/(\tilde{\lambda} + \tilde{\mu})$ . Отсюда сразу получаем

$$\min \frac{(Ax, x)}{\|Ax\| \cdot \|x\|} = \frac{2\sqrt{\tilde{\lambda}\tilde{\mu}}}{\tilde{\lambda} + \tilde{\mu}}$$

при

$$x = \sqrt{\frac{\tilde{\mu}}{\tilde{\lambda} + \tilde{\mu}}} s_1 + \sqrt{\frac{\tilde{\lambda}}{\tilde{\lambda} + \tilde{\mu}}} s_2,$$

где  $s_1, s_2$  — нормированные собственные векторы оператора  $A$ , соответствующие минимальному и максимальному собственным числам.

Обратимся теперь к теореме 2.7. Для нашего примера можем принять  $\cos \varphi = 2\sqrt{\tilde{\lambda}\tilde{\mu}}/(\tilde{\lambda} + \tilde{\mu})$ , при этом  $\sin \varphi = (\mu - \lambda)/(\mu + \lambda)$ , и обобщенный градиентный спуск, описанный в условии этой теоремы, дает процесс, сходящийся к точке минимума: по норме — со скоростью геометрической прогрессии со знаменателем (при  $\varphi \geq \pi/4$ )

$$q = (\mu - \lambda)/(\mu + \lambda) = (\rho - 1)/(\rho + 1),$$

где  $\rho = \mu/\lambda$  — число обусловленности, а по функционалу — со знаменателем  $q^2$ .

**З а м е ч а н и е.** Наши рассуждения почти полностью сохраняют силу, если  $A$  — неотрицательно определенный оператор,  $A \neq 0$ , только под  $\lambda$  нужно понимать наименьшее положительное собственное число.

Сопоставим полученный выше результат с результатом Л. В. Канторовича [38] о скорости сходимости метода наискорейшего спуска для

квадратичных функционалов. Этот метод сходится по функционалу со скоростью геометрической прогрессии со знаменателем  $q^2$ . Однако схема метода наискорейшего спуска для квадратичных функционалов (или для решения систем линейных уравнений с симметричной положительно определенной матрицей) гораздо сложнее, чем схема, описанная в условиях теоремы, и требует  $3l$  рабочих ячеек, т. е. в три раза больше, чем последняя схема, что имеет значение для задач большого объема (см. [76]).

Таким образом, если мы имеем возможность получить достаточно точную оценку сверху  $\hat{\rho}$  для числа обусловленности  $\rho$  и примем  $\sin \varphi = (\hat{\rho} - 1)/(\hat{\rho} + 1)$ , то алгоритм, описанный в теореме 2.7, может оказаться практически вполне конкурентоспособным с методом наискорейшего спуска.

Заметим, что для квадратичных функционалов  $f(x) = (Ax, x)$  величина  $\sigma$ , фигурирующая в теореме 2.8, может быть взята равной  $\sqrt{\rho}$ , при этом получаем сходимость со скоростью геометрической прогрессии со знаменателем

$$\bar{q} = \sqrt{\sigma^2 - 1}/\sigma = \sqrt{(\rho - 1)/\rho}.$$

Рассмотрим еще один вариант метода обобщенного градиентного спуска, когда шаговый множитель остается в течение определенного числа шагов постоянным, а затем уменьшается в два раза [97].

**Теорема 2.9.** Пусть для выпуклой функции  $f(x)$  выполняются условия теоремы 2.2,  $\sigma \geq 2$ . Рассмотрим при заданном  $x$  следующий итеративный процесс:

$$x_{k+1} = x_k - h_{k+1} \frac{g_f(x_k)}{\|g_f(x_k)\|},$$

где  $h_{k+1} = h_0 \cdot 2^{-l(k+1)/N}$ . При достаточно большом  $h_0$  и  $N \geq 3\sigma^2 + 1$  выполняется неравенство

$$\|x_k - x^*\| \leq 2\sigma h_{k+1}, \quad k = 0, 1, 2, \dots \quad (2.19)$$

**Доказательство.** Пусть  $h_0 \geq \|x_0 - x^*\|/2\sigma$ . Так же, как при доказательстве теоремы 2.2, получаем

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 + h_{k+1}^2 - 2h_{k+1}b_k(x^*).$$

Если  $b_k(x^*) \geq h_{k+1}/2$ , то

$$\|x_{k+1} - x^*\| \leq \|x_k - x^*\|. \quad (2.20)$$

Если  $b_k(x^*) \geq h_{k+1}$ , то

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - h_{k+1}^2. \quad (2.21)$$

Так как  $\|x_0 - x^*\|^2 \leq 4\sigma^2 h_0^2$  и  $h_i = h_0$  для  $i = 1, \dots, [3\sigma^2] + 1$ , то, если на всех шагах будет выполняться (2.21), получим

$$\|x_N - x^*\|^2 \leq 4\sigma^2 h_0^2 - ([3\sigma^2] + 1)h_0^2 \leq \sigma^2 h_0^2.$$

Если на некотором шаге  $\bar{k} < N$  условие (2.21) не будет выполняться, то  $b_{\bar{k}}(x^*) \leq h_0$  и  $\|x_{\bar{k}} - x^*\| \leq \sigma h_0$ . При этом могут возникнуть две ситуации:

а) на шагах  $k = \bar{k} + 1, \dots, N$  будет выполняться условие  $b_k(x^*) \geq h_{k+1}/2$ ; в этом случае в силу (2.20)  $\|x_N - x^*\| \leq \sigma h_0$ ;

б) на некотором шаге  $\bar{k}$  ( $\bar{k} \leq \bar{k} \leq N$ ) выполняется условие  $b_{\bar{k}}(x^*) \leq h_0/2$ , тогда  $\|x_{\bar{k}} - x^*\| \leq h_0\sigma/2$  и  $\|x_k - x^*\| \leq h_0\sigma/2 + h_0 \leq h_0\sigma$  при  $\sigma \geq 2$  для  $\bar{k} \leq k \leq N$ . Таким образом, во всех случаях

$$\|x_N - x^*\| \leq \sigma h_0, \quad \|x_i - x^*\| \leq 2\sigma h_0, \quad i = 0, \dots, N.$$

Принимая  $x_0^{(1)} = x$  и  $h_0^{(1)} = h_0/2$ , получим неравенство  $h_0^{(1)} \geq \|x_0^{(1)} - x^*\|/(2\sigma)$ , аналогичное (2.19), откуда следует

$$\|x_i - x^*\| \leq \sigma h_0, \quad i = N + 1, \dots, 2N - 1, \quad \|x_{2N} - x^*\| \leq \sigma h_0/2,$$

и далее, по индукции,

$$\|x_{rN+i} - x^*\| \leq 2\sigma h_0 2^{-r}, \quad i = 0, \dots, N - 1, \quad r = 1, 2, \dots,$$

что и требовалось доказать.

Рассмотрим вопрос о выборе параметров алгоритма варианта метода ОГС, содержащегося в условиях теоремы 2.7. Для простоты будем считать, что минимум  $f(x)$  достигается в единственной точке  $x^*$ . Для реализации соответствующего алгоритма необходимо задание двух параметров:  $h$  и  $\varphi$ . Если параметр  $\varphi$  выбран так, что справедливо неравенство (2.8) для всех  $x = x_k$  ( $k = 0, 1, \dots$ ), а  $h$  удовлетворяет неравенству (2.9), то теорема 2.7 гарантирует определенную точность решения после  $k$  шагов:

$$\Delta_k = \|x_k - x^*\| \leq h \sin^k \varphi / \cos \varphi. \quad (2.22)$$

Так как условие (2.8) в большинстве случаев эффективно не проверяется, то нужно иметь некоторый косвенный легко проверяемый критерий «правильности» выбора  $h$  и  $\varphi$ .

Возьмем новую начальную точку  $\bar{x}_0$  и  $\bar{\varphi}, \bar{\varphi} \leq \bar{\varphi} < \pi/2$ . Выберем  $\bar{h} = h + \|x_0 - \bar{x}_0\| \cos \bar{\varphi}$ . Легко видеть, что если параметры  $\bar{\varphi}, \bar{h}$  удовлетворяют (2.8), (2.9) для точки  $x_0$ , то  $\bar{\varphi}, \bar{h}$  удовлетворяют этим же условиям. Пусть  $\{\bar{x}_k\}, k = 0, 1, \dots$  — последовательность, получаемая при применении ОГС с начальной точкой  $\bar{x}_0$  и параметрами  $\bar{\varphi}$  и  $\bar{h}$ . Тогда в силу (2.22) должно выполняться неравенство

$$\|x_k - \bar{x}_k\| \leq \frac{h \sin^k \varphi}{\cos \varphi} + \frac{\bar{h} \sin^k \bar{\varphi}}{\cos \bar{\varphi}}, \quad k = 0, 1, \dots \quad (2.23)$$

Если для некоторого  $k$  неравенство не выполняется, это свидетельствует о неправильности выбора параметров  $h$  и  $\varphi$ . С другой стороны, если хотя бы одна из последовательностей  $\{x_k\}$  не сходится к  $x^*$ , то

маловероятно при  $\bar{x}_0 \neq x_0$  и  $\bar{\varphi} \neq \varphi$ , чтобы они сходились к одной и той же точке; если же эти последовательности сходятся к разным точкам, то при достаточно большом  $k$  неравенство (2.23) не будет выполняться.

Таким образом, неравенство (2.23) можно считать удобным критерием для оценки правильности выбора параметров  $h$  и  $\varphi$  при параллельном проведении двух процессов ОГС из разных начальных точек. Практически этот критерий можно использовать следующим образом. Допустим, что нам необходимо получить  $x_{\bar{k}}$  такое, что

$$\|x_{\bar{k}} - x^*\| < \delta. \quad (2.24)$$

Из неравенства (2.22) получаем, что если  $h$  и  $\varphi$  выбраны верно, то для гарантии выполнения (2.24) достаточно, чтобы выполнялось неравенство

$$\bar{k} \geq \left[ \frac{\ln \frac{\delta \cos \varphi}{h}}{\ln \sin \varphi} \right]. \quad (2.25)$$

Проведем процесс из двух начальных точек  $x_0$  и  $\bar{x}_0$ , как указано выше. Если мы проделали  $\bar{k}$  шагов, где  $\bar{k}$  удовлетворяет неравенству (2.25), и при этом для всех  $k \leq \bar{k}$  выполняется неравенство (2.23), то можно утверждать, что  $\|x_{\bar{k}} - x^*\| \leq \delta$ . Для более полной уверенности можно произвести спуск из еще одной начальной точки или более тщательно исследовать окрестность точки  $x_{\bar{k}}$ . Если же для некоторого  $k \leq \bar{k}$  неравенство (2.23) не будет выполнено, то следует изменить параметры  $h$  и  $\varphi$  (например, изменить  $\varphi$  таким образом, чтобы  $\cos \varphi$  уменьшился в два раза) и продолжить счет из наилучшей предыдущей и некоторой контрольной точек.

С методами ОГС тесно связаны релаксационные методы решения систем линейных и выпуклых неравенств и развитые И. И. Ереминым и его учениками методы фейеровских приближений, т. е. итеративные процессы, в которых процесс приближения к искомому множеству идет монотонно по расстоянию [31, 32, 101]. Подробно взаимоотношение релаксационных и субградиентных процессов описано в монографии [102], в работах [136, 138]. Релаксационные процессы минимизации просто реализуются, когда известно значение минимизируемой функции в точке минимума. За рубежом вычислительные эксперименты с субградиентными методами фактически начались с использования релаксационных процессов для получения оценок в задачах дискретного программирования [130, 142, 143]. Успешное решение ряда сложных задач с использованием субградиентных процессов релаксационного типа привело к лавинообразному росту числа публикаций по субградиентным и  $\epsilon$ -субградиентным методам (см. [104, 147]).

В работе [34] введено понятие обобщенного стохастического градиента (ОСГ) и обоснован метод случайного поиска, являющийся стохастическим аналогом ОГС. В дальнейшем этот метод получил развитие в работах Ю. М. Ермольева, Е. А. Нурминского, А. И. Гупала и др. [35, 53, 57]. Этот метод является эффективным средством

решения разнообразных задач стохастического программирования, в том числе транспортных и распределительных стохастических задач. Он принадлежит к разновидности методов случайного поиска, т. е. к таким итеративным процедурам, у которых направление движения на данном шаге определяется в результате реализации некоторого случайного (псевдослучайного) события в отличие от жестко определенных процедур в обычных градиентных методах.

При минимизации выпуклой функции  $f(x)$ , определенной на  $E^n$ , метод ОСГ задается формулой

$$x_{k+1} = x_k - h_k(x_k) g_\omega(x_k), \quad k = 0, 1, \dots, \quad (2.26)$$

где  $h_k(x_k)$  — шаговый множитель на  $k$ -м шаге,  $g_\omega(x_k)$  — случайный вектор, математическое ожидание которого совпадает с обобщенным градиентом функции  $f(x)$  в точке  $x_k$  (стохастический субградиент в точке  $x_k$ ). Предположим для простоты, что вероятностные характеристики вектора  $g_\omega(x_k)$  определяются точкой  $x_k$  и не зависят от предыстории процесса поиска, хотя обоснование сходимости проходит и для более общего случая. Пусть  $x^*$  — единственная точка минимума функции  $f(x)$ . Справедлива следующая теорема [34].

**Теорема 2.10.** Пусть выполняются условия:

$$1) \sum_{k=1}^{\infty} h_k(x_k) = +\infty, \quad h_k(x_k) > 0;$$

$$2) \sum_{k=1}^{\infty} h_k^2(x_k) < +\infty;$$

$$3) M_\omega \|g_\omega(x_k)\|^2 \leq c; \quad c > 0; \quad k = 0, 1, \dots \quad (M_\omega — символ математического ожидания).$$

Тогда с вероятностью 1

$$\lim_{k \rightarrow \infty} \|x_k - x^*\| = 0.$$

Метод ОСГ находит большое практическое применение, особенно при решении многоэтапных задач стохастического программирования. В следующей главе будет описано его применение к решению двухэтапной стохастической транспортной задачи, связанной с определением объемов складов однородной продукции при случайном спросе.

### § 3. Методы обобщенного градиентного спуска с растяжением пространства

При анализе алгоритмов ОСГ, сходящихся со скоростью геометрической прогрессии, выяснилось, что показатель скорости сходимости определяется величиной  $\sin \bar{\varphi}$ , где  $\bar{\varphi}$  — известная априори верхняя граница углов  $\varphi(x)$  между направлением из точки  $x$  на точку минимума и направлением антисубградиента —  $g_f(x)$  в данной точке

(считаем, что  $\bar{\varphi} \geq \pi/4$ ). Можно попытаться улучшить этот показатель, применив линейное неортогональное преобразование пространства переменных (при ортогональном преобразовании углы между соответствующими лучами сохраняют свою величину). Так, если минимизируемая функция является квадратичной положительно определенной (или хорошо аппроксимируется такой функцией), то ее поверхности уровня имеют форму концентрических эллипсоидов и можно подобрать такое линейное преобразование, при котором поверхности уровня становятся сферами, т. е. у соответствующих функций  $\bar{\varphi} = 0$  (близка к 0). Это фактически используется в квазиньютоновских методах минимизации достаточно гладких выпуклых функций, хорошо аппроксимируемых в окрестности минимума положительно определенными квадратичными функциями.

Рассмотрим другой пример, когда области, ограниченные поверхностями уровня, имеют форму подобных  $n$ -мерных параллелепипедов с центром в точке минимума. За счет подходящего линейного преобразования эти области преобразуются в гиперкубы, которым соответствует

$$\bar{\varphi} = \arccos^s \frac{1}{\sqrt{n}}, \quad \sin \bar{\varphi} = \sqrt{1 - \frac{1}{n}} = 1 - \frac{1}{2n} + o\left(\frac{1}{n}\right).$$

Этот пример соответствует задаче минимизации функции  $f(x) = \max_{1 \leq i \leq n} |(c_i, x) + b_i|$ , которая эквивалентна решению системы линейных уравнений  $(c_i, x) + b_i = 0$  ( $i = 1, \dots, n$ ), где  $c_1, \dots, c_n$  —  $n$ -мерные векторы.

В общем случае добиться эффекта ускорения сходимости за счет фиксированного линейного преобразования не удастся. Это связано с тем, что, во-первых, априорная информация о функции, как правило, слишком скудна для того, чтобы выбрать соответствующее преобразование, если оно существует; во-вторых, может оказаться, что  $\bar{\varphi} = \pi/2$  (это характерно, например, для нелинейных функций максимума), и никакое линейное невырожденное преобразование не изменит указанной верхней границы углов  $\varphi(x)$ .

Но если фиксированное линейное преобразование не дает глобального эффекта, то можно попытаться целенаправленно изменять метрику пространства после каждой итерации. При этом появляется возможность накапливать информацию о минимизируемой функции в процессе счета; кроме того, последовательность невырожденных преобразований пространства может стремиться к вырожденному преобразованию, что позволяет надеяться, что и случай  $\bar{\varphi} = \pi/2$  не станет камнем преткновения.

Как мы указывали выше, основным фактором, ограничивающим скорость сходимости ОГС в условиях программной регуляции шага, является близость к нулю косинуса углов между направлением антисубградиентов и направлением на точку минимума. Если бы мы могли значительно уменьшить составляющую антиградиента, ортогональную направлению на точку минимума, оставляя почти без



изменения составляющую, параллельную этому направлению, то получаемый при этом вектор давал бы направление спуска более предпочтительное, чем направление антиградиента, и можно было бы ожидать ускорения сходимости при соответствующей регулировке шага.

Заметим, что именно направление субградиента является в неблагоприятном случае почти ортогональным к направлению на точку минимума, и если это направление после данного шага существенно не изменилось, то мы должны стараться на последующих шагах уменьшить составляющую субградиентов, параллельную данному субградиенту, а этого можно добиться, проведя операцию растяжения пространства в направлении этого субградиента. Так, применяя весьма нестрогие рассуждения, приходим к идее обобщенного градиентного спуска с растяжением пространства в направлении субградиента. Как мы увидим ниже, определенные модификации этого метода сходятся со скоростью геометрической прогрессии.

Перейдем к математическому описанию метода. Для этого нужно вначале пояснить, что мы понимаем под оператором растяжения пространства в заданном направлении.

Пусть заданы вектор  $\xi \in E^n$ ,  $\|\xi\| = 1$ , и число  $\alpha \geq 0$ . Каждый вектор  $x \in E^n$  однозначно представим в виде

$$x = \gamma_\xi(x) \xi + d_\xi(x) \quad (2.27)$$

при условии

$$(\xi, d_\xi(x)) = 0. \quad (2.28)$$

Из соотношений (2.27), (2.28) получаем  $\gamma_\xi(x) = (x, \xi)$ ;  $d_\xi(x) = x - (x, \xi)\xi$ .

Оператором растяжения пространства  $E^n$  в направлении  $\xi$  с коэффициентом  $\alpha$  назовем оператор  $R_\alpha(\xi)$ , действующий следующим образом на вектор  $x$ , представленный в форме (2.27):

$$R_\alpha(\xi)x = \alpha\gamma_\xi(x)\xi + d_\xi(x).$$

Из этого определения вытекают следующие свойства оператора  $R_\alpha(\xi)$ .

1.  $R_\alpha(\xi)x = \alpha(x, \xi)\xi + [x - (x, \xi)\xi] = (\alpha - 1)(x, \xi)\xi + x. \quad (2.29)$

2. Оператор  $R_\alpha(\xi)$  — линейный симметричный:

$$(R_\alpha(\xi)x, y) = (\alpha - 1)(x, \xi)(y, \xi) + (x, y) = (x, R_\alpha(\xi)y).$$

3.  $R_{\alpha\beta}(\xi) = R_\alpha(\xi)R_\beta(\xi).$

4. При  $\alpha > 0$   $R_\alpha(\xi)R_{1/\alpha}(\xi) = R_1(\xi) = I$  ( $I$  — единичный оператор).

5. Оператор  $R_0(\xi)$  является оператором проектирования на подпространство, ортогональное вектору  $\xi$ :

$$R_0(\xi)x = d_\xi(x).$$

6. Оператор  $R_\alpha(\xi)$  при  $n \geq 2$  имеет два собственных числа  $\lambda_1 = \alpha$ ,  $\lambda_2 = 1$ ; первому из них соответствует подпространство соб-

ственных векторов, порожденное вектором  $\xi$ , второму — подпространство собственных векторов, состоящее из векторов, ортогональных  $\xi$ .

7. Пусть координаты вектора  $\xi$  в некоторой ортонормированной системе координат  $s = \{e_1, e_2, \dots, e_n\}$  равны  $\xi_1, \xi_2, \dots, \xi_n$ , тогда в силу (2.29) в этой системе координат преобразованию  $R_n(\xi)$  соответствует матрица  $R_\alpha(\xi)$  с элементами  $\{r_{ij}\}$ , вычисленными по формулам

$$r_{ij} = (R_\alpha(\xi) e_i, e_j) = \begin{cases} (\alpha - 1) \xi_i \xi_j & \text{для } i \neq j, \\ (\alpha - 1) \xi_i^2 + 1 & \text{для } i = j. \end{cases}$$

8. В силу формулы (2.29) вычисление вектора  $R_\alpha(\xi)x$  требует  $(2n + 1)$  операций умножения, вычисление матриц вида  $R_\alpha(\xi)A$  или  $AR_\alpha(\xi)$  при заданных  $A$ ,  $\xi$  и  $\alpha$  требует  $n(2n + 1)$  операций умножения.

9. Пусть  $x$  — произвольный ненулевой вектор из  $E^n$ ; тогда

$$\|R_\alpha(\xi)x\| = \sqrt{\|x\|^2 + (\alpha^2 - 1)(x, \xi)^2}. \quad (2.30)$$

Этот результат получается непосредственным применением формулы (2.29); в самом деле,

$$\begin{aligned} \|R_\alpha(\xi)x\| &= (R_\alpha(\xi)x, R_\alpha(\xi)x) = \\ &= (x + (\alpha - 1)(x, \xi)\xi, x + (\alpha - 1)(x, \xi)\xi) = \\ &= \|x\|^2 + 2(\alpha - 1)(x, \xi)^2 + (\alpha - 1)^2(x, \xi)^2 = \|x\|^2 + (\alpha^2 - 1)(x, \xi)^2, \end{aligned}$$

откуда и следует равенство (2.30).

10.  $R_\alpha(\xi)$  в матричной форме представим в виде

$$R_\alpha(\xi) = I + (\alpha - 1)\xi\xi^T; \quad (2.31)$$

действительно,

$$(I + (\alpha - 1)\xi\xi^T)x = x + (\alpha - 1)\xi(\xi, x),$$

что соответствует определяющему соотношению (2.29).

Рассмотрим класс алгоритмов минимизации выпуклых функций, на каждом шаге которых движение в направлении обобщенного градиента будет сочетаться с операцией растяжения пространства аргументов в этом же направлении. Алгоритмы этого класса будем называть алгоритмами обобщенного градиентного спуска с растяжением пространства в направлении градиента (*ОГСРП-алгоритмами*).

Предположим, что имеются: алгоритм, позволяющий точно вычислить обобщенный градиент  $g_f(x)$  минимизируемой выпуклой функции  $f(x)$  в произвольной точке  $x \in E^n$ , а также алгоритмы вычисления последовательностей положительных чисел  $\{h_k\}$  и  $\{\alpha_k\}$  ( $k = 1, 2, \dots$ ) (шаговых множителей и коэффициентов растяжения пространства), задано начальное приближение  $x_0$  и начальная неособая матрица  $B_0 = A_0^{-1}$  (в частности,  $B_0 = I$ ). При этих условиях определим бесконечношаговый процесс,  $(k + 1)$ -й шаг которого ( $k = 0, 1, 2, \dots$ ) описывается следующим образом. Вычисляем:

1)  $g_f(x_k)$  (если  $g_f(x_k) = 0$ , вычисления прекращаются, так как  $x_k$  дает точку минимума);

$$2) \quad g_{\varphi_k}(y_k) = B_k^* g_f(x_k) = \tilde{g}_k, \quad (2.32)$$

где  $\varphi_k(y) = f(B_k y)$ ,  $y_k = A_k x_k$ ,  $A_k = B_k^{-1}$ ,  $B_k^*$  — оператор, сопряженный оператору  $B_k$ .

Формула (2.32) дает возможность вычислять обобщенный градиент от функции  $\varphi_k(y) = f(A_k^{-1}y)$ , которая получается из  $f(x)$  при применении линейного преобразования пространства  $y = A_k x$ . Эта формула является частным случаем формулы для опорного функционала к выпуклому функционалу при линейном преобразовании банахового пространства [61]. Она может быть также элементарно выведена из определения обобщенного градиента. В тех точках, где  $g_f(x)$  определен неоднозначно, формулу (2.32) следует понимать как однозначное отображение множества  $G_f(x)$  на множество  $G_{\varphi_k}(y)$  (если оператор  $B_k$  — неособый, то отображение взаимно однозначное);

$$3) \quad \xi_{k+1} = \frac{g_{\varphi_k}(y_k)}{\|g_{\varphi_k}(y_k)\|} = \frac{\tilde{g}_k}{\|\tilde{g}_k\|}; \quad (2.33)$$

$$4) \quad h_{k+1};$$

$$5) \quad \alpha_{k+1};$$

$$6) \quad x_{k+1} = x_k - B_k h_{k+1} \xi_{k+1}; \quad (2.34)$$

формула (2.34) получается из формулы  $A_k x_{k+1} = y_k - h_{k+1} \xi_{k+1}$ , осуществляющей шаг обобщенного градиентного спуска для функции  $\varphi_k(y)$ , с последующим применением к обеим частям оператора  $B_k$  для отображения в основное пространство;

$$7) \quad B_{k+1} = A_{k+1}^{-1} = B_k R_{1/\alpha_{k+1}}(\xi_{k+1}); \quad (2.35)$$

формула (2.35) дает возможность вычислить оператор  $B_{k+1}$ , обратный результирующему оператору

$$A_{k+1} = R_{\alpha_{k+1}}(\xi_{k+1}) \dots R_{\alpha_1}(\xi_1) A_0$$

преобразования пространства, который получается при последовательном применении операторов растяжения пространства в направлении нормированных обобщенных градиентов  $\xi_1, \xi_2, \dots, \xi_{k+1}$  с коэффициентами  $\alpha_1, \dots, \alpha_{k+1}$ :

$$B_{k+1} = A_{k+1}^{-1} = A_k^{-1} R_{\alpha_{k+1}}^{-1}(\xi_{k+1}) = B_k R_{1/\alpha_{k+1}}(\xi_{k+1});$$

8) переходим к  $(k+2)$ -му шагу.

Каждый шаг ОГСРП требует ряда дополнительных операций по сравнению с методом обобщенного градиентного спуска. Наиболее трудоемкими из них являются операции 2), 6) и 7), каждая из которых требует порядка  $cn^2$  арифметических операций, из них около  $4n^2$  операций умножения. Кроме того, требуется дополнительный массив из  $n^2$  ячеек памяти для запоминания матриц  $B_k$ .

Заметим, что если  $B_0 = I$  и  $\alpha_k = 1$  ( $k = 1, 2, \dots$ ), то метод ОГСРП становится эквивалентным обобщенному градиентному спуску. Метод ОГСРП является более гибким по сравнению с методом ОГС, так как его конкретные реализации зависят от двух последовательностей  $\{\alpha_k\}$  и  $\{h_k\}$ . Применяя различные алгоритмы построения этих последовательностей, в принципе можно на единой основе строить различные модификации алгоритма ОГСРП и выбирать из них наиболее эффективные для данного класса задач.

Алгоритмы типа ОГСРП первоначально были разработаны для минимизации выпуклых функций. В дальнейшем оказалось, что некоторые их модификации применимы для нахождения локальных минимумов почти дифференцируемых функций (при этом, естественно, роль обобщенного градиента в алгоритме играет почти-градиент). В дальнейшем при исследовании сходимости алгоритмов ОГСРП в каждом конкретном случае будем оговаривать, идет ли речь о классе почти дифференцируемых функций или о подклассе выпуклых функций. При этом под обобщенным градиентом читатель должен понимать соответственно почти-градиент либо субградиент.

Покажем, что определенные варианты алгоритмов ОГСРП сходятся по функционалу со скоростью геометрической прогрессии, причем знаменатель этой прогрессии зависит от таких характеристик минимизируемой функции, которые инвариантны по отношению к невырожденным линейным преобразованиям пространства. Отметим, что для класса выпуклых функций удается построить алгоритм типа ОГСРП, который сходится по функционалу со скоростью геометрической прогрессии со знаменателем, зависящим только от размерности области определения функции. Этот алгоритм описан в § 5 гл. 2.

Доказательства указанных результатов основаны на изучении поведения нормы градиента минимизируемой функции в растянутом пространстве.

*Теорема 2.11.* Пусть  $f(x)$  — почти дифференцируемая функция, при минимизации которой с помощью алгоритма ОГСРП (2.32) — (2.35) получается минимизирующая последовательность  $\{x_k\}_{k=0}^{\infty}$ . Если при этом для некоторых положительных  $d, \alpha^*, \delta$  выполняются условия

- 1)  $\|g_j(x_k)\| \leq d$ ;
- 2)  $1 + \delta \leq \alpha_k \leq \alpha^*$ ;  $k = 0, 1, 2, \dots$ ,

то существует такая подпоследовательность  $\{x_{k_p}\}_{p=0}^{\infty}$ ,  $k_p < k_{p+1}$  и  $c > 0$ , что

$$\|\tilde{g}_{k_p}\| < c \left( \prod_{j=1}^{k_p} \alpha_j \right)^{-1/n}, \quad p = 1, 2, \dots$$

*Доказательство.* Запишем матрицу  $A_k$  в виде произведения ортогональной матрицы  $O_k$  и симметричной положительно определенной матрицы  $S_k$  (полярное разложение [76]):

$$A_k = O_k S_k, \quad A_0 = I.$$

Так как  $A_k = R_{\alpha_k}(\xi_k) \dots R_{\alpha_1}(\xi_1)$ , а произведение собственных чисел

матрицы равно ее определителю, то произведение собственных чисел матрицы  $S_k$  равно

$$\det S_k = \det A_k = \prod_{j=1}^k \alpha_j.$$

Пусть  $\{\lambda_i^{(k)}\}_{i=1}^n$  — последовательность собственных чисел оператора  $S_k$ , записанных в порядке неубывания,  $\{e_i^{(k)}\}_{i=1}^n$  — соответствующая ортонормированная система собственных векторов; тогда

$$S_k e_i^{(k)} = \lambda_i^{(k)} e_i^{(k)}, \quad \prod_{i=1}^n \lambda_i^{(k)} = \prod_{j=1}^k \alpha_j, \quad \lambda_n^{(k)} \geq \left( \prod_{j=1}^k \alpha_j \right)^{1/n}. \quad (2.36)$$

Кроме того,  $\lambda_i^{(k)} \geq 1$ , так как  $\alpha_j > 1$ ,  $j = 1, \dots, k$ , и поэтому

$$\|S_k e_i^{(k)}\| = \|A_k e_i^{(k)}\| \geq 1.$$

Пусть  $O_k e_i^{(k)} = \bar{e}_i^{(k)}$ . Система векторов  $\{\bar{e}_i^{(k)}\}_{i=1}^n$  также является ортонормированной ввиду ортогональности  $O_k$ .

Запишем  $g_f(x_k)$  в следующем виде:

$$g_f(x_k) = \sum_{i=1}^n g_i^{(k)} e_i^{(k)}.$$

Так как  $\|g_f(x_k)\| \leq d$ , то

$$|g_i^{(k)}| < d, \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots \quad (2.37)$$

Доказательство теоремы проведем теперь от противного. Допустим, что теорема неверна. Тогда для любого сколь угодно большого  $c$  найдется  $k(c)$  такое, что при  $k > k(c)$

$$\|\tilde{g}_k\| = \|B_k^* g_f(x_k)\| = \left\| \sum_{i=1}^n \frac{g_i^{(k)}}{\lambda_i^{(k)}} \bar{e}_i^{(k)} \right\| \geq c \left( \prod_{j=1}^k \alpha_j \right)^{-1/n}. \quad (2.38)$$

Это значит, что для  $k > k(c)$  существует такое  $i_k^*$ ,  $1 \leq i_k^* \leq n$ , что

$$\frac{|g_{i_k^*}^{(k)}|}{\lambda_{i_k^*}^{(k)}} \geq \frac{c \left( \prod_{j=1}^k \alpha_j \right)^{-1/n}}{\sqrt[n]{n}},$$

откуда  $\lambda_{i_k^*}^{(k)} \leq \frac{\sqrt[n]{nd}}{c} \left( \prod_{j=1}^k \alpha_j \right)^{1/n}$ . Далее,

$$\begin{aligned} \lambda_n^{(k)} = \max_{1 \leq i \leq n} \lambda_i^{(k)} &\geq \left( \frac{\prod_{j=1}^k \alpha_j}{\lambda_{i_k^*}^{(k)}} \right)^{1/(n-1)} &> \left( \frac{c}{\sqrt[n]{nd}} \right)^{1/(n-1)} \left( \prod_{j=1}^k \alpha_j \right)^{1/n} = \\ &= \tilde{c} \left( \prod_{j=1}^k \alpha_j \right)^{1/n}, \quad \tilde{c} = \left( \frac{c}{\sqrt[n]{nd}} \right)^{1/(n-1)}. \end{aligned}$$

Пусть  $e_n^{(k+1)} = \sum_{i=1}^n d_i^{(k)} e_i^{(k)}$ ,  $\sum_{i=1}^n (d_i^{(k)})^2 = 1$ . Тогда

$$A_{k+1} e_n^{(k+1)} = \lambda_n^{(k+1)} \bar{e}_n^{(k+1)} = R_{\alpha_{k+1}}(\xi_{k+1}) \left( \sum_{i=1}^n \lambda_i^{(k)} d_i^{(k)} \bar{e}_i^{(k)} \right).$$

Введем  $r_k = \min \{r: \lambda_n^{(k)} / \lambda_r^{(k)} \leq \alpha_{k+1} c; 1 \leq r \leq n\}$ . Представим  $y = \sum_{i=1}^n \lambda_i^{(k)} d_i^{(k)} \bar{e}_i^{(k)}$  в виде  $y = y_1 + y_2$ , где

$$y_1 = \sum_{i=1}^{r_k-1} \lambda_i^{(k)} d_i^{(k)} \bar{e}_i^{(k)}, \quad y_2 = \sum_{i=r_k}^n \lambda_i^{(k)} d_i^{(k)} \bar{e}_i^{(k)}.$$

Оценим  $\|R_{\alpha_{k+1}}(\xi_{k+1}) y\|$ . Для этого заметим, что

$$\xi_{k+1} = \frac{\tilde{g}_k}{\|\tilde{g}_k\|} = \sum_{i=1}^n \mu_i^{(k)} \bar{e}_i^{(k)},$$

где  $\mu_i^{(k)} = \frac{g_i^{(k)}}{\lambda_i^{(k)} \|\tilde{g}_k\|}$ . Из (2.37), (2.38) получаем

$$|\mu_i^{(k)}| \leq \frac{d \left( \prod_{j=1}^k \alpha_j \right)^{1/n}}{\lambda_i^{(k)} c}.$$

Для  $i \geq r_k$  имеем

$$|\mu_i^{(k)}| \leq \frac{d \left( \prod_{j=1}^k \alpha_j \right)^{\frac{1}{n}} \alpha_{k+1}}{\lambda_n^{(k)}} \leq \frac{d \alpha^*}{c}.$$

Далее,

$$\|R_{\alpha_{k+1}}(\xi_{k+1}) y_1\| \leq \alpha_{k+1} \|y_1\| \leq \lambda_n^{(k)} \frac{1}{c} \left\| \sum_{i=1}^{r_k-1} d_i^{(k)} \bar{e}_i^{(k)} \right\| \leq \frac{\lambda_n^{(k)}}{c};$$

$$\begin{aligned} \|R_{\alpha_{k+1}}(\xi_{k+1}) y_2\| &\leq \|y_2\| + (\alpha_{k+1} - 1) |(y_2, \xi_{k+1})| \leq \\ &\leq \lambda_n^{(k)} \left[ 1 + (\alpha^* - 1) \frac{d \alpha^*}{c} \right]. \end{aligned}$$

Отсюда имеем

$$\lambda_n^{(k+1)} = \|R_{\alpha_{k+1}}(\xi_{k+1}) y\| \leq \lambda_n^{(k)} \left[ 1 + \frac{1}{c} + (\alpha^* - 1) \frac{d \alpha^*}{c} \right]. \quad (2.39)$$

Из (2.39) следует, что для любого  $\varepsilon_0 > 0$  всегда можно подобрать столь большое  $c$ , что  $\lambda_n^{(k+1)} \leq (1 + \varepsilon_0) \lambda_n^{(k)}$  ( $k > k(c)$ ). Но если выбрать  $\varepsilon_0 < \delta$ , то при достаточно большом  $k$   $\lambda_n^{(k)} < c(1 + \delta)^{k/n} \leq c \left( \prod_{j=1}^k \alpha_j \right)^{1/n}$ , что противоречит (2.38). Теорема доказана.

Ближкий результат можно получить для оценки скорости убывания последовательности  $v_k = \min_{1 \leq r \leq k} \|\tilde{g}_r\|$ . Этот результат в дальнейшем будет использоваться для получения оценок скорости сходимости «рекордов» по функционалу к минимуму при использовании ОГСРП.

**Теорема 2.12.** В предположениях теоремы 2.11 при условии  $\alpha_k = \alpha > 1$  справедлива оценка

$$v_k \leq \frac{d \sqrt{k(\alpha^2 - 1)}}{\sqrt{\alpha^{2k/n} - 1}}, \quad k = 1, 2, \dots$$

**Доказательство.** Мы воспользуемся теми же обозначениями, что и для теоремы 2.11; имеем

$$\tilde{\xi}_{k+1} = \frac{\tilde{g}_k}{\|\tilde{g}_k\|} = \sum_{i=1}^n \mu_i^{(k)} \bar{e}_i^{(k)}, \quad k = 1, 2, \dots, \quad (2.40)$$

$$\mu_i^{(k)} = \frac{g_i^{(k)}}{\lambda_i^{(k)} \|\tilde{g}_k\|}, \quad i = 1, 2, \dots, n, \quad \sum_{i=1}^n (g_i^{(k)})^2 \leq d^2.$$

Пусть для некоторого  $k$

$$v_k = \min_{1 \leq r \leq k} \|\tilde{g}_r\| > \frac{d \sqrt{k(\alpha^2 - 1)}}{\sqrt{\alpha^{2k/n} - 1}}.$$

Тогда для  $r \leq k$  из (2.40) получаем

$$|\mu_i^{(r)}| < \frac{g_i^{(r)} \sqrt{\alpha^{2k/n} - 1}}{\lambda_i^{(r)} d \sqrt{k(\alpha^2 - 1)}}, \quad 1 \leq i \leq n. \quad (2.41)$$

Используя это соотношение, оценим при  $1 \leq r \leq k - 1$  разность  $(\lambda_n^{(r+1)})^2 - (\lambda_n^{(r)})^2$  ( $\lambda_n^{(r)}$  — максимальное собственное число оператора  $S_r$ ). Рассмотрим произвольный вектор  $a$  ( $\|a\| = 1$ ):

$$a = \sum_{i=1}^n a_i e_i^{(r)}, \quad \sum_{i=1}^n a_i^2 = 1, \quad (2.42)$$

$$A_r a = \sum_{i=1}^n a_i \lambda_i^{(r)} \bar{e}_i^{(r)},$$

$$\lambda_n^{(r+1)} = \max_{\|a\|=1} \|A_{r+1} a\| = \max_{\|a\|=1} \|R_\alpha(\tilde{\xi}_{r+1}) A_r a\|. \quad (2.43)$$

Воспользуемся формулой (2.30):

$$\|R_\alpha(\xi)x\| = \sqrt{\|x\|^2 + (\alpha^2 - 1)(x, \xi)^2}.$$

Приняв  $x = A_r a$ , из (2.41) — (2.43) получаем

$$\begin{aligned} (\lambda_n^{(r+1)})^2 &= \max_{\|a\|=1} \left( \sum_{i=1}^n a_i^2 (\lambda_i^{(r)})^2 + (\alpha^2 - 1) \left( \sum_{i=1}^n a_i \lambda_i^{(r)} \mu_i^{(r)} \right)^2 \right) < \\ &< \max_{\|a\|=1} \left( (\lambda_n^{(r)})^2 + (\alpha^2 - 1) \left( \sum_{i=1}^n |a_i g_i^{(r)}| \right)^2 \frac{\alpha^{2k/n} - 1}{kd^2 (\alpha^2 - 1)} \right) \leq \\ &\leq (\lambda_n^{(r)})^2 + \frac{(\alpha^{2k/n} - 1)}{k}, \quad r = 0, 1, \dots, k, \\ \lambda_n^{(0)} &= 1. \end{aligned}$$

Отсюда  $(\lambda_n^{(k)})^2 < \alpha^{2k/n}$ , что противоречит тому, что  $\lambda_n^{(k)} \geq \alpha^{k/n}$ . Теорема доказана.

Для того чтобы перейти к оценке скорости сходимости метода ОГСРП по функционалу, нужно получить оценки расстояния до области минимумов в «растянутом» пространстве. Для некоторых вариантов ОГСРП эти оценки удается получить.

**Теорема 2.13.** Пусть  $f(x)$  — почти дифференцируемая функция, определенная в некоторой сферической окрестности  $S_d$  точки  $x^*$ , являющейся локальным минимумом  $f(x)$ ,  $S_d = \{x: \|x - x^*\| \leq d\}$ , и в области  $S_d$  почти-градиент удовлетворяет двустороннему неравенству

$$N(f(x) - f(x^*)) \leq (g_f(x), x - x^*) \leq M(f(x) - f(x^*)), \quad (2.44)$$

где  $M > N$  — положительные константы. Тогда если в алгоритме ОГСРП принять:

$$1) \quad x_0 \in S_d,$$

$$2) \quad h_{k+1} = \frac{2MN}{M+N} \frac{f(x_k) - f(x^*)}{\| \tilde{g}_k \|}, \quad (2.45)$$

$$3) \quad 1 < \alpha_{k+1} \leq \frac{(M+N)}{(M-N)}, \quad k = 0, 1, 2, \dots, \quad (2.46)$$

то для всех  $k = 0, 1, \dots$   $\|A_k(x_k - x^*)\| \leq d$ .

**Доказательство.** Этот результат доказывается методом математической индукции. Пусть для некоторого  $p$

$$\|A_p(x_p - x^*)\| \leq d.$$

Оценим

$$\|A_{p+1}(x_{p+1} - x^*)\|.$$



Обозначим  $A_k(x_k - x^*) = z_k$ . Тогда

$$\begin{aligned} \|z_{p+1}\|^2 &= \|R_{\alpha_{p+1}}(\xi_{p+1})(z_p - h_{p+1}\xi_{p+1})\|^2 = \\ &= \|R_{\alpha_{p+1}}(\xi_{p+1})z_p\|^2 - 2h_{p+1}(R_{\alpha_{p+1}}(\xi_{p+1})z_p, \\ &R_{\alpha_{p+1}}(\xi_{p+1})\xi_{p+1}) + \alpha_{p+1}^2 h_{p+1}^2 = \left(1 + (\alpha_{p+1}^2 - 1)\frac{(z_p, \xi_{p+1})^2}{\|z_p\|^2}\right) \|z_p\|^2 - \\ &- 2h_{p+1}\alpha_{p+1}^2(z_p, \xi_{p+1}) + \alpha_{p+1}^2 h_{p+1}^2 = \\ &= \|z_p\|^2 + \alpha_{p+1}^2((z_p, \xi_{p+1}) - h_{p+1})^2 - (z_p, \xi_{p+1})^2. \quad (2.47) \end{aligned}$$

Заметим, что

$$\begin{aligned} (z_p, \xi_{p+1}) &= (A_p(x_p - x^*), \xi_{p+1}) = \\ &= (x_p - x^*, A_p^* \xi_{p+1}) = (g_f(x_p), x_p - x^*) / \|\tilde{g}_p\|. \end{aligned}$$

Рассмотрим два случая: а)  $(z_p, \xi_{p+1}) \leq h_{p+1}$ ; тогда из (2.44), (2.45)

$$h_{p+1} = \frac{(f(x_p) - f(x^*)) 2MN}{\|\tilde{g}_p\| (M+N)} \leq \frac{(g_f(x_p), x_p - x^*) 2M}{\|\tilde{g}_p\| (M+N)} = (z_p, \xi_{p+1}) \frac{2M}{M+N},$$

откуда

$$\begin{aligned} \alpha_{p+1}^2((z_p, \xi_{p+1}) - h_{p+1})^2 - (z_p, \xi_{p+1})^2 &\leq \\ &\leq \alpha_{p+1}^2 \left( (z_p, \xi_{p+1}) - (z_p, \xi_{p+1}) \frac{2M}{M+N} \right)^2 - (z_p, \xi_{p+1})^2 \leq \\ &\leq \left( \left( \frac{M+N}{M-N} \right)^2 \left( 1 - \frac{2M}{M+N} \right)^2 - 1 \right) (z_p, \xi_{p+1})^2 = 0; \end{aligned}$$

б)  $(z_p, \xi_{p+1}) \geq h_{p+1}$ ; тогда из (2.44), (2.45) получаем

$$\begin{aligned} h_{p+1} &= \frac{(f(x_p) - f(x^*)) 2MN}{\|\tilde{g}_p\| (M+N)} \geq \frac{(g_f(x_p), x_p - x^*) 2N}{M+N} = \\ &= (z_p, \xi_{p+1}) \frac{2N}{M+N}, \end{aligned}$$

откуда

$$\begin{aligned} \alpha_{p+1}^2((z_p, \xi_{p+1}) - h_{p+1})^2 - (z_p, \xi_{p+1})^2 &\leq \\ &\leq \alpha_{p+1}^2 \left( (z_p, \xi_{p+1}) - (z_p, \xi_{p+1}) \frac{2N}{M+N} \right)^2 - (z_p, \xi_{p+1})^2 \leq \\ &\leq \left( \left( \frac{M+N}{M-N} \right)^2 \left( 1 - \frac{2N}{M+N} \right)^2 - 1 \right) (z_p, \xi_{p+1})^2 = 0. \end{aligned}$$

Таким образом, в обоих случаях

$$\alpha_{p+1}^2((z_p, \xi_{p+1}) - h_{p+1})^2 - (z_p, \xi_{p+1})^2 \leq 0.$$

Отсюда, используя равенство (2.47), получаем  $\|z_{p+1}\| \leq \|z_p\|$ . Из

этого неравенства и условия 1) теоремы следует, что  $\|A_k(x_k - x^*)\| \leq d$  для любого  $k = 0, 1, \dots$ . Теорема доказана.

Как следствие из теорем 2.11, 2.12 и 2.13 получаем следующий результат.

**Теорема 2.14.** В условиях теоремы 2.13 при  $\alpha_k = \alpha$  ( $k = 0, 1, \dots$ ) найдутся такая подпоследовательность индексов  $k_1, k_2, \dots$  и такое положительное число  $c$ , что

$$f(x_{k_p}) - f(x^*) < c\alpha^{-k_p/n}.$$

Кроме того,

$$\min_{1 \leq i \leq k} (f(x_i) - f(x^*)) \leq \frac{G \sqrt{k(\alpha^2 - 1)} d}{\sqrt{\alpha^{2k/n} - 1}},$$

где  $G = \max_{x \in S_d} \|g_f(x)\|$ .

**Доказательство.** Так как процесс спуска проходит в ограниченном множестве  $S_d$ , то множество почти-градиентов ограничено, значит, выполняются условия теорем 2.11 и 2.12.

Пусть  $\{k_p\}_{p=1}^{\infty}$  — подпоследовательность, получаемая в соответствии с теоремой 2.11, причем  $\|\tilde{g}_{k_p}\| < c\alpha^{-k_p/n}$ . Используя (2.44), получаем

$$\begin{aligned} N(f(x_{k_p}) - f(x^*)) &\leq (g_f(x_{k_p}), x_{k_p} - x^*) = \\ &= (A_{k_p}^* g_{k_p}, x_{k_p} - x^*) = (\tilde{g}_{k_p}, A_{k_p}(x_{k_p} - x^*)) \leq \|\tilde{g}_{k_p}\| d < c d \alpha^{-k_p/n}. \end{aligned}$$

Приняв  $\bar{c}d/N = c$ , получим

$$f(x_{k_p}) - f(x^*) < \left(\frac{\bar{c}d}{N}\right) \alpha^{-k_p/n} = c\alpha^{-k_p/n}.$$

Аналогично, используя теорему 2.12, получаем

$$\min_{1 \leq r \leq k} (f(x_r) - f(x^*)) \leq \min_{1 \leq r \leq k} \|\tilde{g}_r\| \frac{d}{N},$$

откуда

$$\min_{1 \leq r \leq k} (f(x_r) - f(x^*)) \leq \frac{G \sqrt{k(\alpha^2 - 1)} d}{N \sqrt{\alpha^{2k/n} - 1}}.$$

Теорема доказана.

Теорема 2.13 обобщается на случай, когда точка минимума функции  $f(x)$  определена неоднозначно.

Пусть множество  $M^*$  точек минимума почти дифференцируемой функции  $f(x)$  ограничено и  $f(x)$  принимает на  $M^*$  значение  $f^*$ . Введем обозначение

$$\rho_k(x) = \min_{x^* \in M^*} \|A_k(x_k - x^*)\|.$$

Точку, в которой достигается минимум, обозначим через  $x_k^*(x)$ .

**Теорема 2.15.** Пусть для всех  $x, x^*$ , удовлетворяющих условию  $\rho_0(x) \leq d, x \notin M^*, x^* \in M^*$ , выполняются неравенства

$$N(f(x) - f^*) \leq (g_f(x), x - x^{**}) \leq M(f(x) - f^*),$$

где  $x^{**}$  — ближайшая к  $x$  точка минимума, лежащая на луче  $y = x + t(x - x^*) (t \geq 0)$ . Тогда, если при применении алгоритма ОГСРП

$$1) \rho_0(x_0) \leq d,$$

$$2) h_{k+1} = \frac{2MN}{M+N} \frac{f(x_k) - f^*}{\|g_{\Phi_k}(y_k)\|},$$

$$3) 1 < \alpha_{k+1} \leq \frac{M+N}{M-N},$$

то  $\rho_{k+1}(x_{k+1}) \leq d (k = 0, 1, 2, \dots)$ .

**Доказательство.** Покажем, что последовательность  $\{\rho_{k+1}(x_{k+1})\}_{k=0}^{\infty}$  является невозрастающей:

$$\begin{aligned} \rho_{k+1}^2(x_{k+1}) &= \min_{y \in M^*} \|R_{\alpha_{k+1}}(\xi_{k+1})A_k(x_{k+1} - y)\|^2 = \\ &= \min_{y \in M^*} \|R_{\alpha_{k+1}}(\xi_{k+1})A_k(x_k - y) - R_{\alpha_{k+1}}(\xi_{k+1})h_{k+1}\xi_{k+1}\|^2 \leq \\ &\leq \|R_{\alpha_{k+1}}(\xi_{k+1})A_k(x_k - x_k^*(x_k)) - \alpha_{k+1}h_{k+1}\xi_{k+1}\|^2. \end{aligned}$$

Введем обозначение  $z_k = A_k(x_k - x_k^*(x_k))$ . После этого доказательство аналогично доказательству теоремы 2.13. В итоге получаем

$$\|z_{k+1}\| \leq \|z_k\| \leq \|z_0\| \leq d \quad (k = 0, 1, 2, \dots).$$

**Теорема 2.16.** Пусть выпуклая функция  $f(x)$  обладает следующим свойством: существует число  $M > 1$  такое, что если

$$\varphi(\alpha) = f((1 - \alpha)x_1 + \alpha x_2), \quad 0 \leq \alpha \leq 1,$$

строго убывает по  $\alpha$ , то выполняется неравенство

$$(g_f(x_1), x_1 - x_2) \leq M(f(x_1) - f(x_2)); \quad (2.48)$$

кроме того,  $\lim_{\|x\| \rightarrow \infty} f(x) = +\infty$ . Тогда, если при применении алгоритма ОГСРП

$$\alpha_{k+1} = \frac{M+1}{M-1}, \quad h_{k+1} = \frac{2M}{M+1} \frac{f(x_k) - \bar{f}}{\|g_k\|}$$

и  $\bar{f} \geq f^*$ , то последовательность  $\{h_k\}$  является ограниченной и для произвольного  $\epsilon > 0$  найдется  $\bar{k}$  такое, что  $f(x_{\bar{k}}) \leq \bar{f} + \epsilon$  (если на некотором шаге  $f(x_k) < \bar{f}$ , то итерации прекращаются); если же  $\bar{f} < f^*$ , то последовательность  $\{h_k\}$  является неограниченной.

**Доказательство.** Если  $\bar{f} \geq f^*$ , то рассмотрим функцию

$$\bar{f}(x) = \begin{cases} f(x), & f(x) \geq \bar{f}, \\ \bar{f}, & f(x) < \bar{f}. \end{cases}$$

Для этой функции справедливы условия теоремы 2.15, следовательно, последовательность  $\{\rho_k(x_k)\}$  ограничена; значит, справедлива теорема 2.14 о сходимости по функционалу. Далее, из выпуклости  $\bar{f}(x)$  следует, что

$$\frac{(f(x_k) - \bar{f})}{\|\tilde{\rho}_k\|} \leq \rho_k(x_k).$$

Таким образом,  $h_k \leq \frac{2M}{M+1} \rho_k(x_k)$ , т. е.  $\{h_k\}$  ограничена.

Если  $\bar{f}$  выбрано меньшим  $f^*$ , то возможны два случая:

а)  $\sup_k f(x_k) < \infty$ ; в этом случае выполняются условия теоремы 2.11,

из которой следует существование подпоследовательности  $\{\|\tilde{g}_{k_p}\|\}_{p=1}^\infty$ , стремящейся к нулю, значит,

$$\sup_k h_k = \sup_k \frac{2M}{M+1} \frac{f(x_k) - f}{\|\tilde{g}_k\|} \geq \sup_k \frac{f^* - f}{\|\tilde{g}_k\|} = +\infty;$$

б)  $\sup_k f(x_k) = +\infty$ .

Рассмотрим луч, исходящий из точки  $x_k$  и соединяющий ее с некоторой точкой минимума  $f(x)$ . Пусть  $x_k^*$  — первая точка минимума на этом луче. Применим неравенство (2.48):

$$\frac{f(x_k) - f^*}{\|g_f(x_k)\|} \geq \frac{1}{M} \frac{(g_f(x_k), x_k - x^*)}{\|g_f(x_k)\|} = \frac{R_k}{M},$$

где  $R_k$  — расстояние от точки  $x_k^*$  до опорной гиперплоскости к множеству Лебега  $\{x : f(x) \leq f(x_k)\}$ , проведенной через точку  $x_k$ . Так как область минимумов  $M^*$  ограничена, а последовательность  $\{f(x_k)\}$  неограничена, то  $\sup_k R_k = +\infty$ . Далее,

$$\begin{aligned} \sup_k h_k &= \sup_k \frac{2M}{M+1} \frac{f(x_k) - f}{\|\tilde{g}_k\|} \geq \\ &\geq \sup_k \frac{2M}{M+1} \frac{f(x_k) - f^*}{\|g_f(x_k)\|} \geq \frac{2}{M+1} \sup_k R_k = +\infty. \end{aligned}$$

Теорема доказана.

Теорема 2.16 позволяет построить алгоритм минимизации функции  $f(x)$ , удовлетворяющей условию этой теоремы, при неизвестном  $f^*$ . Этот алгоритм будет состоять из последовательности этапов, на каждом из которых будет применяться ОГСРП в форме, указанной в теореме 2.16.

Вначале выбираем  $x_0^1 \in E^n$ ,  $h > 0$ ,  $\Delta_0 > 0$ ,  $\tilde{f}_0 = f(x_0^1)$ . Пусть выполнены  $r$  этапов алгоритма. Перед  $(r+1)$ -м этапом имеем

$$x_0^{r+1} \in E^n, \quad h > 0, \quad \Delta_r > 0, \quad \tilde{f}_r = f(x_0^{r+1}),$$

где  $x_0^{r+1}$  — точка, в которой получено рекордное значение функционала на  $r$  этапах,  $\bar{f}$  — значение этого рекорда. Возможны два события:

- а)  $f(x_0^{r+1}) \leq f(x_0^r) - \Delta_r/2$ , в этом случае  $\Delta_{r+1} = \Delta_r$ ;
- б)  $f(x_0^{r+1}) > f(x_0^r) - \Delta_r/2$ , в этом случае  $\Delta_{r+1} = \Delta_r/2$ .

В обоих случаях  $\bar{f}_{r+1} = f(x_0^{r+1}) - \Delta_{r+1}$ . Приняв  $x_0 = x_0^{r+1}$  и  $\bar{f} = \bar{f}_{r+1}$ , применяем алгоритм ОГСРП в форме, описанной в теореме 2.16. На некотором шаге этого алгоритма  $k = \bar{k}_{r+1}$  произойдет одно из двух событий:

- а)  $f(x_{\bar{k}_{r+1}}) \leq f(x_0^{r+1}) - \Delta_{r+1}/2$ ;
- б)  $h_{\bar{k}_{r+1}+1} > h$ .

На этом  $(r+1)$ -й этап прекращается, запоминаются  $\min_{0 \leq k \leq \bar{k}_{r+1}} f(x_k^{r+1})$

и соответствующая точка  $x_{\bar{k}_{r+1}}^{r+1} = x_0^{r+2}$ .

Перейдем к  $(r+2)$ -му этапу. Покажем, что  $\lim_{r \rightarrow \infty} f(x_0^r) = f^*$ . Так как последовательность  $\{f(x_0^r)\}$  является невозрастающей и ограниченной снизу, то  $\lim_{r \rightarrow \infty} f(x_0^r)$  существует. Из описания алгоритма и доказательства теоремы 2.16 ясно, что при достаточно малом  $\Delta_r$  случай б) может произойти только тогда, когда  $\bar{f}_r < f^*$ . С другой стороны, случай а) может произойти после случая б) только конечное число раз подряд. Таким образом,  $\lim_{r \rightarrow +\infty} \Delta_r = 0$ . Начиная с достаточно малых  $\Delta_r$  (при фиксированном  $h$ ), если произошел случай б), то  $f(x_0^r) - \Delta_r$  служит оценкой снизу для  $f^*$ ; поэтому

$$\lim_{r \rightarrow \infty} f(x_0^r) = f^*,$$

что и требовалось доказать.

При определенных условиях, например при решении систем нелинейных уравнений (см. [97]),  $f^*$  известно, но могут возникнуть трудности с оценкой  $M$  и  $N$ . При неправильном выборе констант  $M$  и  $N$  сходимость  $(M, N)$ -алгоритма ОГСРП, описанного в теореме 2.13, может оказаться слишком медленной или вообще отсутствовать. Поэтому важно иметь признаки, с помощью которых можно получить информацию о «ненормальном» ходе процесса минимизации. Наиболее простой из этих признаков основан на изучении поведения последовательности  $\{h_k\}$ .

**Теорема 2.17.** *Если параметры  $M, N$  в  $(M, N)$ -алгоритме выбраны правильно, то последовательность  $\{h_k\}$  ограничена. Если при неправильном выборе параметров  $M$  и  $N$  отсутствует сходимость, то одна из последовательностей  $\{h_k\}$  и  $\{f(x_k)\}$  не ограничена.*

Эта теорема доказывается по той же схеме, что и теорема 2.16. Таким образом, если в процессе минимизации по схеме ОГСРП, описанной в теореме 2.13, при заданных  $M$  и  $N$   $h_k$  либо  $f(x_k)$  превзошли

достаточно большие заданные числа  $h^{\max}$  и  $f^{\max}$ , то следует константу  $M$  увеличить,  $N$  уменьшить и начать процесс заново из начальной или наилучшей достигнутой точки.

#### § 4. Субградиентные методы, использующие операцию растяжения пространства в направлении разности двух последовательных субградиентов

Среди методов недифференцируемой оптимизации особое положение занимают по своей практической эффективности методы с растяжением пространства в направлении разности двух последовательных субградиентов (или так называемые  $r$ -алгоритмы). По своей структуре они очень близки к методам типа ОГСРП, но между ними есть важное различие: сходящиеся к минимуму методы типа ОГСРП в принципе не могут быть монотонными по значению минимизируемой функции, в то же время  $r$ -алгоритмы при определенной регулировке шаговых множителей могут быть сделаны монотонными. Это связано с простым геометрическим фактом: если мы находимся на границе двух «кусков» кусочно гладкой поверхности, причем градиенты к этим кускам, вычисленные в данной точке, образуют между собой тупой угол, то любое растяжение пространства в направлении одного из градиентов (или последовательное растяжение попеременно в направлении двух указанных градиентов) не может превратить этот угол в острый, он может лишь приближаться к  $\pi/2$ , оставаясь тупым. В то же время растяжение пространства в направлении разности двух указанных градиентов с достаточно большим коэффициентом растяжения превращает тупой угол между градиентами в острый.

Пусть рассматриваемая на рис. 2.1 кусочно гладкая поверхность  $U$  является поверхностью уровня некоторой выпуклой функции  $f(x)$ . До растяжения пространства направления антиградиентов  $-g_1$  и  $-g_2$  давали направление возрастания функции  $f(x)$  (вне области, ограниченной поверхностью уровня  $U$ ). После операции растяжения пространства в направлении  $g_2 - g_1$  антиградиенты  $-\bar{g}_1$  и  $-\bar{g}_2$  дадут направления внутрь области, ограниченной соответствующей поверхностью  $\bar{U}$  в преобразованном пространстве, т. е. направления  $-\bar{g}_1$  и  $-\bar{g}_2$  в преобразованном пространстве будут направлениями убывания функции.

Рассмотрим общую схему  $r$ -алгоритмов при минимизации почти дифференцируемой функции  $f(x)$ , определенной на  $E^n$ . Будем предполагать, что

$$\lim_{\|x\| \rightarrow \infty} f(x) = +\infty.$$

Выбираем начальное приближение  $x_0 \in E^n$  и неособую матрицу  $B_0$  (чаще всего  $B_0$  совпадает с единичной матрицей  $I_n$  или с диагональной матрицей  $D_n$  с положительными элементами на диагонали, с помощью которой осуществляется масштабирование переменных). Первый шаг алгоритма производим по формуле

$$x_1 = x_0 - h_0 \eta_0,$$

где  $\eta_0 = B_0 B_0^* g_f(x_0)$ ,  $h_0 \geq 0$  — некоторый шаговый множитель, выбираемый из условия существования в точке  $x_1$  почти-градиента  $g_f(x_1)$  такого, что

$$(g_f(x_1), \eta_0) \leq 0.$$

При  $B_0 = I_n$  имеем  $\eta_0 = g_f(x_0)$ , и первый шаг совпадает с итерацией субградиентного процесса.

Пусть в результате вычислений после  $k$  ( $k = 1, 2, \dots$ ) шагов процесса получены определенные значения  $x_k \in E^n$  и матрицы  $B_k$  размера

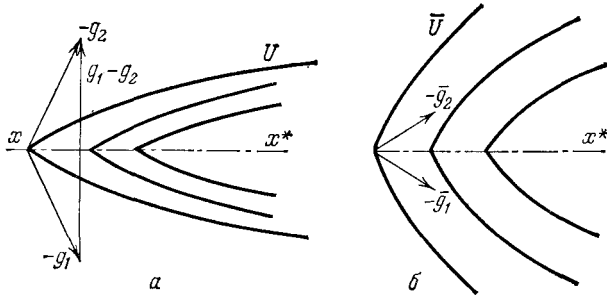


Рис. 2.1

$n \times n$ . Опишем  $(k + 1)$ -й шаг процесса. Вычисляем следующие величины:

- 1)  $g_f(x_k)$  — почти-градиент функции  $f(x)$  в точке  $x_k$ ;
- 2)  $\Delta_k = g_f(x_k) - g_f(x_{k-1})$ ;
- 3)  $r_k = B_k(g_f(x_k) - g_f(x_{k-1}))$  — значение разности двух последовательных градиентов в преобразованном пространстве.

Переход от начального пространства к преобразованному задается формулой  $y = A_k x$ , где  $A_k = B_k^{-1}$ . Определим функцию  $\varphi_k(y) = f(B_k y)$ ; тогда  $g_{\varphi_k}(y) = B_k^* g_f(x)$ . Таким образом,  $r_k$  есть разность двух почти-градиентов от функции  $\varphi_k(y)$ , вычисленных в точках  $y_k = A_k x_k$  и  $y_{k-1} = A_k x_{k-1}$ ;

4)  $\xi_k = r_k / \|r_k\|$ ; нормировка вектора  $r_k$  необходима для подготовки очередной операции растяжения пространства в направлении, определенном этим вектором;

5) задаем величину  $\beta_k$ , обратную коэффициенту растяжения  $\alpha_k$  пространства перед  $(k + 1)$ -м шагом;

6) вычисляем  $B_{k+1} = B_k R_{\beta_k}(\xi_k)$ ; заметим, что  $B_{k+1} = A_{k+1}^{-1}$ , где  $A_{k+1} = R_{\alpha_k}(\xi_k) A_k$  — оператор растяжения пространства перед  $(k + 1)$ -м шагом;

7)  $\tilde{g}_k = B_{k+1}^* g_f(x_k)$  — почти-градиент функции  $\varphi_{k+1}(y) = f(B_{k+1} y)$ , взятый в точке  $y_{k+1} = A_{k+1} x_k$ ;

$$8) \quad x_{k+1} = x_k - h_k B_{k+1} \tilde{g}_k / \|\tilde{g}_k\| \quad (2.49)$$

—«рабочий» шаг алгоритма, имитирующий шаг обобщенного градиентного спуска в растянутом под воздействием оператора  $A_{k+1}$  пространстве. Применяя к обеим частям формулы (2.49) оператор  $A_{k+1}$ , получим

$$y_{k+1} = A_{k+1}x_{k+1} = \bar{y} - h_k \tilde{g}_k / \|\tilde{g}_k\|, \quad (2.50)$$

где  $\bar{y}_k = A_{k+1}x_k$ ;

9) переход к следующему шагу или окончание работы алгоритма.

Выбор шагового множителя  $h_k$  — существенная часть алгоритма, от которого во многом зависит его практическая эффективность. В  $r$ -алгоритме выбор  $h_k$  осуществляется из условия приближенного поиска минимума по направлению, при этом при минимизации выпуклых функций должно соблюдаться условие  $h_k \geq h_k^*$  ( $h_k^*$  — значение шагового множителя, соответствующего минимуму по направлению), в общем же случае нужно следить за тем, чтобы направление обобщенного градиента в точке  $x_{k+1}$  образовывало нетупой угол с направлением спуска из точки  $x_k$ . При минимизации негладких выпуклых функций, определенных на  $E^n$ , наиболее удачными оказались при проведении экспериментальных и практических расчетов следующие варианты алгоритма.

Коэффициенты растяжения пространства  $\alpha_k$  выбираются в пределах 2—3, для шагового множителя  $h_k$  применяется адаптивный способ регулировки. задается некоторое натуральное число  $\bar{m}$ , константы  $q > 1$  и  $t_0^0 > 0$ . После  $k$  шагов получаем константу  $t_k^0$ . Двигаемся из точки  $x_k$  в направлении спуска с шагом, равным  $t_k^0$ , до тех пор, пока не будет выполнено условие завершения спуска по направлению, либо число шагов не станет равным  $\bar{m}$ . Условие завершения спуска может состоять в том, что значение функции в очередной точке не меньше, чем значение функции в предыдущей точке; другой вариант такого условия — производная по направлению спуска в данной точке неотрицательна. Если прошло  $\bar{m}$  шагов, а условие завершения спуска не выполнено, то вместо  $t_k^0$  запоминаем  $t_k^1 = qt_k^0$ , где  $q > 1$ , и продолжаем спуск в том же направлении с большим шагом. Если после очередных  $\bar{m}$  шагов условие завершения спуска не выполнено, то вместо  $t_k^1$  берем  $t_k^2 = qt_k^1$  и т. д. Так как мы предполагаем, что  $\lim_{\|x\| \rightarrow \infty} f(x) = +\infty$ ,

то после конечного числа шагов в определенном направлении обязательно выполнится условие завершения спуска. Константа шага  $t_k^{pk} = q^{pk} t_k^0$  ( $p \in \{0, 1, 2, \dots\}$ ), которая использовалась на последнем шаге, принимается в качестве начальной при спуске в новом направлении из точки  $x_{k+1}$ , полученной при завершении спуска, т. е.  $t_{k+1}^0 = t_k^{pk}$ .

Может возникнуть вопрос: шаговый множитель не убывает — как же получается сходящийся процесс? Сходимость  $r$ -алгоритма связана с тем, что, как правило, последовательность матриц  $B_k$  стремится к нулевой матрице при  $k \rightarrow \infty$ . Лишь в исключительных случаях, когда траектория спуска, начиная с определенного момента, лежит в некотором линейном многообразии размерности, меньшей  $n$ , либо



последовательность векторов  $\{x_{k+1} - x_k\}$  образует углы, стремящиеся к 0, с некоторым, подпространством,  $B_k$  может не стремиться к 0, но при этом  $B_k \bar{g}_k / \|\bar{g}_k\|$  стремится к 0.

Приведенный выше способ регулировки шагового множителя основан на следующих соображениях. Допустим, мы применим  $r$ -алгоритм с постоянным коэффициентом растяжения пространства  $\alpha$ . Тогда за  $n$  итераций произвольное направление растянется в «среднем» в  $\alpha$  раз, т. е. если мы будем идти в растянутом пространстве с постоянным шагом, то в первоначальном пространстве за  $n$  итераций шаг будет уменьшаться примерно в  $\alpha$  раз. Если расстояние до точки минимума при этом будет уменьшаться с такой же средней скоростью, т. е. примерно в  $\alpha$  раз за  $n$  итераций, то число шагов по направлению будет ограниченным. Если же темп сходимости будет более медленным, то шаг в первоначальном пространстве будет «в среднем» уменьшаться быстрее, чем расстояние до точки минимума, при этом число шагов по направлению может неограниченно возрастать. Для предотвращения этого и вводится адаптивная процедура увеличения шагового множителя  $h$ , которая стабилизирует число шагов по направлению в пределах от одной до нескольких единиц. Эта же процедура позволяет быстрее выявить направления, в которых функция неограниченно убывает (такие случаи часто встречаются на практике, когда задачу линейного программирования пытаются решать в пространстве двойственных переменных с использованием схемы декомпозиции по ограничениям, а задача оказывается несовместной). В этом случае шаговый множитель резко возрастает, и это служит сигналом к остановке процесса спуска.

Как показали многочисленные вычислительные эксперименты и практические расчеты, в большинстве случаев при  $\alpha \in [2, 3]$  и указанном выше способе регулировки  $h$  число шагов по направлению в среднем редко превосходит 2, при этом за  $n$  шагов  $r$ -алгоритма точность по функционалу, как правило, улучшается в 3—5 раз. В том случае, когда минимизируемая функция является гладкой, для ускорения сходимости можно применять и более тонкие способы поиска минимума по направлению, например квадратичную аппроксимацию по трем точкам, процесс «золотого сечения» и др. Следует заметить, что в типичных задачах негладкой оптимизации различие в трудоемкости вычисления функции и градиента невелико, поэтому нужно обращать особое внимание на уменьшение числа вычислений функции. В гладких задачах трудоемкость вычисления градиента может во много раз (до  $n$  раз) превосходить трудоемкость вычисления функции, и в такой ситуации можно позволить себе находить минимум по направлению с большой точностью.

В гладком случае хорошо зарекомендовал себя и адаптивный способ регулировки шага по направлению, похожий на приведенный нами выше, с небольшим изменением: если на данной итерации функция приняла уже после первого шага большее значение, то шаговый множитель умножается на заданное число, меньшее единицы (порядка 0,8—0,95). Дело в том, что в гладком случае скорость сходи-

мости может оказаться более быстрой при более точном нахождении минимума по направлению, а дополнительное измельчение шага способствует увеличению точности поиска минимума по направлению.

При проведении сравнительных численных экспериментов по решению задач недифференцируемой оптимизации Лемарешаль в [156] применил модификацию  $r$ -алгоритма, которая отличается от вышеприведенной лишь в формуле (2.49):

$$x_{k+1} = x_k - t_k B_k \tilde{g}_k / \|B_k \tilde{g}_k\|, \quad t_k = t_0 q^k, \quad q < 1$$

( $t_0$  и  $q$  выбираются из эвристических соображений). Фактически здесь получилась смесь двух алгоритмов:  $r$ -алгоритма и ОГС с регуляризацией шага по формуле геометрической прогрессии. И хотя  $r$ -алгоритм в модификации Лемарешаля почти на всех тестовых задачах (кроме одной) показал свое преимущество по сравнению с другими испытываемыми методами (в основном это были методы  $\epsilon$ -субградиентного типа), признать такой способ удачным в общем случае нельзя: хотя он и гарантирует сходимость последовательности  $\{x_k\}_{k=0}^{\infty}$  к предельной точке, эта точка не обязательно окажется точкой минимума при выбранных  $t_0$  и  $q$ .

Несмотря на то что  $r$ -алгоритмы сейчас являются наиболее эффективным и проверенным практически средством решения широкого круга задач недифференцируемой оптимизации, теоретическое обоснование этого класса алгоритмов проведено недостаточно полно.

Отметим некоторые известные результаты в этом направлении.

Первый из них относится к идеализированному варианту  $r$ -алгоритма, когда в формуле (2.49)  $h_k$  выбирается из условия минимума по направлению. Если полученное направление  $-B_k \tilde{g}_k$  не является направлением спуска, то  $h_k = 0$ ,  $x_{k+1} = x_k$  и в качестве  $g_f(x_{k+1})$  выбирается обобщенный градиент, образующий с  $B_k \tilde{g}_k$  неострый угол.

Справедлива следующая теорема [97].

*Теорема 2.18. Пусть  $f(x)$  — почти дифференцируемая кусочно гладкая функция такая, что  $\lim_{\|x\| \rightarrow \infty} f(x) = +\infty$ ,  $x_0$  — заданное начальное приближение,  $\alpha_k = \alpha > 1$  ( $k = 0, 1, 2, \dots$ ) и в  $r$ -алгоритме шаговый множитель выбирается из условия минимума по направлению. Тогда, если последовательность  $\{x_k\}_{k=0}^{\infty}$ , полученная в процессе реализации  $r$ -алгоритма, удовлетворяет условию  $\lim_{k \rightarrow +\infty} \|x_{k+1} - x_k\| = 0$ , то в множестве  $U = \{x : f(x) = f_{\infty}\}$  найдется точка  $x^*$  такая, что множество векторов  $\tilde{G}_f(x^*)$  будет линейно зависимым (здесь  $f_{\infty} = \lim_{k \rightarrow \infty} f(x_k)$ ;  $\tilde{G}_f(x^*)$  — множество почти-градиентов в точке  $x^*$ ).*

Рассмотрим, насколько общими являются условия и выводы теоремы. Так как  $f(x_{k+1}) \geq f(x_k)$  ( $k = 0, 1, \dots$ ) и  $f(x)$  ограничена снизу, то условие сходимости к 0 длины отрезка траектории спуска является естественным, хотя возможно построение «патологических»

примеров, когда оно не выполняется (для метода наискорейшего спуска такие примеры построены, см. [60]).

Наиболее типичная ситуация, связанная с нарушением условия линейной независимости векторов  $G_f(z)$  (исключая точки локально-глобального минимума), — это тот случай, когда  $z$  лежит на границе более чем  $n$  кусков гладкости функции  $f(x)$ . Таких точек, как правило, конечное число, и по отношению к  $r$ -алгоритму эти точки не служат «ловушками» для минимизирующей последовательности. Во всяком случае, многочисленные тестовые эксперименты и решение практических задач вынуждой оптимизации с помощью  $r$ -алгоритма не дали пока ни одного «патологического» случая явной расходимости процесса или сходимости к точке, не являющейся точкой минимума.

Рассмотрим предельный вариант  $r$ -алгоритма, когда  $\beta_k = 0$ ,  $\alpha_k = +\infty$ , а  $h_k$  ( $k = 1, 2, \dots$ ) находится из условия минимума по направлению.

Пусть  $f(x) = (Ax, x)/2$  — положительно определенная квадратичная функция и задано начальное приближение  $x_0$ . Далее,

$$x_1 = x_0 - h_0 g(x_0) = x_0 - \frac{(A^2 x_0, x_0)}{(A^2 x_0, A x_0)} A x_0,$$

$$g(x_1) = A x_0 - \frac{(A^2 x_0, x_0)}{(A^2 x_0, A x_0)} A^2 x_0,$$

$$\Delta_1 = g(x_0) - g(x_1) = \frac{(A^2 x_0, x_0)}{(A^2 x_0, A x_0)} A^2 x_0,$$

$$(x_1, g(x_0) - g(x_1)) = 0.$$

Таким образом, разность градиентов, взятых в точках  $x_0$  и  $x_1$ , оказывается ортогональной вектору  $x_1$ . Так как  $-x_1$  — направление на точку минимума, то разность двух последовательных градиентов оказывается ортогональной направлению на точку минимума. Растяжению пространства в направлении  $\Delta_1 = g(x_1) - g(x_0)$  с  $\alpha = +\infty$  будет соответствовать операция проектирования градиента  $g(x_1)$  на подпространство, ортогональное  $\Delta_1$ . Дальнейший спуск из точки  $x_1$  в соответствии с  $r$ -алгоритмом будет проходить в линейном многообразии, проходящем через  $x_1$  ортогонально  $\Delta_1$ . Это многообразие содержит начало координат — точку минимума  $f(x)$ . По индукции легко показать, что спуск из точки  $x_k$  ( $k=2, 3, \dots$ ) будет проходить в многообразии, ортогональном векторам  $\Delta_1, \Delta_2, \dots, \Delta_k$ , проходящим через начало координат. Таким образом, для квадратичной положительно определенной функции предельный вариант  $r$ -алгоритма сходится к точке минимума не более чем за  $n$  шагов. Его можно рассматривать как проективный вариант метода сопряженных градиентов.

В [89] предельный вариант  $r$ -алгоритма с восстановлением матрицы  $B_k$  (заменой ее единичной матрицей) после каждых  $n$  шагов рассматривается применительно к задаче минимизации выпуклой дважды непрерывно дифференцируемой функции. Доказан следующий результат [97].

**Теорема 2.19.** Пусть выпуклая функция  $f(x)$ , определенная в  $E^n$ , дважды непрерывно дифференцируема в некоторой окрестности  $S$  точки минимума  $x^*$ , причем в этой окрестности матрица вторых производных (гессиан)  $H(x)$  удовлетворяет условию Липшица

$$\|H(x) - H(x')\| \leq L \|x - x'\|, \quad x, x' \in S,$$

а в точке  $x^*$  — положительно определена. Тогда найдется такая окрестность  $S' \subset S$  точки  $x^*$ , что если  $x_0 \in S'$ , то существует число  $c > 0$ , для которого

$$\|x_n - x^*\| \leq c \|x_0 - x^*\|^2;$$

здесь  $x_n$  — точка, получаемая после  $n$  шагов предельного варианта  $r$ -алгоритма ( $\beta_k = 0$ ,  $h_k$  находятся из условия минимума по направлению ( $k = 0, 1, \dots, n-1$ )).

Таким образом, предельный вариант  $r$ -алгоритма, реализующий, по сути, метод проективных сопряженных градиентов, обладает по отношению к большому циклам, состоящим из  $n$  шагов  $r$ -алгоритма, квадратичной скоростью сходимости при обычных условиях гладкости и регулярности.

Мы продемонстрировали «родственность»  $r$ -алгоритмов, разработанных для решения задач негладкой оптимизации, с классом алгоритмов сопряженных градиентов, предназначенных для минимизации гладких функций. Как заметил В. А. Скоков [63],  $r$ -алгоритмы по своей формальной структуре близки к алгоритмам квазиньютоновского типа с переменной метрикой. Для того чтобы показать это, нужно перейти от формул пересчета последовательности матриц  $B_k$  ( $k = 0, 1, 2, \dots$ ) к формулам пересчета последовательности матриц  $H_k = B_k B_k^*$ . В  $r$ -алгоритмах переход от матрицы  $B_k$  к матрице  $B_{k+1}$  и от точки  $x_k$  к  $x_{k+1}$  имеет вид

$$B_{k+1} = B_k R \beta (\xi_k), \quad x_{k+1} = x_k - h_k B_{k+1} \tilde{g}_k,$$

где

$$\xi_k = \frac{B_k^* (g_f(x_k) - g_f(x_{k-1}))}{\|B_k^* (g_f(x_k) - g_f(x_{k-1}))\|} = \frac{B_k^* \Delta_k}{\|B_k^* \Delta_k\|},$$

$$\tilde{g}_k = \frac{B_{k+1}^* g_f(x_k)}{\|B_{k+1}^* g_f(x_k)\|}.$$

Воспользовавшись формулой (2.31), получаем

$$H_{k+1} = B_{k+1} B_{k+1}^* = B_k R \beta^2 (\xi_k) B_k^* = B_k R \beta^2 (\xi_k) B_k^* = B_k (I + (\beta^2 - 1) \xi_k \xi_k^T) \times$$

$$\times B_k^* = H_k + (\beta^2 - 1) \frac{B_k B_k^* \Delta_k \Delta_k^T B_k B_k^*}{(B_k^* \Delta_k, B_k^* \Delta_k)} = H_k + (\beta^2 - 1) \frac{H_k \Delta_k \Delta_k^T H_k}{(H_k \Delta_k, \Delta_k)}. \quad (2.51)$$

Аналогично получаем

$$x_{k+1} = x_k - h_k B_{k+1} \frac{B_{k+1}^* g_f(x_k)}{\|B_{k+1}^* g_f(x_k)\|} = x_k - h_k \frac{H_{k+1} g_f(x_k)}{\sqrt{(H_{k+1} g_f(x_k), g_f(x_k))}}. \quad (2.52)$$

Таким образом,  $r$ -алгоритм с помощью формул (2.51), (2.52) может быть записан в  $H$ -форме. Так как матрицы  $H_k$  являются симметричными, для их запоминания достаточно  $n(n+1)/2$  ячеек. Преобразование  $H_k \rightarrow H_{k+1}$  требует примерно  $2n^2$  операций умножения. Определенные дополнительные осложнения при применении описанной схемы связаны с тем, что из-за накопления погрешностей вычислений при больших  $k$  возмущенная матрица  $\tilde{H}_k$  перестает быть положительно определенной, что препятствует получению решения задачи с высокой точностью. Особенно это заметно при решении задач на ЭВМ с относительно короткими ячейками памяти (например, на ЕС ЭВМ). Поэтому при использовании  $r$ -алгоритма в  $H$ -форме приходится работать со сдвоенными ячейками и двойной точностью вычислений. Первоначальная  $B$ -форма вычислений гораздо более устойчива к погрешностям округления. Поэтому, если требования к точности результатов высоки, нужно использовать  $B$ -форму  $r$ -алгоритма.

**З а м е ч а н и я:** 1. Алгоритмы типа ОГСРП и метод эллипсоидов также могут быть записаны в  $H$ -форме. При использовании такой формы алгоритмов нужно обязательно принимать во внимание возможное накопление погрешностей округления и связанную с этим неустойчивость вычислений при небольшой используемой разрядности. Вычислительные эксперименты с различными формами алгоритма эллипсоидов описаны в [120].

2. Запишем формулу перехода  $H_k \rightarrow H_{k+1}$  в известном методе с переменной метрикой Давидона — Флетчера — Пауэлла (ДФП). Она имеет следующий вид [60]:

$$H_{k+1} = H_k - \frac{H_k \Delta_k \Delta_k^T H_k}{(H_k \Delta_k, \Delta_k)} + \frac{\Delta x_k \Delta x_k^T}{(\Delta x_k, \Delta x_k)}, \quad (2.53)$$

где  $\Delta x_k = x_{k+1} - x_k$ . По внешнему виду формулы (2.51) и (2.53) похожи. При  $\beta = 0$  из (2.51) получаем формулу

$$H_{k+1} = H_k - \frac{H_k \Delta_k \Delta_k^T H_k}{(H_k \Delta_k, \Delta_k)}, \quad (2.54)$$

соответствующую методу проективных сопряженных градиентов [112].

Несмотря на очевидное формальное сходство, нужно иметь в виду, что методы с переменной метрикой и методы сопряженных градиентов построены с учетом возможности удачной аппроксимации минимизируемой функции в окрестности минимума квадратичной функцией. Так, при определенных условиях в методе ДФП последовательность матриц  $H_k$  стремится к матрице, обратной матрице гессиана в точке минимума.  $r$ -алгоритм применим к гораздо более широкому кругу негладких задач, где вопрос о хорошей аппроксимации квадратичной функцией просто не имеет смысла (например, для кусочно-линейной функции, гессиан которой почти везде равен 0).

При решении задач минимизации высокой размерности (порядка  $10^3$  и выше) представление операторов  $B_k$  (или  $H_k$ ) в матричной

форме требует большой памяти, а их преобразование — большого объема вычислений. В то же время может оказаться, что ограничения по времени вычислений не позволяют выполнить большое число итераций  $r$ -алгоритма. В этих условиях может оказаться полезной идея представления операторов  $H_k$  в форме указания отличных от 1 собственных чисел и соответствующих собственных векторов. Техника преобразования информации об операторе  $H_k$  при переходе  $H_k \rightarrow H_{k+1}$  рассмотрена в [99].

## § 5. Методы эллипсоидов

Большое внимание в последние годы привлек так называемый метод эллипсоидов, развитый независимо в работах [93, 95], который можно рассматривать, с одной стороны, как частный случай метода ОГСП с постоянным коэффициентом растяжения пространства и шаговым множителем, меняющимся по формуле геометрической прогрессии; с другой стороны, этот метод представляет собой метод последовательных отсечений, в котором область локализации решения аппроксимируется на каждом шаге эллипсоидом. Предложенный первоначально для решения задач выпуклого программирования, этот метод легко обобщается на более широкий класс задач нахождения аналогов неподвижных точек векторных полей специального вида.

Пусть на  $E^n$  задано векторное поле  $g(x)$ , не обязательно непрерывное,  $g(x) \in E^n$  ( $x \in E^n$ ). Найдем такую точку  $x^*$ , что  $(g(x), x - x^*) \geq 0$  при всех  $x \in E^n$ . Допустим, что эта задача имеет решение, причем известно, что  $x^* \in S(x_0, R)$ , где  $S(x_0, R)$  — замкнутый шар с центром в точке  $x_0$  и радиусом  $R$ .

Рассмотрим следующий итеративный алгоритм решения указанной задачи при  $n > 1$  (при описании этого алгоритма будем предполагать, что  $g(x) \neq 0$  при  $x \neq x^*$ ).

Перед началом вычислений имеем:  $x_0 \in E^n$ ,  $B_0 = I_n$  — единичная матрица,  $h_0 = R/(n+1)$ . Пусть на  $k$ -м шаге мы получили:  $x_k \in E^n$ ,  $B_k$  — матрица  $n \times n$ ,  $h_k > 0$ . На  $(k+1)$ -м шаге вычисляем:

1)  $g(x_k)$ ; если  $g(x_k) = 0$ , то  $x_k$  — искомая точка;

$$2) \quad \xi_k = \frac{B_k^* g(x_k)}{\|B_k^* g(x_k)\|}; \quad (2.55)$$

$$3) \quad x_{k+1} = x_k - h_k B_k \xi_k; \quad (2.56)$$

$$4) \quad B_{k+1} = B_k R_\beta(\xi_k), \quad \beta = \sqrt{(n-1)/(n+1)}, \quad (2.57)$$

где  $R_\beta(\xi_k)$  — оператор растяжения пространства в направлении  $\xi_k$  с коэффициентом  $\beta$ ;

$$5) \quad h_{k+1} = h_k r, \quad r = n/\sqrt{n^2 - 1}. \quad (2.58)$$

Теорема 2.20. Последовательность  $\{x_k\}_{k=0}^{\infty}$ , генерируемая алгоритмом (2.55) — (2.58), удовлетворяет неравенству

$$\|A_k(x_k - x^*)\| \leq h_k(n+1), \quad k = 0, 1, \dots; \quad (2.59)$$

здесь  $A_k = B_k^{-1}$ .

Доказательство. Доказательство проведем индукцией по  $k$ . Для  $k = 0$  неравенство (2.59) переходит в  $\|x_0 - x^*\| \leq R$  и выполняется по предположению. Предположим, что (2.59) выполняется для  $k = \bar{k}$ . Докажем его выполнение для  $k = \bar{k} + 1$ .

Пусть для краткости  $A_k(x_k - x^*) = z_k$ ,  $\alpha = 1/\beta$ ; тогда

$$\begin{aligned} \|z_{\bar{k}+1}\|^2 &= \|R_\alpha(\xi_{\bar{k}})(z_{\bar{k}} - h_{\bar{k}}\xi_{\bar{k}})\|^2 = \|z_{\bar{k}} - h_{\bar{k}}\xi_{\bar{k}} + (\alpha - 1)(z_{\bar{k}} - \\ &\quad - h_{\bar{k}}\xi_{\bar{k}}, \xi_{\bar{k}})\xi_{\bar{k}}\|^2 = \|z_{\bar{k}} + (-\alpha h_{\bar{k}} + (\alpha - 1)(z_{\bar{k}}, \xi_{\bar{k}}))\xi_{\bar{k}}\|^2 = \\ &= \|z_{\bar{k}}\|^2 - 2h_{\bar{k}}(z_{\bar{k}}, \xi_{\bar{k}})\alpha^2 + (\alpha^2 - 1)(z_{\bar{k}}, \xi_{\bar{k}})^2 + \alpha^2 h_{\bar{k}}^2 = \|z_{\bar{k}}\|^2 - \\ &\quad - 2h_{\bar{k}}(z_{\bar{k}}, \xi_{\bar{k}}) \frac{n+1}{n-1} + \frac{2}{n-1}(z_{\bar{k}}, \xi_{\bar{k}})^2 + \frac{n+1}{n-1} h_{\bar{k}}^2. \end{aligned}$$

Заметим, что

$$(z_{\bar{k}}, \xi_{\bar{k}}) = \frac{1}{\|\tilde{g}(x_{\bar{k}})\|} (A_{\bar{k}}(x_{\bar{k}} - x^*), B_{\bar{k}}^*g(x_{\bar{k}})) = \frac{1}{\|\tilde{g}(x_{\bar{k}})\|} (x_{\bar{k}} - x^*, g(x_{\bar{k}})) \geq 0;$$

здесь  $\tilde{g}(x_{\bar{k}}) = B_{\bar{k}}^*g(x_{\bar{k}})$ ,  $(z_{\bar{k}}, \xi_{\bar{k}}) \leq \|z_{\bar{k}}\| \leq h_{\bar{k}}(n+1)$ . Отсюда

$$\frac{2}{n-1}(z_{\bar{k}}, \xi_{\bar{k}})^2 - 2h_{\bar{k}} \frac{n+1}{n-1}(z_{\bar{k}}, \xi_{\bar{k}}) \leq \frac{2}{n-1} [(z_{\bar{k}}, \xi_{\bar{k}})^2 - (z_{\bar{k}}, \xi_{\bar{k}})^2] = 0.$$

Таким образом,

$$\begin{aligned} \|z_{\bar{k}+1}\|^2 &\leq \|z_{\bar{k}}\|^2 + \frac{n+1}{n-1} h_{\bar{k}}^2 \leq (n+1)^2 h_{\bar{k}}^2 + \frac{n+1}{n-1} h_{\bar{k}}^2 = \\ &= (n+1)^2 \frac{n^2}{n^2-1} h_{\bar{k}}^2 = (n+1)^2 h_{\bar{k}+1}^2, \end{aligned}$$

откуда следует справедливость (2.59). Теорема доказана.

Множество точек  $x$ , удовлетворяющих неравенству

$$\|A_k(x_k - x)\| \leq (n+1)h_k = R(n/\sqrt{n^2-1})^k,$$

представляет собой эллипсоид  $\Phi_k$ , объем которого  $v(\Phi_k)$  равен  $v_0 R^n (n/\sqrt{n^2-1})^{nk} / \det A_k$ , где  $v_0$  — объем единичного  $n$ -мерного шара. Получаем

$$\begin{aligned} \frac{v(\Phi_{k+1})}{v(\Phi_k)} &= \frac{(n/\sqrt{n^2-1})^n \det A_k}{\det A_{k+1}} = \frac{(n/\sqrt{n^2-1})^n \det A_k}{\det R_\alpha(\xi_k) \det A_k} = \\ &= \frac{1}{\alpha} \left( \frac{n}{\sqrt{n^2-1}} \right)^n = \sqrt{\frac{n-1}{n+1}} \left( \frac{n}{\sqrt{n^2-1}} \right)^n = q_n < 1. \quad (2.60) \end{aligned}$$

Таким образом, объем эллипсоида, в котором локализуется искомая точка, в соответствии с неравенством (2.60) убывает со скоростью геометрической прогрессии со знаменателем  $q_n$ .

Рассмотрим возможность применения описанной процедуры для решения ряда задач математического программирования.

**1. Задача минимизации выпуклой функции на шаре.** Пусть на  $E^n$  определена выпуклая функция  $f(x)$ . Требуется найти минимальное значение этой функции на шаре  $S(x_0, R) = \{x: \|x - x_0\| \leq R\}$ .

Пусть  $x^* \in S(x_0, R)$  — искомый минимум. Тогда для всех  $x \in S(x_0, R)$

$$(g_f(x), x - x^*) \geq f(x) - f(x^*) \geq 0.$$

Для  $x \notin S(x_0, R)$

$$\begin{aligned} (x - x_0, x - x^*) &= \|x - x_0\|^2 - (x - x_0, x^* - x_0) \geq \\ &\geq \|x - x_0\| (\|x - x_0\| - \|x^* - x_0\|) \geq 0. \end{aligned}$$

Таким образом, можно построить векторное поле  $g(x)$ , удовлетворяющее свойству  $(g(x), x - x_0) \geq 0$  ( $x \in E^n$ ) следующим образом:

$$g(x) = \begin{cases} g_f(x), & x \in S(x_0, R), \\ (x - x_0)/\|x - x_0\|, & x \notin S(x_0, R). \end{cases}$$

Применим для решения нашей задачи алгоритм (2.55) — (2.59), определив  $g(x)$  указанным образом. Тогда в соответствии с формулой (2.60) получим, что объем области, в которой локализован минимум, убывает со скоростью геометрической прогрессии со знаменателем

$$q_n = \sqrt{\frac{n-1}{n+1}} \left( \frac{n}{\sqrt{n^2-1}} \right)^n.$$

Рассмотрим случай, когда  $\min_{x \in E^n} f(x)$  достигается в точке  $x^*$ , являющейся внутренней точкой шара  $S(x_0, R)$ . Так как алгоритм (2.55) — (2.59) применительно к задаче минимизации функции  $f(x)$  фактически реализует метод ОГСРП с небольшими изменениями, то для оценки скорости сходимости по функционалу можно применить теорему 2.12. Получаем следующую оценку:

$$\begin{aligned} \rho_k &= \min_{1 \leq i \leq k} \|B_i^* g(x_i)\| \leq \frac{d_k \sqrt{k(\alpha^2-1)}}{\sqrt{\alpha^{2k/n}-1}}, \quad (2.61) \\ d_k &= \max_{1 \leq i \leq k} \|g(x_i)\|. \end{aligned}$$

Пусть минимум в (2.61) достигается при  $i = i(k)$ . Тогда

$$\begin{aligned} \rho_k &= (x_{i(k)} - x^*, g(x_{i(k)})) = (A_{i(k)}(x_{i(k)} - x^*), B_{i(k)}^* g(x_{i(k)})) \leq \\ &\leq \|A_{i(k)}(x_{i(k)} - x^*)\| \frac{d_k \sqrt{k(\alpha^2-1)}}{\sqrt{\alpha^{2k/n}-1}} \leq h_{i(k)}(n+1) \frac{d_k \sqrt{k(\alpha^2-1)}}{\sqrt{\alpha^{2k/n}-1}} \leq \\ &\leq h_k(n+1) d_k \frac{\sqrt{k(\alpha^2-1)}}{\sqrt{\alpha^{2k/n}-1}} = \frac{R d_k \sqrt{k(\alpha^2-1)}}{\sqrt{1-\beta^{2k/n}}} q_n^{k/n}. \end{aligned}$$



Заметим, что если  $x_{i(k)} \notin S(x_0, R)$ , то  $g(x_{i(k)}) = \frac{x - x_0}{\|x - x_0\|}$  и, так как  $x^*$  лежит внутри  $S(x_0, R)$ ,  $\left(x_{i(k)} - x^*, \frac{x_{i(k)} - x_0}{\|x_{i(k)} - x_0\|}\right) \geq \delta > 0$ . Таким образом, из того, что  $\mu_k \rightarrow 0$ , следует, что, начиная с достаточно больших  $k$ ,  $x_{i(k)} \in S(x_0, R)$ . Но если  $x_{i(k)} \in S(x_0, R)$ , то  $g(x_{i(k)}) = g_f(x_{i(k)})$  и справедливо неравенство

$$f(x_{i(k)}) - f(x^*) \leq (g_f(x_{i(k)}), x_{i(k)} - x^*) \leq \frac{R d_k \sqrt{k(\alpha^2 - 1)}}{\sqrt{1 - \beta^{2k/n}}} q^{k/n}.$$

Таким образом, если  $\min_{x \in E^n} f(x)$  достигается в точке  $x^*$ , являющейся внутренней точкой шара  $S(x_0, R)$ , то при использовании алгоритма вида (2.55) — (2.58) отклонение «рекорда» по функционалу от  $f(x^*)$  удовлетворяет неравенству

$$r_k = \min_{1 \leq i \leq k} (f(x_i) - f(x^*)) \leq \frac{R d_k \sqrt{k(\alpha^2 - 1)}}{\sqrt{1 - \beta^{2k/n}}} q_n^{k/n}.$$

Если при всех  $k = 0, 1, \dots$   $x_k \in S(x_0, R)$ , то можно получить более точную оценку

$$f(x_{k_i}) - f(x^*) \leq c q_n^{k_i/n}$$

для некоторой подпоследовательности  $\{k_i\}_{i=1}^{\infty}$ , используя теорему 2.11. При больших  $n$   $q_n$  хорошо приближается асимптотической формулой  $q_n \approx 1 - 1/(2n)$ . Отсюда следует, что, хотя  $r_k$  сходится примерно со скоростью геометрической прогрессии (если пренебречь медленно растущим множителем  $\sqrt{k}$ ), при больших  $n$  знаменатель этой прогрессии близок к единице, т. е. практически скорость сходимости может оказаться медленной.

Положительным свойством алгоритма (2.55) — (2.58) является то, что гарантированная скорость сходимости зависит лишь от размерности пространства и не требует знания специфических особенностей функции  $f(x)$ .

## 2. Общая задача выпуклого программирования: найти

$$\min f_0(x) \tag{2.62}$$

при ограничениях

$$f_i(x) \leq 0, \quad i = 1, 2, \dots, m; \quad x \in E^n, \tag{2.63}$$

где  $f_v(x)$  — выпуклые функции, определенные на  $E^n$ ,  $g_v(x)$  — соответствующие субградиенты ( $v = 0, 1, \dots, m$ ), причём имеется априорная информация, что оптимальная точка  $x^*$  существует и находится в шаре  $S(x_0, R)$  (формально к системе ограничений (2.63) можно добавить ограничение  $\|x - x_0\| \leq R$ ).

Пусть  $n > 1$ . Рассмотрим поле  $g(x)$ , построенное следующим образом:

$$g(x) = \begin{cases} g_0(x), & \text{если } \max_{1 \leq i \leq m} f_i(x) \leq 0, \\ g_{i^*}(x), & \text{если } \max_{1 \leq i \leq m} f_i(x) > 0. \end{cases} \tag{2.64}$$

Покажем, что  $(g(x), x - x^*) \geq 0$  при всех  $x \in E^n$ . Если  $\max_{1 \leq i \leq m} f_i(x) \leq 0$ , то  $g(x) = g_0(x)$  и  $(g(x), x - x^*) = (g_0(x), x - x^*) \geq f_0(x) - f_0(x^*) \geq 0$ . Если  $\max_{1 \leq i \leq m} f_i(x) > 0$ , то  $g(x) = g_{i^*}(x)$ , причем  $f_{i^*}(x) > 0$ ,  $f_{i^*}(x^*) \leq 0$ , поэтому  $(g(x), x - x^*) = (g_{i^*}(x), x - x^*) \geq f_{i^*}(x) - f_{i^*}(x^*) \geq 0$ . Таким образом,  $(g(x), x - x^*) \geq 0$  при всех  $x \in E^n$ . Используя это неравенство, применим для локализации алгоритм (2.55) — (2.58), вычисляя  $g(x)$  по формуле (2.64).

Из теоремы 2.20 вытекает, что после  $k$  шагов алгоритма (2.55) — (2.58)  $x^*$  будет локализован в эллипсоиде объемом  $v(\Phi_k) = v_0 q_n^k (v_0 - \text{объем шара } S(x_0, R))$  с центром в точке  $x_k$ .

Отметим, что этот результат не изменится, если во второй формуле (2.64) вместо  $g_{i^*}(x)$  брать  $g_{\bar{i}}(x)$ , где  $\bar{i}$  — произвольный индекс, для которого  $f_{\bar{i}}(x) > 0$ .

**3. Задача о седловой точке.** Пусть задана выпукло-вогнутая функция двух векторных переменных  $x \in E^n$ ,  $y \in E^m$ ,  $z = \{x, y\} \in E^n \times E^m \equiv E^{n+m}$ ,  $z^*$  — седловая точка этой функции,  $z_0$  — заданное начальное приближение и априори известно, что  $\|z_0 - z^*\| \leq R$ .

Рассмотрим псевдоградиентное множество

$$G(z) = G_f^x(x, y) \times (-G_f^y(x, y)),$$

где  $G_f^x(x, y)$  — множество частных субградиентов функции  $f(x, y)$ , рассматриваемой как функция от  $x$  при фиксированном  $y$ ;  $-G_f^y(x, y)$  — множество субградиентов от функции  $-f(x, y)$  по  $y$  при фиксированном  $x$ .

Сформируем векторное поле  $g(z)$  следующим образом:

$$g(z) = \{g_f^x(z), -g_f^y(z)\}; \quad g_f^x(z) \in G_f^x(z); \quad g_f^y(z) \in G_f^y(z).$$

Покажем, что  $(g(z), z - z^*) \geq 0$ . Из определения седловой точки следует, что  $f(x, y^*) \geq f(x^*, y^*) \geq f(x^*, y)$ . Далее,

$$\begin{aligned} 0 &\leq f(x, y^*) - f(x^*, y) = f(x, y^*) - f(x, y) + f(x, y) - f(x^*, y) \leq \\ &\leq (g_f^x(z), x - x^*) - (g_f^y(z), y - y^*) = (g(z), z - z^*) \end{aligned}$$

Таким образом, для локализации седловой точки  $z^*$  можем применить алгоритм (2.55) — (2.58).

К алгоритму типа (2.55) — (2.58) можно прийти, исходя из схемы последовательных отсечений. А. Ю. Левин в 1965 г. предложил для минимизации произвольной выпуклой функции и решения задач выпуклого программирования метод центрированных сечений [44]. Суть этого метода состоит в том, чтобы, используя информацию о локализации оптимума в некотором выпуклом замкнутом многограннике, на каждом шаге находить центр тяжести этого многогранника и вычислять субградиент в этой точке. Тогда оптимум будет найдется по определенную сторону от соответствующей опорной гиперплоскости к линии уровня, и часть многогранника, лежащую в

другом полупространстве, отделяемом гиперплоскостью, можем отсечь. Опираясь на геометрическую лемму (см. [44]), можно показать, что на каждом шаге отсекается не менее чем  $(n/(n+1))^n$ -я часть объема многогранника, что приводит к убыванию объема области локализации со скоростью, не меньшей, чем скорость убывания геометрической прогрессии со знаменателем  $Q_n = 1 - (n/(n+1))^n$  (при больших  $n$   $Q_n \approx 1 - 1/e$ ). Однако этот алгоритм при  $n \geq 3$  оказывается практически непригодным, так как вычисление центра тяжести многомерного многогранника — весьма громоздкая процедура. Д. Б. Юдин и А. С. Немировский [95] предложили для решения задачи выпуклого программирования модифицированный метод центрированных сечений (ММЦС). Пусть оптимум  $x^*$  локализован в шаре  $S(x_0, R)$ . Проведем гиперплоскость  $(g(x_0), x - x_0) = 0$ . Область локализации  $x^*$  сужается до полусферы. Опишем вокруг этой полусферы эллипсоид минимального объема, центр которого лежит на луче  $x_0 - hg(x_0)$ ,  $h > 0$ ; эллипсоид оказывается сплюснутым в направлении  $g(x_0)$ . Для того чтобы цикл вычислений замкнуть, нужно превратить эллипсоид в шар, для чего достаточно произвести растяжение пространства в направлении  $g(x_0)$  с определенным коэффициентом. Таким образом, схема отсечения естественным путем приводит к алгоритму градиентного типа с растяжением пространства в направлении градиента.

Следует отметить, что схема отсечения позволяет генерировать целое семейство алгоритмов различной сложности. Например, если производить операцию вложения области локализации  $x^*$  в эллипсоид после двух отсечений, то нам потребуется, вообще говоря, растяжение пространства в двух направлениях. Следует ожидать, что скорость сходимости при этом значительно возрастет по сравнению со скоростью сходимости алгоритма (2.55) — (2.58).

Хотя метод эллипсоидов сходится по функционалу в определенном смысле со скоростью геометрической прогрессии, требуется примерно  $4,6 n^2$  итераций, чтобы гарантировать уменьшение в 10 раз отклонения рекорда по функционалу от оптимального значения. Это практически медленная сходимость. Поэтому были предприняты определенные усилия, чтобы, оставаясь в рамках общей схемы метода эллипсоидов, ускорить сходимость.

При конструировании алгоритмов этого типа обычно используются два обстоятельства.

1) Возможность проведения более «глубоких» отсечений. Например, если мы решаем задачу выпуклого программирования (2.62), (2.63) и в точке  $x_0$  для некоторого  $i = \bar{i}$   $f_{\bar{i}}(x_0) > 0$ , то в качестве отсекающей гиперплоскости можно использовать следующую:

$$P = \left\{ x: (g_{f_{\bar{i}}}(x_0), x - x_0) + \frac{f_{\bar{i}}(x_0)}{\|g_{f_{\bar{i}}}(x_0)\|} = 0 \right\}. \quad (2.65)$$

Если для всех  $i = 1, \dots, m$   $f_i(x_0) \leq 0$  и  $f(x_0) > \bar{f}$ , где  $\bar{f}$  — достигнутый рекорд по функционалу, то в качестве отсекающей гиперплоскости

$$P = \left\{ x : (g_f(x_0), x - x_0) + \frac{(f(x_0) - \bar{f})}{\|g_f(x_0)\|} = 0 \right\}; \quad (2.66)$$

при этом свойство правильности отсечения не нарушается. Указанные в (2.65), (2.66) гиперплоскости отличаются от используемых в простейшем алгоритме (2.55) — (2.58) параллельным сдвигом в сторону более сильного отсечения области локализации. Величину  $\bar{f}$  в формуле (2.66) назовем *константой сдвига*; в (2.65)  $\bar{f} = 0$ .

2) Описание нового эллипсоида вокруг полученной области локализации можно производить после двух или большего числа правильных отсечений предыдущего эллипсоида. При этом следует ожидать более быстрого (в среднем на одно отсечение) уменьшения объема области локализации.

На основе указанных возможностей разработано несколько вариантов алгоритмов с кратными (в основном двойными) и «глубокими» отсечениями [21, 98]. Основную техническую сложность представляет нахождение эллипсоида минимального объема, описанного вокруг тел, получаемых из сферы после двух или большего числа отсечений. Заметим, что после  $r$ -кратного отсечения сферы ( $r < n$ ) оптимальный эллипсоид сохраняет из-за свойств симметрии  $n - r$  одинаковых главных осей, так что для превращения его в сферу потребуется не более чем  $r$  преобразований типа растяжения пространства.

Остановимся подробнее на случае  $r = 2$ .

Стандартная ситуация заключается в следующем: рассматривается задача минимизации выпуклой функции  $f(x)$ , задана  $n$ -мерная сфера  $S_0 = \{x : \|x - x_0\| \leq r_0\}$  — область локализации минимума — и константа сдвига  $\bar{f}$ . Предлагается следующая процедура конструирования отсекающего тела. Вычислим значения  $f(x_0)$  и  $g_f(x_0)$  и получим

$$P(x_0, \bar{f}) = \left\{ x : (g_f(x_0), x - x_0) + \frac{(f(x_0) - \bar{f})}{\|g_f(x_0)\|} = 0 \right\}$$

— первую отсекающую гиперплоскость,  $H(x_0, \bar{f})$  — отсекаемое этой гиперплоскостью полупространство. Вычисляя  $f(\bar{x})$  и  $g_f(\bar{x})$  в точке  $\bar{x}$  интервала  $I_0 = \{x : x = x_0 - tg_f(x_0), t > 0, x \in S_0\}$ , мы получим  $P(\bar{x}, \bar{f})$  — вторую отсекающую гиперплоскость,  $H(\bar{x}, \bar{f})$  — отсекаемое этой гиперплоскостью полупространство (в направлении антиградиента).

Выбор локализирующего тела  $T$  может быть в зависимости от ситуации осуществлен двумя способами:

1) пусть  $(g_f(\bar{x}), g_f(x_0)) > 0$  («односторонние» отсечения); тогда в качестве  $T$  возьмем тот из сегментов  $H(x_0, \bar{f}) \cap S_0$  и  $H(\bar{x}, \bar{f}) \cap S_0$ , который дальше от точки  $x_0$ ;

2) пусть  $(g_f(x_0), g_f(\bar{x})) \leq 0$  («встречное» отсечение). Положим  $W = H(x_0, \bar{f}) \cap H(\bar{x}, \bar{f})$  и  $V = P(x_0, \bar{f}) \cap P(\bar{x}, \bar{f})$  и зафиксируем точку

$v_0 \in V$ , лежащую в плоскости, которая проходит через  $x_0$  параллельно  $g_f(x_0)$  и  $g_f(\bar{x})$ . Если  $v_0 \notin \text{int } S_0$ , то в качестве  $T$  выбирается слой (фигура, получаемая в результате пересечения сферы двумя параллельными гиперплоскостями) минимальной толщины, содержащий  $W \cap S_0$ . Если  $v_0 \in \text{int } S_0$ , то в качестве  $T$  выбирается так называемая  $s$ -пирамида  $\Pi = W \cap S_0$ . Таким образом, выбор точки  $\bar{x} \in l_0$  и  $g(\bar{x})$  однозначно определяют форму локализирующего тела: сегмент, слой или  $s$ -пирамида. Для этих фигур выведены формулы, задающие параметры описанных эллипсоидов минимального объема.

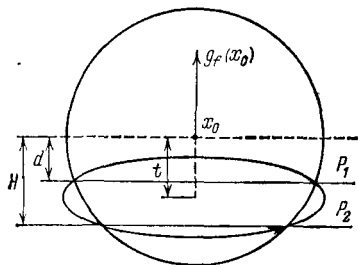


Рис. 2.2

Дадим вывод соответствующих формул для слоя. Пусть слой задается двумя параллельными гиперплоскостями  $P_1$  и  $P_2$  (основания слоя). Обозначим расстояния от  $x_0$  до  $P_1$  и  $P_2$  соответственно через  $d$  и  $H$  ( $d < H$ ). Из геометрических соображений ясно, что описанный эллипсоид минимального объема  $\Phi$  будет представлять собой тело вращения (вокруг оси, проходящей через

точку  $x_0$  в направлении  $g_f(x_0)$ ) соответствующей размерности, поверхность которого проходит по множеству точек пересечения поверхности сферы с плоскостями  $P_1$  и  $P_2$  (рис. 2.2). Для определения параметров эллипсоида  $\Phi$  достаточно найти величину  $t$  расстояния от центра  $\Phi$  до верхней полуплоскости  $P_1$ .

Пусть координатные оси  $x_1, \dots, x_{n-1}$  соответствуют  $n-1$  одинаковым осям эллипсоида, ось  $y$  коллинеарна  $g_f(x_0)$ , а начало координат совпадает с центром  $\Phi$ . Тогда в этой системе координат поверхность эллипсоида будет удовлетворять следующему уравнению:

$$\sum_{i=1}^{n-1} \frac{x_i^2}{a^2} + \frac{y^2}{b^2} = 1.$$

Далее, она проходит через точки с координатами  $(\sqrt{R^2 - d^2}, 0, \dots, 0, t)$  и  $(\sqrt{R^2 - H^2}, 0, \dots, 0, H - d - t)$ . Учитывая это, определим  $a^2$  и  $b^2$ :

$$\frac{(R^2 - d^2)}{a^2} + \frac{t^2}{b^2} = 1, \quad \frac{(R^2 - H^2)}{a^2} + \frac{(H - d - t)^2}{b^2} = 1,$$

откуда

$$a^2 = R^2 - d^2 + \frac{t^2(H+d)}{(H-d-2t)}, \quad b^2 = \frac{a^2(H-d-2t)}{(H+d)}, \quad \beta = \frac{b}{a}, \quad (2.67)$$

Заметим, что  $H - d - 2t > 0$ , т. е.  $t < \frac{(H-d)}{2}$ . Далее имеем

$$v^2(\Phi) = c_n a^{2(n-1)} b^2 = c_n a^{2n} \frac{(H-d-2t)}{(H+d)};$$

$$\ln v^2(\Phi) = \bar{c} + n \ln \left( R^2 - d^2 + \frac{t^2(H+d)}{(H-d-2t)} \right) + \ln \left( \frac{(H-d-2t)}{(H+d)} \right).$$

Определим оптимальное значение  $t$ , приравняв производную  $\ln v^2(\Phi)$  нулю. Получим уравнение

$$\frac{n(2t(H+d) - 2(R^2 - d^2))}{(R^2 - d^2)(H - d - 2t) + t^2(H+d)} + \frac{2(n-1)}{H-d-2t} = 0,$$

которое преобразуется в квадратное:

$$t^2(H+d)(n+1) - 2t\left((R^2 - d^2) + \frac{(H^2 - d^2)n}{2}\right) + (R^2 - d^2)(H-d) = 0. \quad (2.68)$$

Меньший корень  $t^*$  указанного квадратного уравнения соответствует эллипсоиду минимального объема (большой корень не удовлетворяет условию  $t < (H-d)/2$ ).

Получив  $t^*$ , легко получаем центр  $x_\Phi$  описанного эллипсоида  $\Phi$ :

$$x_\Phi = x_0 - (t^* + d) \frac{g_f(x_0)}{\|g_f(x_0)\|}.$$

По формулам (2.67), (2.68) вычисляем  $a$  и  $b$ . Для того чтобы превратить эллипсоид в сферу, достаточно произвести растяжение пространства в направлении  $\frac{g_f(x_0)}{\|g_f(x_0)\|}$  с коэффициентом растяжения  $\alpha = \frac{a}{b} = \sqrt{\frac{H+d}{(H-d-2t^*)}}$ . Эти формулы мы можем использовать для уточнения алгоритма ОГСРП (2.32) — (2.35) в том случае, когда известны значения функции в точке минимума  $f^*$ , «константа роста» функции  $M$  в неравенстве

$$(g_f(x), x - x^*) \leq M(f(x) - f(x^*)), \quad M \geq 1, \quad (2.69)$$

и оценка расстояния до точки минимума:

$$\|x_0 - x^*\| \leq R. \quad (2.70)$$

Из (2.69), (2.70) следует, что  $x^*$  принадлежит некоторому слою, заключенному между двумя параллельными гиперплоскостями  $P_1$  и  $P_2$ , где

$$P_1 = \{x : (g_f(x_0), x - x_0) = f(x_0) - f^*\},$$

$$P_2 = \{x : (g_f(x_0), x - x_0) = M(f(x_0) - f^*)\}.$$

Расстояние от  $x_0$  до верхнего основания  $P_1$  равно

$$d = \frac{(f(x_0) - f^*)}{\|g_f(x_0)\|},$$

до нижнего основания  $P_2$  — равно

$$H = \frac{M(f(x_0) - f^*)}{\|g_f(x_0)\|}.$$

Подставив  $H = Md$  в (2.68), получаем квадратное уравнение

относительно  $t$ :

$$t^2 - 2t \left( \frac{R^2 - d^2}{(n+1)(M+1)d} + \frac{(M-1)nd}{2(n+1)} \right) + \frac{(R^2 - d^2)(M-1)}{(n+1)(M+1)} = 0. \quad (2.71)$$

Искомый корень  $t^*$  представляется в виде

$$t^* = \frac{nd}{2(n+1)(M+1)} \left( \mu + \frac{n+1}{n} (M^2 - 1) - \sqrt{\mu^2 + \frac{n^2 - 1}{n} (M^2 - 1)^2} \right), \quad (2.72)$$

где

$$\mu = 2 \frac{R^2 - d^2}{nd^2} - \frac{M^2 - 1}{n}. \quad (2.73)$$

Подставляя (2.72) в (2.67), получаем

$$\beta^2 = \frac{b^2}{a^2} = \frac{n}{(n+1)(M+1)^2} \left( \sqrt{\mu^2 + \frac{n^2 - 1}{n^2} (M^2 - 1)^2} - \mu \right).$$

Радиус нового шара после операции растяжения равен большой полуоси эллипсоида  $\Phi$ :

$$a = \frac{\alpha n}{\sqrt{2(n^2 - 1)}} \sqrt{\mu n + \sqrt{\mu^2 + \frac{(n^2 - 1)}{n^2} (M^2 - 1)^2}}. \quad (2.74)$$

Отношение  $q$  объема  $\Phi$  к объему  $S_0$  задается формулой

$$q = \beta (a/R)^n. \quad (2.75)$$

Подставляя в нее выражения для  $\beta$  и  $a$ , получаем:

$$q^2 = \frac{n}{n+1} \left( \frac{n^2}{n^2 - 1} \right)^n \left( \frac{\mu n + \sqrt{\mu^2 + \frac{n^2 - 1}{n^2} (M^2 - 1)^2}}{\mu n + M^2 + 1} \right)^n \times \\ \times \frac{\sqrt{\mu^2 + \frac{n^2 - 1}{n^2} (M^2 - 1)^2} - \mu}{(M+1)^2}. \quad (2.76)$$

При больших  $n$  справедлива асимптотическая формула

$$q^2 = \frac{\sqrt{\mu^2 + (M^2 - 1)^2} - \mu}{(M+1)^2} \times \\ \times \exp \left\{ - \frac{\sqrt{\mu^2 + (M^2 + 1)^2} - (M^2 + 1)}{\mu} \right\} + O \left( \frac{1}{n} \right).$$

Максимум по  $\mu$  выражения, стоящего в правой части формулы (2.76), достигается при  $\mu = 2M$  и равен  $((M-1)/(M+1))^2$ . Таким образом, если константа роста  $M$  невелика и известно  $f^*$ , то использование шаровых слоев может обеспечить гораздо большую скорость сходимости, чем простейший вариант метода эллипсоидов.

Отметим, что при  $H = R$  вместо шарового слоя получается сегмент, при этом уравнение (2.68) приобретает вид

$$(n+1)t^2 - t(R-d)(n+2) + (R-d)^2 = 0. \quad (2.77)$$

Пусть  $R-d = h$ . Меньший корень уравнения (2.77)  $t^* = h/(n+1)$ . При этом

$$\alpha = \sqrt{\frac{n+1}{n-1}} \sqrt{\frac{2R-h}{h}}, \quad (2.78)$$

$$b = \frac{nh}{n+1}, \quad a = \frac{n}{\sqrt{n^2-1}} \sqrt{(2R-h)h}. \quad (2.79)$$

Уменьшение объема за одну итерацию равно

$$q = \frac{a^{n-1}b}{R^n} = \frac{h}{R} \left( \left( 2 - \frac{h}{R} \right) \frac{h}{R} \right)^{(n-1)/2} \left( \frac{n}{\sqrt{n^2-1}} \right)^{n-1} \frac{n}{n+1}.$$

Пусть задана система из  $m$  ( $m \leq n$ ) неравенств

$$(g_i, x) \leq f_i, \quad i = 1, \dots, m, \quad (2.80)$$

таких, что векторы  $g_1, g_2, \dots, g_m$  линейно независимы и  $(g_i, g_j) \leq 0$  при  $i \neq j$ . Тело

$$W = \{x: x \in E^n, (g_i, x) \leq f_i, i = 1, \dots, m\}, \quad (2.81)$$

представляющее собой пересечение полупространств, задаваемых неравенствами (2.80), непусто. Далее, множество

$$V = \{x: x \in E^n, (g_i, x) = f_i, i = 1, \dots, m\}$$

является  $(n-m)$ -мерным линейным многообразием,  $V \subseteq W$ . Для характеристики взаимного расположения  $W$  и  $n$ -мерного шара  $S_0 = \{x: \|x - x_0\| \leq r_0\}$  выделим точки  $v_0$  и  $w_0$ , определяемые соотношениями

$$\|x_0 - v_0\| = \min_{x \in V} \|x_0 - x\|, \quad \|x_0 - w_0\| = \min_{x \in W} \|x_0 - x\|,$$

т. е.  $v_0$  является проекцией  $x_0$  на линейное многообразие  $V$ . При заданном шаре  $S_0$  будем рассматривать лишь такие тела  $W$  вида (2.81), для которых выполняются условия

$$v_0 \in \text{int } S_0, \quad x_0 \notin \text{int } W. \quad (2.82)$$

При выполнении для тела  $W$  вида (2.79) и шара  $S_0 = \{x: \|x - x_0\| \leq r_0\}$  условий (2.82) тело  $\Pi = W \cap S_0$  назовем  $s$ -пирамидой, образованной на основе шара  $S_0$ , или просто  $s$ -пирамидой. Точку  $v_0$  назовем *вершиной*  $s$ -пирамиды  $\Pi$ . Если  $v_0 = w_0$ , то  $s$ -пирамиду назовем *регулярной*, в противном случае — *нерегулярной*.

Для аналитического описания  $s$ -пирамиды нам понадобятся некоторые конструкции. Пусть  $L(a_1, a_2, \dots, a_l)$  — линейное пространство, определяемое набором линейно независимых векторов  $a_i$ ,



$a_2, \dots, a_l \in E^n$ ,  $L^*(a_1, a_2, \dots, a_l)$  — ортогональное дополнение к  $L(a_1, a_2, \dots, a_l)$ ,  $K(a_1, a_2, \dots, a_l) = \left\{ x : x = \sum_{i=1}^l \lambda_i a_i, \lambda_i \geq 0, i = 1, \dots, l \right\}$  — многогранный конус, натянутый на «остов»,  $a_1, a_2, \dots, a_l$  — «ребра» конуса. Через  $\Gamma(a_1, a_2, \dots, a_l)$  обозначим матрицу Грама векторов  $a_1, a_2, \dots, a_l$ :

$$\Gamma(a_1, a_2, \dots, a_l) = \{r_{ij}\}_{i,j=1}^l, \quad r_{ij} = (a_i, a_j).$$

Известно, что  $\det \Gamma(a_1, a_2, \dots, a_l) \geq 0$ , причем для линейно независимых векторов  $a_1, \dots, a_l$  неравенство строгое.

Пусть  $b = (b^{(1)}, \dots, b^{(n)})$  — произвольный вектор. Обозначим через  $N(b, L(a_1, \dots, a_l))$  ортогональную составляющую этого вектора к подпространству  $L(a_1, \dots, a_l)$ . Если  $b_0 = N(b, L(a_1, \dots, a_l))$ , то  $b - b_0 = \sum_{i=1}^l \lambda_i a_i$ ; при этом  $\lambda_i$  находятся из системы уравнений

$$\sum_{i=1}^l \lambda_i (a_i, a_j) = (b, a_j), \quad j = 1, \dots, l, \quad (2.83)$$

матрица коэффициентов которой совпадает с матрицей Грама  $\Gamma(a_1, \dots, a_l)$ . Отсюда для  $b_0 = N(b, L(a_1, \dots, a_l))$  получаем следующее выражение [16]:

$$N(b, L(a_1, \dots, a_l)) = \frac{1}{\det \Gamma(a_1, \dots, a_l)} \left\{ \begin{array}{c} \Gamma(a_1, \dots, a_l) \\ \vdots \\ a_l^j \\ (b, a_1) \dots (b, a_l) \quad b^j \end{array} \right\}_{j=1}^n, \quad (2.84)$$

в котором  $a_i^j, b^j$  —  $j$ -е координаты векторов  $a_i$  и  $b$  соответственно.

В определении тел  $V$  и  $W$  участвуют линейно независимые вектора  $g_1, \dots, g_m$  такие, что  $(g_i, g_j) \leq 0$  ( $i \neq j$ ). Ясно, что  $V = v_0 + L^*(g_1, \dots, g_m)$ . Если  $q_1, q_2, \dots, q_{n-m}$  — некоторый базис в подпространстве  $L^*(g_1, \dots, g_m)$ , то

$$V = v_0 + L(q_1, \dots, q_{n-m}).$$

Рассмотрим векторы

$$p_i = -N(g_i, L(g_1, \dots, g_{i-1}, g_{i+1}, \dots, g_m)).$$

Эти векторы являются образующими конуса, который получается при пересечении тела  $W$  с  $L(g_1, \dots, g_m)$ . Легко показать, что  $p_1, \dots, p_m$  линейно независимы,  $(p_i, p_j) \geq 0$ ,  $L(g_1, \dots, g_m) = L(p_1, \dots, p_m)$ .

Тело  $W$  запишется в следующем виде:

$$W = v_0 + K(p_1, \dots, p_m) + L^*(p_1, \dots, p_m).$$

Зная строение тел  $V$  и  $W$ , нетрудно описать строение  $s$ -пирамиды  $\Pi$ , которую можно представить в виде «дольки», вырезаемой в шаре  $S_0$  совокупностью из  $m$  гиперплоскостей. Гиперплоскости попарно расположены под нетупыми углами и пересекаются по  $(n - m)$ -мерному линейному многообразию

$$V = v_0 + L(q_1, \dots, q_{n-m}) = L^{n-m}.$$

Многообразие  $L^{n-m}$  имеет общие точки с внутренностью шара  $S_0$ , и среди таких точек ближайшей к  $x_0$  является точка  $v_0$  — вершина  $s$ -пирамиды  $\Pi$ . Далее,  $\Pi$  симметрична относительно ортогонального к  $L^{n-m}$  и проходящего через  $x_0$  линейного многообразия

$$L^m = V_0 + L(p_1, \dots, p_m).$$

В пересечении  $\Pi$  с  $L^m$  образуется  $m$ -мерное тело

$$\pi = [v_0 + K(p_1, \dots, p_m)] \cap S_0,$$

которое будем называть *c-пирамидой*, соответствующей  $s$ -пирамиде  $\Pi$  с вершиной  $v_0$ . Тело  $\pi$  представляет собой часть  $m$ -мерного конусообразного тела (заключенную в шаре  $S_0$ ), полученного сдвигом конуса  $K(p_1, \dots, p_m)$  на вектор  $v_0$ . Ребра  $p_1, \dots, p_m$  конуса  $K(p_1, \dots, p_m)$  могут быть выбраны путем умножения на положительные скаляры так, чтобы  $v_0 + p_1, \dots, v_0 + p_m$  принадлежали сферической поверхности шара  $S_0$ . Всюду ниже будем предполагать, что векторы  $p_1, \dots, p_m$  удовлетворяют указанному условию. При таком выборе ребер однозначно определен  $m$ -мерный симплекс  $M$  с вершинами в точках  $v_0, v_0 + p_1, \dots, v_0 + p_m$ .

Введем следующие обозначения:  $t(a_0; a_1, \dots, a_m)$  —  $m$ -мерный симплекс с вершинами в точках  $a_0, a_0 + a_1, \dots, a_0 + a_m$ ;  $c(a_0; a_1, \dots, a_m)$  —  $c$ -пирамида с вершиной  $a_0$  и ребрами  $a_1, \dots, a_m$ ;  $s(a_0; a_1, \dots, a_m)$  — соответствующая  $s$ -пирамида.

Очевидно, что

$$M = t(v_0; p_1, \dots, p_m);$$

$$\pi = c(v_0; p_1, \dots, p_m);$$

$$\Pi = s(v_0; p_1, \dots, p_m); \quad M \subset \pi \subset \Pi.$$

**Лемма 2.1.** Пусть  $\Pi = s(v_0; p_1, \dots, p_m)$  — регулярная  $s$ -пирамида, образованная на основе шара  $S_0 = \{x: \|x - x_0\| \leq r_0\}$ . Тогда  $\|p_i\| \leq r_0$  ( $i = 1, \dots, m$ ).

**Доказательство.** Так как  $\|v_0 - x_0 + p_i\| = r$ , достаточно показать, что  $(v_0 - x_0, p_i) \geq 0$ . Рассмотрим множество точек вида  $x(\lambda) = v_0 + \lambda p_i$ ,  $\lambda \in (0, 1)$ . Очевидно,  $x(\lambda) \in W$ . Имеем

$$\|x_0 - x(\lambda)\|^2 = \|x_0 - v_0\|^2 + \lambda \|p_i\|^2 (\lambda + 2(v_0 - x_0, p_i) / \|p_i\|^2).$$

Если  $(v_0 - x_0, p_i) < 0$ , то при достаточно малом  $\lambda > 0$   $\|x_0 - x(\lambda)\|^2 \leq \|x_0 - v_0\|^2$ , что противоречит регулярности  $s$ -пирамиды  $\Pi$ . Таким образом,  $(v_0 - x_0, p_i) \geq 0$ , откуда вытекают оценки  $\|p_i\| \leq r_0$  ( $i = 1, \dots, m$ ).

Без ограничения общности можно считать, что все используемые выше векторы из  $E^n$  заданы координатами в ортонормированном базисе  $e_1, e_2, \dots, e_n$  таким, что

$$\begin{aligned} L(e_1, e_2, \dots, e_m) &= L(p_1, p_2, \dots, p_m), \\ L(e_{m+1}, e_{m+2}, \dots, e_n) &= L(q_1, q_2, \dots, q_{n-m}). \end{aligned}$$

Поэтому у любого вектора  $p_i$  ( $i = 1, 2, \dots, m$ ) ( $n - m$ ) последних компонент равны 0, т. е.  $p_i = \{p_{1i}, \dots, p_{mi}, 0, \dots, 0\}$ . Поэтому эти векторы можно рассматривать как векторы в  $L(e_1, \dots, e_m)$  —  $m$ -мерном подпространстве, в котором задана  $s$ -пирамида  $\pi$ . Введем матрицу  $P = \{p_{ij}\}_{i,j=1}^m$ , элементы  $i$ -го столбца которой совпадают с первыми  $m$  компонентами вектора  $p_i$ .

**Теорема 2.21.** Пусть локализирующее тело — регулярная  $s$ -пирамида  $\Pi = s(v_0; p_1, p_2, \dots, p_m)$ , образованная на основе шара  $S_0 = \{x: \|x - v_0\| \leq r_0\}$ . Тогда существует включающий  $\Pi$   $n$ -мерный эллипсоид  $\Phi(\Pi)$  с центром в точке  $v_0$  такой, что

$$q(\Pi) = \frac{v(\Phi(\Pi))}{v(S_0)} = \left( \sqrt{1 - \frac{\|x_0 - v_0\|^2}{r_0^2}} \right)^{n-m} \frac{|\det P|}{r_0^m}. \quad (2.85)$$

При этом в некоторой ортогональной системе координат (с началом в точке  $v_0$ ) в  $E^n$  эллипсоид  $\Phi(\Pi)$  задается неравенством

$$\sum_{i,j=1}^m q_{ij} z_i z_j + (r_0^2 - \|x_0 - v_0\|^2)^{-1} \sum_{i=m+1}^n z_i^2 \leq 1,$$

причем параметры  $q_{ij}$  являются элементами симметричной матрицы

$$Q = \{q_{ij}\}_{i,j=1}^m = (P^{-1})^* P^{-1}.$$

**Доказательство.** Пусть  $\Phi$  — семейство  $n$ -мерных эллипсоидов, имеющих своим центром точку  $v_0$ . Построим (минимальный по объему) эллипсоид  $\Phi(\Pi) \in \Phi$ , включающий  $\Pi$ . При этом мы будем учитывать симметрию  $\Pi$  относительно линейного многообразия  $L^m$ . Из  $n$  главных осей  $\Phi(\Pi)$   $m$  связаны с линейным многообразием  $L^m$ , а остальные  $n - m$  — с линейным многообразием  $L^{n-m}$ . Величины полуосей  $\Phi(\Pi)$  обозначим  $a_1, \dots, a_n$ . Значения величин  $a_{m+1}, \dots, a_n$  в силу симметрии равны между собой и вычисляются по формуле  $a_i = \rho = \sqrt{r_0^2 - \|x_0 - v_0\|^2}$ ,  $i = m + 1, \dots, n$ . Меньшими  $a_i$  не могут быть, так как  $\rho$  — расстояние от  $v_0$  до принадлежащих  $\Pi$  точек пересечения  $L^{n-m}$  со сферической поверхностью шара  $S_0$ . Поэтому минимизация объема  $v(\Phi(\Pi))$  сводится к минимизации объема  $m$ -мерного эллипсоида с центром в точке  $v_0$ , описанного вокруг  $s$ -пирамиды  $\pi = s(v_0; p_1, \dots, p_m)$ .

Рассмотрим более простую задачу — найти  $m$ -мерный эллипсоид с центром в точке  $v_0$ , включающий симплекс  $M = t(v_0; p_1, \dots, p_m)$ . Все  $m$ -мерные невырожденные симплексы аффинно подобны, то есть для любой пары  $M_1$  и  $M_2$  таких симплексов существует невырожденное линейное преобразование  $A$  такое, что  $AM_1 = M_2$ . Отношение

объемов тел в  $E^m$  инвариантно по отношению к таким преобразованиям. Поэтому достаточно построить оптимальный эллипсоид для одного «стандартного» симплекса, а затем при рассмотрении конкретного симплекса применить соответствующее линейное преобразование. В качестве стандартного возьмем симплекс  $M_0 = t(v_0; e_1, \dots, e_m)$  с ребрами из базисных векторов. В силу симметрии оптимальным эллипсоидом с центром в точке  $v_0$ , включающим  $M_0$ , будет шар

$$S_0^m = \{x : \|x - v_0\| \leq 1; x \in L^m\}.$$

Рассмотрим в  $E^m$  невырожденное линейное преобразование  $A$ , переводящее  $M$  в  $M_0$ . Имеем  $Ap_i = e_i$  ( $i = 1, 2, \dots, m$ ), или  $AP = I_m$ ; отсюда  $A^{-1} = P$ . Преобразование  $A^{-1}$  переводит  $M_0$  в  $M$ , а  $m$ -мерный единичный шар в  $m$ -мерный эллипсоид, полуоси которого соответствуют полуосям оптимального эллипсоида  $\varphi(M)$ , включающего  $M$ . При этом для произведения полуосей эллипсоида  $\varphi(M)$  имеем

$$a_1 \times a_2 \times \dots \times a_m = \frac{v(\varphi(M))}{v(S_0^m)} = \frac{v(M)}{v(M_0)} = |\det A^{-1}| = |\det P|.$$

В  $L^m$  перейдем к системе координат с центром в точке  $v_0$  и с осями  $e_1, \dots, e_m$ . В этой системе координат эллипсоид  $\varphi(M)$  описывается неравенством  $(Qz, z) \leq 1$ , где  $Q$  — положительно определенная матрица. Это неравенство получается из неравенства для шара  $(y, y) \leq 1$  после подстановки  $y = Az = P^{-1}z$ .

Таким образом,  $(Qz, z) = (P^{-1}z, P^{-1}z) = ((P^{-1})^*P^{-1}z, z)$ , откуда  $Q = (P^{-1})^*P^{-1}$ .

Теперь покажем, что эллипсоид  $\varphi(M)$  включает не только симплекс  $M$ , но и всю  $c$ -пирамиду  $\pi$ . Рассмотрим произвольную точку  $\bar{y}$  на сферической части границы тела  $\pi_0 = c(0; e_1, \dots, e_m)$ . Она может быть представлена в следующем виде:

$$\bar{y} = \sum_{i=1}^m \lambda_i e_i; \quad \lambda_i \geq 0; \quad \sum_{i=1}^m \lambda_i^2 = 1.$$

Образ этой точки в  $\pi = c(v_0; p_1, \dots, p_m)$  имеет вид

$$\bar{x} = v_0 + \sum_{i=1}^m \lambda_i p_i; \quad \sum_{i=1}^m \lambda_i^2 = 1; \quad \lambda_i \geq 0, \quad i = 1, \dots, m.$$

Для того чтобы доказать, что  $\varphi(M) \supset \pi$ , достаточно показать, что  $\|\bar{x} - x_0\| \geq r_0$ . Без ограничения общности можно считать, что  $x_0 = 0$ . Тогда  $\|p_i + v_0\| = r_0$  ( $i = 1, \dots, m$ ). Далее,

$$(p_i, v_0) \geq 0; \quad (p_i, p_j) \geq 0, \quad i, j = 1, \dots, m, \quad (2.86)$$

$$\begin{aligned} \|\bar{x}\|^2 &= \left\| v_0 + \sum_{i=1}^m \lambda_i p_i \right\|^2 \geq \|v_0\|^2 + \sum_{i=1}^m \lambda_i^2 \|p_i\|^2 + 2 \left( v_0, \sum_{i=1}^m \lambda_i p_i \right) = \\ &= \|v_0\|^2 + \sum_{i=1}^m \lambda_i^2 \|p_i\|^2 + 2 \sum_{i=1}^m \lambda_i^2 (v_0, p_i) + 2 \sum_{i=1}^m (\lambda_i - \lambda_i^2) (v_0, p_i). \end{aligned}$$

Из  $\lambda_i \geq \lambda_i^2$  и (2.86) получаем  $\|\bar{x}\|^2 \geq \sum_{i=1}^m \lambda_i^2 \|v_0 + p_i\|^2 \geq r_0^2$ , что и требовалось доказать.

Оценим отношение объемов  $v(\Phi(\Pi))/v(S_0)$ . Эллипсоид  $\Phi(\Pi)$  в системе координат, у которой ортами являются  $e_1, \dots, e_n$ , а начало находится в точке  $v_0$ , задан неравенством

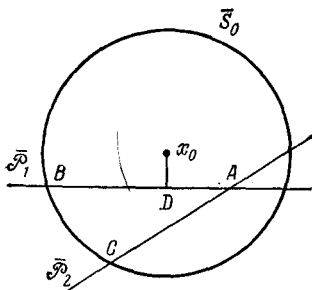


Рис. 2.3

$$(Qz, z) + (r_0^2 - \|x_0 - v_0\|^2)^{-1} \sum_{i=m+1}^n z_i^2 \leq 1, \quad (2.87)$$

где  $Q = (P^{-1})^* P^{-1}$ .

Отсюда объем эллипсоида  $\Phi(\Pi)$

$$v(\Phi(\Pi)) = k_n \sqrt{\frac{r_0^2 - \|x_0 - v_0\|^2}{|\det Q|}}^{n-m}$$

где  $k_n$  — объем шара радиуса 1 в пространстве  $E^n$ .

Получаем равенство

$$q = \frac{v(\Phi(\Pi))}{v(S_0)} = \frac{1}{r_0^m} \left(1 - \frac{\|x_0 - v_0\|^2}{r_0^2}\right)^{(n-m)/2} |\det P|.$$

Теорема 2.21 доказана.

Величина  $|\det P|$  выражает объем параллелепипеда, построенного на векторах  $p_1, \dots, p_m$ , и может быть значительно меньше, чем

$\prod_{i=1}^m |p_i| \leq r_0^m$ . Таким образом,  $q$  для регулярной  $s$ -пирамиды может

значительно отличаться от 1 даже при больших  $n$ .

При  $m = 2$   $|\det P| = |p_1| \cdot |p_2| \cdot \sin(\widehat{p_1, p_2})$  и, следовательно,  $q \leq \sin(\widehat{p_1, p_2})$ .

К сожалению, не всегда  $s$ -пирамида является регулярной. Построение оптимального эллипсоида, включающего нерегулярную  $s$ -пирамиду, сопряжено с определенными осложнениями, поэтому мы рассмотрим этот вопрос только для  $m = 2$ , тем более, что при  $m > 2$  алгоритм, основанный на методе эллипсоидов, резко усложняется в целом.

Пусть  $\Pi = s(v_0; p_1, p_2)$  — нерегулярная  $s$ -пирамида. Рассмотрим плоскость  $L^2$ , в которой расположена  $s$ -пирамида  $\pi = c(v_0; p_1, p_2)$  (см. рис. 2.3). Прямые  $\overline{\mathcal{P}}_1, \overline{\mathcal{P}}_2$  и окружность  $\overline{S}_0$  представляют собой линии пересечения с  $L^2$  секущих гиперплоскостей  $\mathcal{P}_1$  и  $\mathcal{P}_2$  (с градиентами  $g_1$  и  $g_2$  соответственно) и поверхности шара  $S_0$ ,  $A = v_0$  — точка пересечения прямых  $\overline{\mathcal{P}}_1$  и  $\overline{\mathcal{P}}_2$ ,  $D = w_0$  — основание перпендикуляра, опущенного из  $x_0$  на  $\overline{\mathcal{P}}_1$ , ближайшую отсекающую гиперплоскость.

Положим  $\rho_1 = \vec{AB}$ ,  $\rho_2 = \vec{AC}$  и определим

$$\begin{aligned}\hat{\rho}_1 &= \vec{DB} = \rho_1 + (v_0 - w_0), \\ \hat{\rho}_2 &= \vec{DC} = \rho_2 + (v_0 - w_0).\end{aligned}$$

Получаем  $w_0 - v_0 = \gamma \hat{\rho}_1$ , где  $0 < \gamma < 1$ .

Рассмотрим определяемую векторами  $\hat{\rho}_1$  и  $\hat{\rho}_2$   $s$ -пирамиду  $\hat{\Pi} = s(w_0; \hat{\rho}_1, \hat{\rho}_2)$ ;  $\hat{\Pi}$  регулярна по построению. Построим оптимальный эллипсоид  $\Phi(\hat{\Pi})$ , включающий  $\hat{\Pi}$ . Ниже даются условия, при которых  $\Phi(\hat{\Pi})$  включает  $\Pi$ .

**Теорема 2.22.** При выполнении неравенства

$$\gamma(\hat{\rho}_1, \hat{\rho}_2) + (\gamma + 1)(\hat{\rho}_1, x_0 - w_0) \leq 0 \quad (2.88)$$

справедливо включение  $\Pi \subset \Phi(\hat{\Pi})$ .

**Доказательство.** Рассмотрим множество  $X$  «критических» точек, которые лежат на поверхности шара  $S_0$  и при ортогональном проектировании на  $L^2$  попадают на отрезок  $AB$  (рис. 2.3). Для доказательства теоремы достаточно проверить включение  $X \subset \Phi(\hat{\Pi})$ .

Введем в  $E^n$  систему координат с началом в точке  $w_0$  и осями  $e_1, \dots, e_n$ , причем  $e_1, e_2 \in L^2$ . Точке  $x_0$  соответствует в этой системе координат вектор  $\hat{x}_0 = x_0 - w_0$ . Любой вектор  $x \in E^n$  представим в виде  $x = z + y$ , где  $z \in L^2$ ,  $y \perp L^2$ . Из доказательства предыдущей теоремы следует, что  $x = z + y \in \Phi(\hat{\Pi})$  удовлетворяет неравенству

$$(Qz, z) + \frac{\|y\|^2}{(r_0^2 - \|\hat{x}_0\|^2)} - 1 \leq 0, \quad (2.89)$$

при этом выполняются соотношения

$$(\hat{\rho}_1, Q\hat{\rho}_1) = 1; \quad (\hat{\rho}_2, Q\hat{\rho}_2) = 1; \quad (2.90)$$

$$(Q\hat{\rho}_2, \hat{\rho}_1) = (\hat{\rho}_2, Q\hat{\rho}_1) = (e_1, e_2) = 0.$$

Пусть  $x = y + z \in X$ . Тогда справедливо соотношение  $\|y\|^2 = r_0^2 - \|z - \hat{x}_0\|^2$ . Далее, найдется  $\lambda \in [0, 1]$  такое, что

$$z = (1 - \lambda)\hat{\rho}_1 - \lambda\gamma\hat{\rho}_2. \quad (2.91)$$

Имеем:

$$\|z - \hat{x}_0\|^2 = \|(1 - \lambda)(\hat{\rho}_1 - \hat{x}_0) - \lambda\gamma(\hat{\rho}_2 - \hat{x}_0) - \lambda(\gamma + 1)\hat{x}_0\|^2.$$

Учитывая, что  $\|\hat{\rho}_1 - \hat{x}_0\| = \|\hat{\rho}_2 - \hat{x}_0\| = r_0$ ,  $(\hat{\rho}_2, \hat{x}_0) = 0$ , получим:

$$\begin{aligned}\|z - \hat{x}_0\|^2 &= r_0^2 [(1 - \lambda)^2 + (\lambda\gamma)^2] + \|\hat{x}_0\|^2 [1 - (1 - \lambda)^2 - (\gamma\lambda)^2] - \\ &\quad - 2\lambda(1 - \lambda)\gamma(\hat{\rho}_1, \hat{\rho}_2) - 2\lambda(1 - \lambda)(\gamma + 1)(\hat{\rho}_2, \hat{x}_0).\end{aligned}$$

Из (2.88) имеем:

$$\|y\|^2 = (r_0^2 - \|\hat{x}_0\|^2) [1 - (1 - \lambda)^2 - (\lambda\gamma)^2] + 2\lambda(1 - \lambda)\gamma(\hat{\rho}_1, \hat{\rho}_2) + \\ + 2\lambda(1 - \lambda)(\gamma + 1)(\hat{\rho}_2, \hat{x}_0). \quad (2.92)$$

Неравенство (2.89) после подстановок (2.91), (2.92) с учетом (2.90) принимает вид:

$$\frac{2\lambda(1 - \lambda)}{r_0^2 - \|\hat{x}_0\|^2} [\gamma(\hat{\rho}_1, \hat{\rho}_2) + (\gamma + 1)(\hat{\rho}_2, \hat{x}_0)] \leq 0,$$

откуда следует справедливость теоремы.

Если условие теоремы 2.22 нарушено, то в качестве  $\Phi(\Pi)$  может быть взят эллипсоид  $\Phi_*(T)$ , где  $T$  — слой минимальной толщины, содержащий  $\Pi$ .

Теперь мы можем приступить к описанию одного из вариантов метода эллипсоидов, когда локализуемые тела строятся с помощью двух отсекающих гиперплоскостей ( $m = 2$ ). Пусть имеется шар  $S_0(x_0, r_0)$ , в котором локализован минимум. Как мы указывали выше, первая отсекающая гиперплоскость получается путем параллельного сдвига гиперплоскости с нормалью  $g_f(x_0)$ , проходящей через  $x_0$ , в сторону антиградиента на отрезок длины

$$\frac{\max\{0; f(x_0) - f^s\}}{\|g_f(x_0)\|},$$

где  $f^s$  — оценка сверху для  $\min f(x)$  (например,  $f^s = f'$ ,  $f'$  — «рекорд», т. е. минимальное из значений  $f(x)$ , полученных на предыдущих итерациях; или  $f^s = f^*$ , где  $f^*$  — значение функции в точке минимума); вторая гиперплоскость может быть получена аналогичным путем на основании информации о некоторой точке  $x$ ,  $x \in S_0$ , лежащей на луче

$$x = x_0 - t g_f(x_0), \quad t > 0.$$

После проведения двух отсекающих гиперплоскостей мы можем получить следующие отсекающие тела.

1. Сегмент  $S$  (если отсечения «попутные»). Используя формулы (2.78), (2.79), мы получаем параметры описанного вокруг  $S$  эллипсоида  $\Phi^*(S)$ , переходим в точку  $x_1$  — центр  $\Phi^*(S)$  — и производим растяжение пространства в направлении, ортогональном отсекающей гиперплоскости с коэффициентом, вычисляемым по формуле (2.78). В растянутом пространстве в качестве локализующего тела мы опять получаем сферу и переходим к новому циклу вычислений.

2. Слой  $T$  (если отсечения «встречные», но пересечение отсекающих гиперплоскостей происходит вне шара  $S_0$ ). Включающий  $T$  эллипсоид  $\Phi^*(T)$  строится по формулам (2.67), (2.68), затем переходим в точку  $x_1$  — центр этого эллипсоида, — производим растяжение пространства с коэффициентом  $\alpha$  (см. (2.67)) ортогонально основаниям слоя с тем, чтобы превратить эллипсоид в шар, и переходим к новому циклу вычислений.

3.  $s$ -пирамида  $\Pi$  (отсечения «встречные» и пересечение отсекающих гиперплоскостей имеет общие с  $S_0$  точки). Проверяем регулярность пирамиды. Если она регулярна, то параметры  $\Phi(\Pi)$  вычисляются в соответствии с процедурами, описанными выше. Найдя направления главных осей в плоскости  $L^2$  и величины соответствующих полуосей и используя последовательно операцию растяжения пространства в направлении этих полуосей с соответствующими коэффициентами, превращаем  $\Phi(\Pi)$  в шар и переходим к новому циклу вычислений. Если пирамида  $\Pi$  не регулярна, то проверяем условие (2.88). Если оно выполняется, то сводим задачу построения  $\Phi(\Pi)$  к построению  $\Phi(\hat{\Pi})$ , где  $\hat{\Pi}$  — регулярная  $s$ -пирамида, в противном случае — к задаче для слоя  $T$  с соответствующими преобразованиями пространства для превращения  $\Phi(\Pi)$  в шар.

Выбор точки  $\bar{x}$  для построения второй отсекающей гиперплоскости может производиться различными способами, например назначением определенной длины шага (эвристические соображения подсказывают, что этот шаг должен быть порядка  $r_0/\sqrt{n}$ ), либо путем некоторой процедуры поиска (скажем, идти с постоянным шагом до получения «встречного» отсечения), однако это требует дополнительных затрат. В качестве точки  $\bar{x}$  можно выбирать и центр тяжести сегмента, получившегося после первого отсечения. В работах [20, 21, 98] подробно описаны конкретные алгоритмы метода эллипсоидов, основанные на двойных отсечениях. Приведем доказательство достаточно общей теоремы Гершовича о скорости сходимости различных модификаций метода эллипсоидов при решении задачи минимизации выпуклой функции  $f(x)$ ,  $x \in E^n$  [22]. Пусть  $X^*$  — область минимумов  $f(x)$  и в  $E^n$  задана сфера  $S_0 = \{x : \|x - x_0\| \leq r_0\}$ , содержащая точки из  $X^*$ . Опуская несущественные для данного рассмотрения подробности, общую схему метода эллипсоидов можно описать следующим образом.

Перед началом  $k$ -го шага ( $k = 0, 1, \dots$ ) имеем точку  $x_k \in E^n$ , особую  $n \times n$ -матрицу  $B_k$  и число  $r_k > 0$ , задающее в  $E^n$  эллипсоид  $E_k = \{x : \|B_k^{-1}(x - x_k)\| \leq r_k\}$  такой, что  $x^* \in E_k$  (матрица  $B_0$  — единичная и  $E_0 = S_0$ ).  $k$ -й шаг состоит из двух этапов.

Этап 1 (уточнение локализации). В соответствии с некоторой процедурой выбираем совокупность пар  $\{x_k^l, f_{kl}^s\}$  ( $x_k^l$  — вектор из  $E^n$ ,  $f_{kl}^s$  — число, так называемое «значение для сдвигов»),  $l = 0, 1, \dots, l_k$ . Каждая пара из этой совокупности определяет отсекающую гиперплоскость

$$\mathcal{P}(x_k^l, f_{kl}^s) = \{x : (g(x_k^l), x - x_k^l) = f_{kl}^s - f(x_k^l)\},$$

задающую в  $E^n$  полупространство вида

$$\mathcal{X}(x_k^l, f_{kl}^s) = \{x : (g(x_k^l), x - x_k^l) \leq f_{kl}^s - f(x_k^l)\} \quad (2.93)$$

такое, что  $x^* \in \mathcal{X}(x_k^l, f_{kl}^s)$ . Тело

$$Z_k = S_k \cap \left( \bigcap_{l=0}^{l_k} \mathcal{X}(x_k^l, f_{kl}^s) \right) \quad (2.94)$$



оказывается «локализирующим», так как, очевидно,  $x^* \in Z_h$ . Основное свойство, которым должна обладать указанная процедура, заключается в следующем: для тела  $Z_h$ , определенного (в силу (2.94)) выбранными в процедуре парами, существует «включающий» эллипсоид  $E(Z_h)$  ( $E(Z_h) \supset Z_h$ ) такой, что

$$q(Z_h) = \frac{v(E(Z_h))}{v(E_h)} \leq Q < 1. \quad (2.95)$$

Этап 2 (пересчет параметров). Полагаем  $E_{h+1} = E(Z_h)$ , т. е. вычисляем вектор  $x_{h+1} \in E^n$ , неособую  $n \times n$ -матрицу  $B_{h+1}$  и число  $r_{h+1} > 0$  такие, что

$$E_{h+1} = \{x : \|B_{h+1}^{-1}(x - x_{h+1})\| \leq r_{h+1}\} = E(Z_h). \quad (2.96)$$

Рассмотрим последовательность «рекордов»  $\{f_h^r\}_{h=0}^\infty$ , порождаемую при работе алгоритма эллипсоидов по правилу

$$f_h^r = \min \{f_{h-1}^r, \min \{f(x_h^l), l = 0, 1, \dots, l_h\}\}, \quad (2.97)$$

$$k = 1, 2, \dots; \quad f_0^r = f(x_0).$$

**Теорема 2.23.** Пусть  $x^* \in \text{int } S_0$  и  $\|g(x)\| \leq L$  при  $x \in S_0$ . Пусть выбор значений для сдвигов подчинен условию

$$f_{hl}^s = \min \{f_{h-1}^r, \min \{f(x_h^j), j = 0, 1, \dots, l\}\}, \quad l = 0, 1, \dots, l_h, \quad (2.98)$$

$$k = 0, 1, \dots$$

Тогда последовательности матриц  $\{B_h\}_{h=0}^\infty$ , чисел  $\{r_h\}_{h=0}^\infty$  и  $\{f_h^r\}_{h=0}^\infty$ , генерируемые алгоритмом эллипсоидов, удовлетворяют неравенствам

$$f_h^r - f(x^*) \leq 2Lr_{h+1} \sqrt{\lambda(B_{h+1}^* B_{h+1}^*)}, \quad k = 0, 1, \dots, \quad (2.99)$$

где  $\lambda(B_{h+1} B_{h+1}^*)$  — минимальное собственное значение матрицы  $H_h = B_{h+1} B_{h+1}^*$ .

**Доказательство.** Положим

$$F_h = \bigcap_{i=0}^{l_h} \mathcal{X}(x_h^i, f_{hi}^s), \quad G_h = \bigcap_{i=0}^h F_i, \quad k = 0, 1, \dots \quad (2.100)$$

Покажем, что

$$f(x) > f_k^r \text{ при } x \in G_h. \quad (2.101)$$

Пусть  $x \in F_i$ . Значит, существует такой номер  $l$ ,  $0 \leq l \leq l_i$ , что  $x \in \mathcal{X}(x_i^l, f_{il}^s)$ . Тогда с учетом (2.98) и выпуклости  $f(x)$  имеем

$$f_{il}^s - f(x_i^l) < (g(x_i^l), x - x_i^l) \leq f(x) - f(x_i^l),$$

т. е.  $f(x) > f_{il}^s$ . Но из (2.97), (2.98) очевидно  $f_{il}^s \geq f_i^r$ . Поэтому  $f(x) > f_i^r$ . Пусть теперь  $x \in G_h$ . Значит, существует такой номер  $i$ ,  $0 \leq i \leq h$ ,

что  $x \in F_i$ . Тогда  $f(x) > f'_i$  и, тем более,  $f(x) > f'_k$ , поскольку в силу (2.97)  $f'_i$  не возрастает с ростом  $i$ . Итак, (2.101) установлено. С учетом непрерывности  $f(x)$  (2.100) дает

$$f(x) \geq f'_k \text{ при } x \in \partial G_k \quad (2.102)$$

( $\partial G_k$  означает границу множества  $G_k$ ).

Положим

$$D_k = S_0 \cap G_k, \quad k = 0, 1, \dots \quad (2.103)$$

Если  $D_k = \emptyset$ , то в силу (2.103) для любого  $x \in S_0$ , а значит, и для  $x^* f(x) > f'_k$ , и (2.99) выполняется. Пусть  $D_k \neq \emptyset$ . Тогда и  $D_i \neq \emptyset$ ,  $i \leq k$ .

Покажем по индукции, что  $D_i \subset Z_i$  для любого  $i \leq k$ . Для  $i = 0$  это очевидно, так как  $D_0 = Z_0$ . Предположим, что  $D_{i-1} \subset Z_{i-1}$ . Имеем  $D_{i-1} \subset E_i$ . Отсюда

$$D_{i-1} \cap F_i \subset Z_{i-1} \cap F_i \subset E_i \cap F_i.$$

Но по определению (см. (2.94), (2.100) и (2.103))  $D_{i-1} \cap F_i = D_i$ , а  $E_i \cap F_i = Z_i$ . Следовательно,  $D_i \subset Z_i$ . Индуктивный переход завершен.

Применяя доказанное включение при  $i = k$ , имеем  $D_k \subset Z_k \subset E_{k+1}$ . Итак,  $D_k \subset E_{k+1}$ .

Воспользуемся конструкцией, предложенной в [78]. Для произвольной ограниченной замкнутой фигуры  $A \subset E^n$  под ее толщиной  $t(A)$  будем понимать радиус наибольшего евклидова шара, который можно целиком заключить в  $A$ . Очевидно, если  $A \subset B$ , то  $t(A) \leq t(B)$ . В случае, когда  $A$  — эллипсоид,  $t(A)$  — величина его минимальной полуоси. Справедливо следующее простое утверждение [78]: пусть  $A_1$  и  $A_2$  — два евклидовых шара в  $E^n$  радиусов  $r_1$  и  $r_2$  соответственно, причем  $0 < r_2 < r_1$ ; если центр шара  $A_2$  принадлежит  $A_1$ , то  $t(A_1 \cap A_2) \geq r_2/2$ .

Пусть  $t(E_{k+1}) = a$ , тогда  $t(D_k) \leq a$ . Далее, пусть  $S$  — евклидов шар радиуса  $2(a + \varepsilon)$ ,  $\varepsilon > 0$ , с центром в точке  $\xi \in D_k$ , такой, что

$$f(\xi) = \min_{x \in D_k} f(x). \quad (2.104)$$

Положим  $R = S \cap D_k$ . Покажем, что  $R \cap \Gamma \neq \emptyset$ , где  $\Gamma$  — совместная граница тел  $G_k$  и  $D_k$  (таковую они имеют лишь внутри  $S_0$ ):  $\Gamma = \partial G_k \cap \partial D_k \subset S_0$ .

Допустим противное. Тогда ясно, что  $R = S \cap S_0 \subset D_k$  и  $t(D_k) \geq t(S \cap S_0) \geq a + \varepsilon > a$  — противоречие.

Возьмем  $\eta \in R \cap \Gamma$ . По определению  $\Gamma$   $\eta \in \partial G_k$ . Следовательно, (см. (2.97))  $f(\eta) \geq f'_k$ . Далее,

$$f(\xi) \geq f(\eta) - (g(\eta), \eta - \xi) \geq f'_k - \|g(\eta)\| \|\xi - \eta\|.$$

Отсюда с учетом очевидного  $\eta \in S \cap S_0$  имеем

$$f(\xi) \geq f'_k - L \cdot 2(a + \varepsilon).$$

Переходя к пределу по  $\varepsilon \rightarrow 0$ , получим

$$f(\xi) \geq f_k^f - 2La. \quad (2.105)$$

Так как для  $x \in S_0$  либо  $x \notin G_k$ , либо  $x \in D_k$ , то (2.101), (2.104) и (2.105) совместно дают

$$f(x^*) \geq f_k^f - 2La. \quad (2.106)$$

Из неравенства

$$r_{k+1}^2 \geq \|B_{k+1}^{-1}(x - x_{k+1})\|^2 = (x - x_{k+1}, (B_{k+1}B_{k+1}^*)^{-1}(x - x_{k+1})),$$

задающего эллипсоид  $E_{k+1}$ , ясно, что  $a = r_{k+1} \sqrt{\lambda(B_{k+1}B_{k+1}^*)}$ . Подставляя это в (2.106), получим (2.99). Теорема доказана.

*С л е д с т в и е.* В условиях теоремы 2.23 справедливы неравенства

$$f_k^f - f(x^*) \leq 2Lr_0 \left( \prod_{i=0}^k q(Z_i) \right)^{1/n} \leq 2Lr_0 \cdot Q^{k/n}, \quad k = 0, 1, \dots \quad (2.107)$$

Доказательство. Имеем (см. (2.95)) для  $i = 0, 1, \dots, k$ :

$$\frac{r_{i+1}^n |\det B_{i+1}|}{r_i^n |\det B_i|} = \frac{v(E_{i+1})}{v(E_i)} = q(Z_i) \leq Q < 1.$$

Отсюда

$$\left( \frac{r_{k+1}}{r_0} \right)^n \cdot |\det B_{k+1}| = \prod_{i=0}^k q(Z_i).$$

Следовательно,

$$|\det B_{k+1}| = \left( \frac{r_0}{r_{k+1}} \right)^n \prod_{i=0}^k q(Z_i).$$

Далее,

$$\sqrt{\lambda(B_{k+1}B_{k+1}^*)} \leq |\det B_{k+1}|^{1/n} = \frac{r_0}{r_{k+1}} \left( \prod_{i=0}^k q(Z_i) \right)^{1/n}.$$

Подставляя это выражение в (2.99), получим (2.107).

Отметим, что теорема 2.23 обобщает результат, полученный в [148].

## § 6. Вычислительная сложность решения задач математического программирования и метод эллипсоидов

В последние 15 лет широкое развитие получила теория сложности переборных задач, основы которой заложены в работах С. Кука, Р. Карпа и Л. Левина. Главную роль в этой теории играет понятие полиномиальной сводимости задач.

Под массовой задачей понимается некоторый вопрос, сформулированный в форме выражения, в которое входят параметры задачи, на

который должен быть дан определенный ответ, зависящий от этих параметров. Если ответ имеет форму «да» или «нет», то соответствующая задача называется задачей распознавания. Другие задачи, как правило, сравнительно просто сводятся к последовательности задач распознавания.

Массовая задача (или просто задача)  $\Pi$  определяется следующей информацией [26]:

- 1) списком ее параметров;
- 2) формулировкой свойств, которым должен удовлетворять ответ.

При конкретных значениях параметров получаем индивидуальную задачу  $I$ . Предполагается, что значения параметров закодированы определенным способом и что для описания задачи применяется определенный язык. Таким образом, каждой массовой задаче соответствует определенная *схема кодирования*. *Входная длина* индивидуальной задачи  $I$  из  $\Pi$  может быть определена как число символов в описании задачи  $I$  по ее схеме кодирования.

Пусть имеется алгоритм, преобразующий описание любой задачи  $I$  из  $\Pi$  в ответ за конечное число операций. Тогда, имея в виду, что алгоритм выполняется некоторым автоматом, в котором каждая операция занимает определенное время, можно говорить о *временной сложности* алгоритма, понимая под этим функцию, которая каждой входной длине  $n$  ставит в соответствие минимальное время, достаточное для решения любой индивидуальной задачи длины  $n$  на этом автомате. В теории вычислительной сложности алгоритмов принято считать, что, несмотря на разнообразие ЭВМ, отношение временных сложностей различных алгоритмов не очень существенно зависит от вида устройства, на котором реализуется алгоритм. При этом можно абстрагироваться от конкретных устройств и рассматривать число тактов работы некоторой идеализированной ЭВМ (например, машины Тьюринга).

Будем говорить, что алгоритм решения задачи  $\Pi$  *полиномиальный*, если его временная сложность  $t_{\Pi}(n) = O(p(n))$ , где  $p(n)$  — некоторый полином, т. е. существует константа  $c > 0$  такая, что  $t_{\Pi}(n) \leq c|p(n)|$ . В противном случае алгоритм называется *экспоненциальным*. Мы будем говорить, что задача *полиномиально разрешима*, или принадлежит классу  $P$ , если существует полиномиальный алгоритм ее решения. Такой алгоритм называется также *эффективным*, хотя это понятие в некоторых случаях имеет весьма отдаленное отношение к тому, что мы понимаем под практической эффективностью (например, если  $t(n) \leq 10^{20}n^2$  либо  $t(n) \leq n^{10}$ ), и дает лишь принципиальный ориентир. Доказательство полиномиальной разрешимости некоторой задачи служит стимулом для поиска практически эффективных алгоритмов.

Орудием доказательства полиномиальной разрешимости (или не разрешимости) отдельных задач служит важное понятие *полиномиальной сводимости*. Говорят, что задача распознавания  $\Pi_1$  сводится к задаче распознавания  $\Pi_2$ , если от схемы кодирования задачи  $\Pi_1$  можно перейти к схеме кодирования задачи  $\Pi_2$  для каждой индивидуальной задачи из  $\Pi_1$  таким образом, что ответы будут совпадать.

Задача  $\Pi_1$  полиномиально сводима к задаче  $\Pi_2$ , если перекодировка может быть произведена за полиномиальное время. Если задача  $\Pi_1$  сводима полиномиально к задаче  $\Pi_2$ , а задача  $\Pi_2$  сводима полиномиально к задаче  $\Pi_1$ , то говорят, что  $\Pi_1$  и  $\Pi_2$  *полиномиально эквивалентны*.

Большое значение имеет класс  $NP$  задач, разрешимых за полиномиальное время на так называемых недетерминированных машинах Тьюринга. В нестрогих терминах речь идет не о решении задачи, а о возможности *проверки правильности ответа* за полиномиальное время. Более точно: для каждой индивидуальной задачи распознавания из  $\Pi \in NP$  некий «оракул» может выдать в случае положительного ответа («да») соответствующим образом закодированную дополнительную информацию, длина которой ограничена полиномом от первоначальной длины входа, после чего «расширенная» задача перерабатывается полиномиальным алгоритмом, подтверждающим ответ «да».

Принадлежность к классу  $NP$  не симметрична по отношению к замене в задаче распознавания вопроса на противоположный. Например, задача: «Существует ли в данном графе гамильтонов цикл?» принадлежит к классу  $NP$  («оракул» в случае существования такого цикла выдает последовательность ребер, задающих такой цикл, при этом алгоритм проверки тривиален). Проверить ответ на вопрос: «Правильно ли, что в данном графе нет гамильтонового цикла?» такого рода оракул уже не поможет; для проверки правильности ответа в данной задаче нужно рассмотреть, вообще говоря, экспоненциальное от числа вершин графа число циклов.

Отметим, что приведенные выше определения недостаточно строги с точки зрения требований математической формализации, они приведены здесь лишь для более содержательного восприятия неискушенным в данном вопросе читателем последующего материала. Подробное изложение теории алгоритмической сложности задач читатель найдет в монографии М. Гэри, Д. Джонсона [26].

Из определения класса  $NP$  вытекает, что  $P \subseteq NP$ . Вопрос о совпадении классов  $P$  и  $NP$  является принципиальной теоретической задачей теории сложности. Гипотеза  $P \neq NP$  считается весьма достоверной. Дело в том, что в классе  $NP$  найдено большое число содержательно формулируемых так называемых универсальных переборных задач, к любой из которых полиномиально сводима любая задача из класса  $NP$ . Универсальные задачи или, как говорят,  $NP$ -полные задачи являются, так сказать, первыми кандидатами на принадлежность к множеству  $NP \setminus P$ , если оно не пусто. Среди  $NP$ -полных много просто формулируемых комбинаторных задач, имеющих большое прикладное значение (список более 300 таких задач с комментариями содержится в упомянутой выше монографии [26]). В их число входят задачи, возникающие в математической логике, теории автоматов и алгоритмов, теории игр, теории графов, математическом программировании, теории расписаний и других важных областях. Формулировки большей части этих задач давно известны, попытки построения эффективных алгоритмов их решения предпринимались неоднократно, однако ни для одной из них полиномиальный алго-

ритм не был построен, что делает крайне маловероятным существование такого алгоритма, а значит, и совпадение классов  $P$  и  $NP$ .

Задачу, к которой полиномиально сводима  $NP$ -полная задача, будем называть  $NP$ -трудной. Она не обязательно принадлежит классу  $NP$ , а может принадлежать классу задач более высокой сложности в рассматриваемой иерархии.

Для каждой массовой задачи  $\Pi$  принципиальное значение имеет вопрос ее принадлежности классу  $P$  или классу  $NP$ -трудных задач. В последнем случае поиск для такой задачи полиномиального алгоритма является практически безнадежным делом. Особое внимание исследователей привлекают задачи, для которых не решен вопрос о принадлежности классу  $P$  или  $NP$ -полных задач. Среди таких проблем находилась длительное время и общая задача линейного программирования. В работах Кли и Минти [102], Заде [180], Диница и Карзанова [3] для различных вариантов симплекс-метода были построены примеры последовательностей задач, для которых число итераций симплекс-метода растет экспоненциально с ростом размерности. Лишь в 1979 г. Л. Г. Хачияну удалось построить первый полиномиальный алгоритм для решения задач линейного программирования с целыми (рациональными) коэффициентами [77, 78], при этом основную конструктивную роль сыграл метод эллипсоидов.

Как известно, задача линейного программирования может быть полиномиально сведена к решению системы неравенств. Рассмотрим систему линейных неравенств с целыми коэффициентами вида

$$Ax \leq b \quad \text{или} \quad \sum_{j=1}^n a_{ij}x_j - b_i \leq 0, \quad i = 1, \dots, m, \quad (2.108)$$

где  $A = \{a_{ij}\}$  — матрица  $m \times n$  с целыми элементами  $a_{ij}$  ( $i = 1, \dots, m; j = 1, \dots, n$ ),  $b$  —  $m$ -мерный вектор. Легко показать, что если система (2.108) имеет решение, то она имеет и «опорное» решение, т. е. решение невырожденной системы линейных уравнений, матрица которой получается путем отбрасывания определенных строк и столбцов матрицы  $A$ , а правая часть в оставшихся строках сохраняется без изменения (т. е. отбрасываются несущественные ограничения и некоторые компоненты вектора  $x$  приравняются к 0).

Длина кода задачи (2.108) характеризуется следующим выражением:

$$L = \left[ \sum_{i=1}^m \sum_{j=1}^n \log_2(|a_{ij}| + 1) + \sum_{i=1}^m \log_2(|b_i| + 1) + \log_2 nm \right] + 1. \quad (2.109)$$

Ненулевые компоненты опорного решения могут быть записаны по теореме Крамера в виде отношения

$$x_i = \det \Delta_i / \det \Delta, \quad i \in \{1, \dots, n\},$$

где  $\Delta_i$  — некоторый минор расширенной матрицы  $[A, b]$ ,  $\Delta$  — минор матрицы  $A$ . Оценив  $|\Delta_i|$  и учитывая, что  $|\Delta| \geq 1$ , получаем сле-

дующую оценку сверху для радиуса сферы с центром в начале координат, внутри которой находится решение системы (если оно существует):  $R \leq 2^L$ .

Рассмотрим функцию

$$\varphi(x) = \max_i \left( \sum_{j=1}^n a_{ij}x_j - b_i \right).$$

Если  $\min \varphi(x) \leq 0$ , то система (2.108) имеет решение, если  $\min \varphi(x) > 0$ , то система не имеет решения. Оказывается, что если задача (2.108) не имеет решения, то существует «зазор» — число  $\alpha = \alpha(L) > 0$  такое, что выполняется неравенство  $\min \varphi(x) > \alpha$ .

Этот «зазор» позволяет применить для решения системы (2.108) приближенный алгоритм: достаточно найти минимум  $\varphi(x)$  с точностью до  $\alpha/2$  по функционалу. Если при этом найденное значение функционала окажется не превосходящим  $\alpha/2$ , то система (2.108) разрешима, в противном случае система (2.108) противоречива. Так как необходимая точность решения задачи оценивается числом  $2^{-L}$ , экспоненциально зависящим от длины кода задачи, то для получения полиномиального числа итераций необходим алгоритм, сходящийся со скоростью геометрической прогрессии, знаменатель которой не зависит от исходной информации. Таким условиям как раз и удовлетворяет метод эллипсоидов.

Зная длину кода системы линейных неравенств (2.108) и используя оценки  $R \leq 2^L$  и  $\alpha \geq 2 \cdot 2^{-L}$ , легко получаем, применяя теорему 2.20, число итераций, которое гарантирует получение решения системы (2.108) с точностью до  $\alpha/2$ :

$$N \leq 6n^2L. \quad (2.110)$$

Общее число арифметических операций не превышает при этом  $O(mn^3L)$ . Эти оценки получены для точной арифметики, т. е. когда вычисления производятся с неограниченным числом разрядов.

В работах [78, 114] эти результаты модифицируются для «реальной» арифметики, когда вычисления производятся с ограниченным числом разрядов (порядка длины кода  $L$  задачи), при этом полученные оценки лишь слегка изменяются, но порядок их остается прежним:  $O(mn^3L)$  при  $n \leq m$ .

Допустим, что, используя метод эллипсоидов, мы показали, что система вида (2.108) совместна. Теперь уже несложно построить полиномиальный алгоритм нахождения вектора решения. Для этого мы сначала последовательно, начиная с  $x_1$ , проверяем, каким группам переменных можно придать значение 0, чтобы система при этом оставалась совместной. При подстановке вместо переменного  $x_k$  значения 0 длина кода редуцированной задачи не превышает длины кода первоначальной задачи, проверка совместности осуществляется с помощью описанного выше алгоритма эллипсоидов. Такую проверку нужно осуществить не более  $n$  раз. В редуцированной задаче

отбросим несущественные ограничения. Это можно сделать, заменяя последовательно неравенства равенствами и проверяя полученную систему на несовместность. В результате получим неособенную систему уравнений с целыми коэффициентами, решение которой при добавлении нулей на местах тех переменных, которые были отброшены в процессе редуцирования, дает точку, являющуюся решением системы неравенств (2.108). При этом получаем следующую оценку для числа операций по решению системы неравенств (2.108):  $O(m^2 n^4 \log Hn)$  при  $m \geq n$ ;  $O(m^4 n^2 \log Hm)$  при  $m \leq n$ , где  $H$  — максимальный модуль коэффициентов задачи.

Задача линейного программирования может быть различными способами сведена к решению системы неравенств. При этом сложность решения задачи линейного программирования задается следующей оценкой:  $mn^4 O(n \log^2 Hn + m \log Hn)$  при  $m \geq n$ ,  $nm^4 O(m \log^2 Hm + n \log Hm)$  при  $m \leq n$ .

Метод эллипсоидов оказался весьма полезным при исследовании вычислительной сложности большого числа комбинаторных задач и задач нелинейного программирования.

Типичная задача комбинаторной оптимизации сводится к следующей:  $\min_{x \in S} (c, x)$ , где  $x \in E^n$ ,  $S$  — некоторое конечное множество, причем  $|S|$ , как правило, экспоненциально зависит от  $n$ , и поэтому  $S$  задается не перечислением своих элементов, а указанием свойств (например, описывается множество линейных неравенств, которым удовлетворяют точки  $S$ , плюс условие булевости или целочисленности координат). Рассмотрим многогранник  $P(S)$ , являющийся выпуклой оболочкой множества  $S$ . Пусть размерность его  $\dim P(S) = k \leq n$ . Многогранник  $P(S)$  может быть задан указанием того линейного многообразия  $L$  размерности  $k$ , в котором он находится (если  $k < n$ ), и перечислением его собственных граней размерности  $k - 1$ . Для этого нужно задать систему  $n - k$  независимых линейных уравнений, определяющих  $L$  при  $k < n$ , и систему линейных неравенств  $l_\alpha(x) \leq 0$ , каждое из которых «вырезает» в  $L$  множество,  $(k-1)$ -мерная граница которого содержит  $(k-1)$ -мерную грань  $P(S)$ . В общем случае описать эту систему неравенств очень трудно, но даже в тех немногих случаях, когда она описывается компактно, число неравенств и соответствующих  $(k-1)$ -мерных граней может оказаться огромным — оно экспоненциально зависит от  $n$ . Следовательно, длина кода соответствующей задачи линейного программирования может оказаться экспоненциально зависящей от  $n$ , и тогда непосредственное применение метода эллипсоидов к полностью выписанной задаче будет неэффективным.

Однако метод эллипсоидов отличается тем, что для его применения не обязательно выписывать в явном виде все ограничения задачи, достаточно на каждом шаге либо находить одно из ограничений, которое не выполняется, либо показать, что текущая точка является допустимой (в первом случае отсекающая плоскость строится с использованием градиента к ограничению, во втором случае используется градиент к целевой функции). В некоторых случаях



задача определения допустимости данной точки оказывается полиномиально разрешимой. Тогда, как показано в работе [141], и первоначальная задача полиномиально разрешима.

Рассмотрим несколько примеров.

1. **Задача о минимальном разрезе.** Пусть задан связный граф  $G(V, E)$  со множеством вершин  $V$  и ребер  $E$ . Каждому ребру  $e \in E$  соответствует целочисленный вес  $c(e) > 0$ . Выделены две вершины  $v_1 \in V, v_2 \in V, v_1 \neq v_2$ . *Разрезом* называется подмножество ребер  $R \in E$  таких, что их удаление разбивает граф по крайней мере на две несвязные компоненты, причем  $v_1$  и  $v_2$  находятся в разных компонентах. Каждому разрезу  $R$  сопоставим число  $c(R)$ :

$$c(R) = \sum_{e \in R} c(e).$$

Задача состоит в нахождении минимального разреза  $R^*$ , т. е. такого разреза, на котором достигается  $\min_R c(R)$ . Задача нахождения минимального разреза эквивалентна следующей задаче булева линейного программирования: найти

$$\min \sum_{e \in E} c(e) x(e)$$

при ограничениях

$$\sum_{e \in \pi} x(e) \geq 1, \quad \pi \in \Pi; \quad x(e) \in \{0, 1\}, \quad e \in E,$$

где  $\Pi$  — множество путей, соединяющих  $v_1$  и  $v_2$ . Как показано в [3], эта задача эквивалентна следующей задаче линейного программирования: найти

$$\min \sum_{e \in E} c(e) x(e) \tag{2.111}$$

при ограничениях

$$\sum_{e \in \pi} x(e) \geq 1, \quad \pi \in \Pi; \tag{2.112}$$

$$0 \leq x(e) \leq 1. \tag{2.113}$$

Применим к задаче (2.111) — (2.113) метод эллипсоидов. Пусть мы находимся в точке  $\bar{x} = \{\bar{x}(e)\}_{e \in E}$ . Проверка ограничений (2.113) выполняется тривиальным способом. Если эти ограничения выполнены, переходим к проверке ограничений (2.112). Число путей  $\pi \in \Pi$  может экспоненциально расти с ростом числа вершин графа. Однако, как легко увидеть, проверка выполнения ограничений (2.112) в точке  $\bar{x}$  сводится к решению задачи о кратчайшем пути между вершинами  $v_1$  и  $v_2$ , когда длины ребер  $e \in E$  определяется числами  $\bar{x}(e)$ .

Если  $\min_{\pi} \sum_{e \in \pi} \bar{x}(e) \geq 1$ , то все ограничения (2.112) выполняются.

Если  $\min_{\pi} \sum_{e \in \pi} \bar{x}(e) < 1$ , то найдется путь  $\pi_{\bar{x}} \in \Pi$  такой, что  $\sum_{e \in \pi_{\bar{x}}} \bar{x}(e) < 1$

(например, кратчайший путь). Это и будет искомое невыполняющееся ограничение, определив которое, можно сделать очередной шаг отсечения в методе эллипсоидов. Как известно, задача нахождения кратчайшего пути в графе полиномиально разрешима ([79, 119]). Таким образом, один шаг метода эллипсоидов при решении задачи (2.111) — (2.113) требует полиномиального числа операций.

Остается оценить число шагов в методе эллипсоидов. Учитывая ограничение (2.113), начальный радиус  $R_0$  сферы, в которой локализован минимум, можно выбрать равным  $\sqrt{|E|}/2$ , а центр в точке  $x_0$ , где  $x_0(e) = 1/2$ ,  $e \in E$ . Далее, используя прием дихотомии и учитывая, что оптимальное значение  $c^*$  целевой функции задачи (2.111) — (2.113) является целым числом, задачу (2.111) — (2.113) можно свести к решению последовательности систем неравенств (2.112), (2.113) в совокупности с неравенством вида

$$\sum_{e \in E} c(e) x(e) \leq K, \quad (2.114)$$

где  $K$  — целое число,  $1 \leq K \leq \sum_{e \in E} c(e) = s$ .

Так как матрица системы неравенств (2.112), (2.113) вполне унимодулярна (т. е. определитель любой квадратной подматрицы принимает одно из значений  $-1, 1$  и  $0$ ), определитель любой квадратной подматрицы системы (2.111), (2.113) и (2.114) ограничен по модулю значением  $s = \sum_{e \in E} c(e)$ . Теперь мы можем использовать оценку (2.110) для числа шагов в методе эллипсоидов, требуемых для проверки совместности данной системы неравенств. В результате получим существование полиномиального алгоритма для задачи о разрезе.

**2. Определение оптимального корневого связывающего дерева в направленном графе.** Пусть  $D(V, A)$  — направленный граф (орграф) и пусть  $r \in V$  — некоторая выделенная вершина, называемая корнем. Связывающим направленным корневым деревом называется минимальный частичный граф, в котором существует путь из корня в любую вершину. Соответствующее множество дуг  $A' \subset A$  называется ветвлением. Корневым сечением называется совокупность дуг  $A'$ , состоящих из  $\bigcup_{v \in V'} \delta^{-1}(v)$ , где  $V'$  — непустое подмножество вершин, не содержащих  $r$ ,  $\delta^{-1}(v)$  — совокупность дуг, входящих в  $V$ . Из теоремы Эдмондса [122] следует, что выпуклая оболочка векторов инцидентности подмножеств дуг, содержащих ветвление как подмножество, является многогранником, определенным следующей системой линейных неравенств:

$$0 \leq x(a) \leq 1, \quad a \in A, \quad (2.115)$$

$$\sum_{a \in A'} x(a) \geq 1, \quad (2.116)$$

где  $A'$  — множество дуг, образующих корневое сечение. Каждой

дуге  $a$  сопоставлен вес  $c(a)$ . Требуется найти ветвление минимального общего веса, т. е. найти

$$\min \sum_{a \in A} c(a) x(a) \quad (2.117)$$

при ограничениях (2.115), (2.116).

Для применения метода эллипсоидов к указанной задаче нужно уметь проверять (2.116) при заданном  $x = \{x(a)\}$  за полиномиальное время. Легко видеть, что эта задача сводится к нахождению разреза минимального веса между вершиной  $r$  и всеми остальными вершинами (в качестве весов дуг выступает  $\{x(a)\}$ ). Как отмечалось в предыдущем примере, эта задача полиномиально разрешима (см. также [3]). Таким образом, один шаг метода эллипсоидов при решении задачи (2.115) — (2.117) требует полиномиального времени счета относительно числа вершин графа  $n$  (предполагается, что  $c(a)$ ,  $a \in A$ , выражаются рациональными числами с фиксированным числом разрядов в числителе и знаменателе). Можно показать, что число шагов метода эллипсоидов, гарантирующих решение задачи (2.115) — (2.117), также является полиномом относительно  $n$  и  $\log \sum_{a \in A} c(a)$ . Отсюда следует полиномиальная разрешимость задачи (2.115) — (2.117).

**3. Задача о максимальном взвешенном совершенном паросочетании графа.** Пусть задан граф  $G(V, E)$  и на множестве его ребер  $E$  определена функция  $c(e)$ ,  $e \in E$ . Совершенным паросочетанием называется такая совокупность ребер  $P$ , что каждой вершине инцидентно одно и только одно ребро из  $P$ . Множество всех совершенных паросочетаний графа  $G$  обозначим через  $K$ .

Введем булевы переменные  $x(e)$ , которые принимают значение 1, если  $e \in P$ , и 0 — в противном случае. Как показал Эдмондс [123], выпуклое замыкание векторов  $x = \{x(e)\}_{e \in E}$  представляет собой многогранник, который можно описать следующим образом:

$$x(e) \geq 0, \quad e \in E, \quad (2.118)$$

$$\sum_{e \in I(v)} x(e) = 1, \quad v \in V, \quad (2.119)$$

$$\sum_{e \in E'} x(e) \geq 1, \quad (2.120)$$

где  $I(v)$  — множество ребер, инцидентных вершине  $v$ ,  $E'$  — произвольный нечетный разрез, т. е. множество ребер, которые имеют ровно одну конечную вершину в некотором подмножестве  $V'$  с нечетным числом вершин.

Число различных нечетных разрезов экспоненциально растет с увеличением числа вершин  $|V|$ . Поэтому проверка справедливости ограничений (2.120) для заданного  $\{x(e)\}_{e \in E}$  — нетривиальная задача. В работе [164] описан алгоритм решения этой задачи, близкий к алгоритму нахождения минимального разреза графа и требующий

полиномиального по  $|E|$  числа операций. Отсюда следует, что один шаг метода эллипсоидов при отыскании  $\max_{e \in E} \sum c(e) x(e)$  при ограничениях (2.118) — (2.120) требует полиномиального числа операций. Далее, число шагов в методе эллипсоидов также оценивается полиномом относительно  $|E|$  и  $\log \sum_{e \in E} c(e)$ , откуда следует известный результат о полиномиальной разрешимости задачи о максимальном взвешенном совершенном паросочетании.

Такого рода примеры можно продолжить. Как показано в работе [141], почти все известные результаты о полиномиальной разрешимости экстремальных комбинаторных задач могут быть получены аналогичным путем: дается описание задачи в терминах линейного программирования, вообще говоря, с экспоненциальным числом ограничений; рассматривается вспомогательная задача проверки выполнимости всех ограничений и нахождения того ограничения, которое не выполняется (если результат проверки отрицательный); если эта задача требует полиномиального числа операций, то с использованием метода эллипсоидов показывается, что и первоначальная задача полиномиально разрешима.

Наиболее общие результаты по анализу сложности решения экстремальных комбинаторных задач с использованием метода эллипсоидов получены в работе [141].

Пусть  $\mathcal{K}$  — непустое выпуклое компактное множество в  $E^n$ . Рассматриваются две задачи, относящиеся к  $\mathcal{K}$ :

1) сильная оптимизационная задача: дан вектор  $c \in E^n$ ; найти вектор  $x \in \mathcal{K}$ , на котором достигается максимум  $(c, x)$  на  $\mathcal{K}$ ;

2) сильная задача отделения: дан вектор  $y \in E^n$ , определить, принадлежит ли  $y$   $\mathcal{K}$ ; и если нет, то найти гиперплоскость, отделяющую  $y$  от  $\mathcal{K}$ , т. е. найти вектор  $c \in E^n$  такой, что  $(c, y) > (c, x) \forall x \in \mathcal{K}$ .

Если не накладывать никаких ограничений на множество  $\mathcal{K}$ , может оказаться, что его параметры (например, координаты вершин) выражаются иррациональными числами, поэтому в строгом смысле с точки зрения теории сложности поставленные задачи некорректны. В общем случае нужно рассмотреть их приближенные модификации.

**1. Оптимизационная задача (в слабом смысле).** Дан вектор  $c \in E^n$  и число  $\varepsilon > 0$ ; найти вектор  $y \in E^n$  такой, что  $d(y, \mathcal{K}) \leq \varepsilon$  ( $d(y, \mathcal{K})$  — евклидово расстояние между  $y$  и  $\mathcal{K}$ ) и для любого  $x \in \mathcal{K}$   $(c, x) \leq (c, y) + \varepsilon$ .

**2. Задача отделения (в слабом смысле).** Дан вектор  $y \in E^n$  и число  $\varepsilon > 0$ ; нужно либо убедиться в том, что  $d(y, \mathcal{K}) \leq \varepsilon$ , либо найти  $c \in E^n$  такой, что  $\|c\| \geq 1$  и для любого  $x \in \mathcal{K}$   $(c, x) \leq (c, y) + \varepsilon$ .

Класс рассматриваемых выпуклых тел следует ограничить. Мы будем рассматривать класс  $\mathcal{K}$ , представители которого  $K \in \mathcal{K}$  имеют следующий вид:  $K = K(n, a_0, r, R)$ , где  $n$  — размерность  $K$ ,  $a_0$  — внутренняя точка такая, что  $S(a_0, r) \subset K \subset S(a_0, R)$  (здесь  $S(a, p)$  —  $n$ -мерный шар с центром в точке  $a$  и радиусом  $p$ ).

Каждая оптимизационная задача (в слабом смысле), относящаяся к телу  $K(n, a_0, r, R)$ , при заданном  $\epsilon$  характеризуется входом, длина которого не меньше  $n + |\log r| + |\log R| + |\log \epsilon|$  (в этом выражении не учтены длины описания тела  $K$  и вектора  $c$ ). Класс  $\mathcal{H}$  называется полиномиальным, если время решения соответствующей оптимизационной задачи ограничено полиномом от длины входа. Для того чтобы получить линейную зависимость времени решения от  $|\log \epsilon|$ , следует применять алгоритмы линейного программирования, сходящиеся со скоростью геометрической прогрессии. Из известных алгоритмов только методы эллипсоидов, симплексных отсечений [2, 102] и Кармаркара обладают указанным свойством.

В [141] приводится такая модификация метода эллипсоидов, которая позволяет учитывать ошибки округления при использовании вычислений с ограниченной разрядностью и получать решение задачи оптимизации (в слабом смысле), используя процедуру SEP решения задачи отделения (в слабом смысле).

Пусть  $K \in E^n$  — компактное выпуклое множество,  $S(a_0, r) \subseteq K \subseteq S(a_0, R)$ ,  $(c, x)$  — линейная целевая функция,  $\epsilon > 0$ . Без потери общности можно предположить, что  $\epsilon < r$ ,  $\|c\| \geq 1$ ,  $n \geq 2$ . Процедура SEP по данному  $y \in E^n$  и  $\delta > 0$  либо заключает, что  $y$  отстоит от  $K$  на расстоянии, не превышающем  $\delta$ , либо вырабатывает вектор  $d$  такой, что  $\|d\| \geq 1$  и  $\max\{(d, x) : x \in K\} \leq (d, y) + \delta$ . Для решения оптимизационной задачи относительно  $K$  (в слабом смысле) предлагается следующий алгоритм.

Положим

$$N = 4n^2 \lceil \log(2R^2 \|c\|/r\epsilon) \rceil, \quad (2.121)$$

$$S = R^2 \cdot 4^{-N} / 300n, \quad (2.122)$$

$$p = 5n. \quad (2.123)$$

Далее вырабатываем последовательность  $x_0, x_1, \dots$  векторов и последовательность  $A_0, A_1, \dots$  положительно определенных матриц следующим образом.

Начинаем с  $x_0 = a_0$ ,  $A_0 = R^2 I_n$ . Пусть уже выполнено  $k$  шагов и определены  $x_k$  и  $A_k$ . Обращаемся к процедуре SEP при  $y = x_k$  и  $\delta$ , определенном согласно (2.122). Если оказывается, что  $x_k$  отстоит от  $K$  на расстоянии, не превышающем  $\delta$ , то  $k$  назовем *допустимым индексом* и положим  $g = c$ . Если же SEP выдает вектор  $d$ ,  $\|d\| \geq 1$ , такой, что  $\max\{(d, x) : x \in K\} \leq (d, x_k) + \delta$ , то  $k$  назовем *недопустимым индексом* и положим  $g = -d$ . Далее определяются:

$$b_k = A_k g / \sqrt{(A_k g, g)}, \quad (2.124)$$

$$x_k^* = x_k + b_k / (n + 1), \quad (2.125)$$

$$A_k^* = \frac{2n^2 + 3}{2n^2} \left( A_k - \frac{2}{n+1} b_k b_k^T \right). \quad (2.126)$$

Эллипсоид

$$E_k^* = \{x \in E^n : (A_k^{*-1} (x - x_k^*), x - x_k^*) \leq 1\}$$

получается из эллипсоида

$$E_k = \{x \in E^n : (A_k^{-1}(x - x_k), x - x_k) \leq 1\}$$

приближенно по формулам обычного метода эллипсоидов; отличие состоит в том, что для компенсации ошибок округления вместо множителя  $n^2/(n^2 - 1)$  в (2.126) используется  $(2n^2 + 3)/2n^2$ .

Пусть  $x_{k+1} \approx x_k^*$ ,  $A_{k+1} \approx A_k^*$ , где знак « $\approx$ » означает, что левая часть получается округлением компонент правой части с точностью до  $p$  двоичных разрядов после запятой.

Справедливы следующие результаты [141]:

1) матрицы  $A_0, A_1, \dots, A_N$  положительно определены, при этом

$$\|x_k\| \leq \|a_0\| + R \cdot 2^k, \quad \|A_k\| \leq R^2 \cdot 2^k, \quad \|A_k^{-1}\| \leq R^{-2} 4^k;$$

2) отношение  $n$ -мерных объемов

$$\frac{v(E_{k+1})}{v(E_k)} < e^{-1/(4n)}; \quad (2.127)$$

3) пусть  $\xi_k = \max_{0 \leq j \leq k} \{(c, x_j) : x_j - \text{допустимое}\}$  и  $K_k = K \cap \{x : (c, x) \geq \xi_k\}$ ; тогда

$$E_k \supset K_k. \quad (2.128)$$

Из этих результатов легко вытекает основная теорема.

**Теорема 2.24.** Пусть  $j$  — допустимый индекс, для которого

$$(c, x_j) = \max_{0 \leq k < N} \{(c, x_k) : x_k - \text{допустимое}\}.$$

Тогда  $(c, x_j) \geq \max\{(c, x) : x \in K\} - \varepsilon$ .

**Доказательство.** Из (2.127) и (2.128) получаем, что

$$v(K_N) \leq v(E_N) \leq e^{-N/(4n)} v(E_0) = e^{-N/(4n)} R^n v_n,$$

где  $v_n$  — объем  $n$ -мерного шара единичного радиуса. Пусть  $y$  — точка максимума  $(c, x)$ ,  $x \in K$  и  $(c, y) = \xi$ . Рассмотрим конус с вершиной в точке  $y$  над  $(n - 1)$ -мерным шаром радиуса  $r$  с центром в точке  $x_0$ , лежащим в гиперплоскости  $(c, x) = (c, x_0)$ . Часть этого конуса в полупространстве  $(c, x) \geq (c, x_j)$  содержится в  $K$ . Объем этой части удовлетворяет неравенству

$$\frac{v_{n-1} r^{n-1} (\xi - (c, x_0))}{n \|c\|} \frac{(\xi - (c, x_j))^n}{(\xi - (c, x_0))^n} \leq v(K_N) \leq e^{-N/(4n)} R^n v_n.$$

Отсюда

$$\xi - (c, x_j) \leq e^{-N/(4n^2)} R \left( \frac{\xi - (c, x_0)}{r} \right)^{(n-1)/n} \left( \frac{nv_n}{v_{n-1}} \right)^{1/n} \|c\|^{1/n}.$$

Учитывая, что

$$|\xi - (c, x_0)| = |(c, y - x_0)| \leq \|c\| \|y - x_0\| \leq R \|c\|,$$

получаем

$$\xi -- (c, x_j) \leq 2e^{-N/(4n^2)} (R^2/r) \|c\| \leq \varepsilon,$$

что и требовалось доказать.

Таким образом, показано, что слабая задача  $\varepsilon$ -оптимизации на классе тел  $K$ , обладающих свойствами

$$S(a_0, r) \subseteq K \subseteq S(a_0, R),$$

требует не более  $N = 4n^2 \lceil \log(2R^2 \|c\|/r\varepsilon) \rceil$  итераций метода, близкого к методу эллипсоидов, с использованием процедуры SEP. Докажем теперь важную теорему.

**Т е о р е м а 2.25.** Пусть  $\mathcal{K}$  — класс выпуклых тел вида  $K(n, a_0, r, R)$ . Тогда полиномиальный алгоритм решения задачи отделения (в слабом смысле) для членов  $\mathcal{K}$  существует тогда и только тогда, когда существует алгоритм решения задачи оптимизации (в слабом смысле) для задач из  $\mathcal{K}$ .

**Д о к а з а т е л ь с т в о.** Из того, что подпрограмма SEP имеет полиномиальную сложность, следует, что результат этой подпрограммы, поступающий на каждой итерации на вход алгоритма  $\varepsilon$ -оптимизации, имеет полиномиальную длину. Анализ приведенного выше алгоритма  $\varepsilon$ -оптимизации показывает, что вычисление  $x_{k+1}$  и  $A_{k+1}$  требует полиномиального времени, а число итераций тоже полиномиально по отношению к  $n$ ,  $|\log R|$ ,  $|\log r|$  и  $|\log \varepsilon|$ . Таким образом, из полиномиальной сложности подпрограммы SEP вытекает полиномиальная сложность алгоритма оптимизации OPT.

Рассмотрим обратный переход. Без потери общности положим  $a_0 = 0$ . Пусть  $K^*$  — полярка  $K$ , определяемая следующим образом:

$$K^* = \{u : (u, x) \leq 1 \text{ для всех } x \in K\}.$$

Легко видеть, что  $K^*$  — выпуклое множество,  $(K^*)^* = K$  и

$$S(0, 1/R) \subset K^* \subset S(0, 1/r).$$

Если  $\mathcal{K}$  — класс выпуклых тел с  $a_0 = 0$ , то обозначим  $\mathcal{K}^* = \{K^* : K \in \mathcal{K}\}$ . Покажем, что слабая проблема отделения для класса  $\mathcal{K}^*$  выпуклых тел с  $a_0 = 0$  полиномиально разрешима тогда и только тогда, когда слабая оптимизационная задача полиномиально разрешима для класса  $\mathcal{K}$ .

Пусть  $K^* \in \mathcal{K}^*$ ,  $v \in E^n$  и  $\varepsilon > 0$ . Применяя OPT для  $K$  с целевой функцией  $(v, x)$  и погрешностью  $\varepsilon r$ , мы получим вектор  $z \in E^n$  такой, что  $d(z, K) \leq \varepsilon r$  и

$$(v, z) \geq \max \{(v, x) : x \in K\} - \varepsilon r.$$

Теперь, если  $(v, z) \leq 1$ , то  $(v, x) \leq 1 + \varepsilon r$ ,  $x \in K$ , и значит,  $v_0 = v/(1 + \varepsilon r) \in K^*$ . Поэтому  $\|v_0\| \leq 1/r$  и  $d(v, K^*) \leq \|v - v_0\| \leq \varepsilon$ .

С другой стороны, если  $(v, z) > 1$ , то  $z$  — решение задачи отделения для  $K^*$ . В самом деле, пусть  $z_0 \in K$  и  $\|z - z_0\| \leq \varepsilon r$ . Тогда для любого  $u \in K^*$  имеем  $\|u\| \leq 1/r$  и  $(z, u) = (z - z_0, u) + (z_0, u) \leq \|u\| \|z -$

$\|z_0\| + 1 \leq \varepsilon + (z, v)$ , откуда следует, что  $z$  — решение задачи отделения для  $K^*$ .

Далее, из первой части доказательства теоремы следует, что полиномиальная разрешимость задачи отделимости для класса  $\mathcal{H}^*$  влечет полиномиальную разрешимость задачи оптимизации для этого класса. Применяя еще раз только что доказанное утверждение, с учетом равенства  $(\mathcal{H}^*)^* = \mathcal{H}$  получим разрешимость проблемы отделения для  $\mathcal{H}$  за полиномиальное время.

Доказательство теоремы 2.25 об эквивалентности полиномиальной разрешимости задачи отделения и оптимизации для класса тел  $\mathcal{H}$  закончено.

Класс  $\mathcal{H}$ , для представителей которого существует полиномиальный алгоритм решения оптимизационной задачи (или задачи отделения), называется *разрешимым*.

Пусть  $\mathcal{K}$  и  $\mathcal{L}$  — два класса выпуклых тел; определим класс  $\mathcal{K} \wedge \mathcal{L}$  следующим образом:

$$\mathcal{K} \wedge \mathcal{L} = \{K \cap L : K \in \mathcal{K}, L \in \mathcal{L}, \dim K = \dim L, a_0(K) = a_0(L)\}.$$

Из доказанной теоремы 2.25 непосредственно вытекают следующие результаты.

I. Если классы  $\mathcal{K}$  и  $\mathcal{L}$  разрешимы, то разрешим и класс  $\mathcal{K} \wedge \mathcal{L}$ .

II. Пусть  $E_+^n$  — неотрицательный ортант в  $E^n$ . Рассмотрим выпуклые тела  $K$  такие, что существуют  $\rho > 0$  и  $R > 0$ , при которых

$$E_+^n \cap S(0, \rho) \subset K \subset E_+^n \cap S(0, R), \quad (2.129)$$

и если  $x \in K$ ,  $0 \leq y \leq x$ , то  $y \in K$ . Антиблокер  $K$  определяется так:

$$A(K) = \{y \in E_+^n : (y, x) \leq 1 \text{ для всех } x \in K\}.$$

Далее,  $A(\mathcal{K}) = \{A(K) : K \in \mathcal{K}\}$ .

Тогда, если  $\mathcal{K}$  — класс выпуклых тел, удовлетворяющих (2.129), то из разрешимости  $\mathcal{K}$  следует разрешимость  $A(\mathcal{K})$  и обратно, из разрешимости  $A(\mathcal{K})$  следует разрешимость  $\mathcal{K}$ .

Доказательство не отличается существенно от доказательства второй части теоремы 2.25.

Уточним полученные выше результаты для класса рациональных выпуклых многогранников. Под рациональным выпуклым многогранником  $P = P(n, a_0, T)$  будем понимать выпуклый многогранник размерности  $n$ , содержащийся в  $E^n$ ,  $a_0 \in \text{int } P$ , такой что каждая компонента точки  $a_0$  и любой вершины  $P$  выражается рациональными числами, числители и знаменатели которых не превышают  $T$  по абсолютной величине.

**Теорема 2.26.** Пусть  $P = P(n, a_0, T)$  — рациональный выпуклый многогранник. Тогда, полагая  $R = 2nT$  и  $r = (2T)^{-n^2-n}$ , получим

$$S(a_0, r) \subset P \subset S(a_0, R).$$

Далее, любая  $(n-1)$ -мерная грань  $P$  может быть записана в виде  $(a, x) \leq b$ , где  $a \neq 0$  — целочисленный вектор,  $b$  — целое чис-



ло, причем компоненты  $a$  и  $b$  не превышают по абсолютной величине  $T' = (nT)^n$ .

Справедливо в определенном смысле и обратное утверждение.

**Теорема 2.27.** Пусть  $P \subset E^n$  — выпуклый многогранник и задано  $a_0 \in \text{int } P$  с рациональными компонентами, числители и знаменатели которых не превышают  $T$  по абсолютной величине. Кроме того, любая  $(n-1)$ -мерная грань  $P$  может быть записана в виде  $(a, x) \leq b$ , где  $a \neq 0$  — целый вектор,  $b$  — целое число и компоненты  $a$  и  $b$  выражаются числами, меньшими  $T$  по абсолютной величине. Тогда  $P = P(n, a_0, T')$  — рациональный выпуклый многогранник с  $T' = (nT)^n$ .

Доказательство этих результатов содержится в [141].

При рассмотрении рациональных многогранников можно перейти от приближенных (слабых) решений задач отделения и оптимизации к их сильным (точным) постановкам. При этом получаются следующие результаты.

**Теорема 2.28.** Пусть  $\mathcal{K}$  — разрешимый класс рациональных выпуклых многогранников. Тогда сильная оптимизационная задача и сильная задача отделения разрешимы для  $\mathcal{K}$  за время, являющееся полиномом относительно  $n$ ,  $\log T$  и  $\log s$ , где  $s$  — максимальное значение абсолютных величин числителей и знаменателей компонент вектора  $c$  (или  $y$  соответственно).

Если решение сильной задачи отделения дает  $y \in P$ , то интересно выразить  $y$  как выпуклую комбинацию вершин  $P$ . Для разрешимого класса многогранников это также можно сделать за полиномиальное время.

**Теорема 2.29.** Пусть  $\mathcal{K}$  — разрешимый класс рациональных выпуклых многогранников. Тогда существует алгоритм, который для данного  $P = P(n, a_0, T) \in \mathcal{K}$  и рационального вектора  $y \in P$  выдает вершины  $x_0, x_1, \dots, x_n \in P_n$  и рациональные коэффициенты  $\lambda_0, \lambda_1, \dots, \lambda_n \geq 0$ ,  $\sum_{i=0}^n \lambda_i = 1$ , такие, что  $\sum_{i=0}^n \lambda_i x_i = y$ , за время, полиномиальное относительно  $n$ ,  $\log T$  и  $\log s$  ( $s$  — максимальное значение абсолютных величин числителей и знаменателей компонент  $y$ ).

Доказана также теорема о полиномиальном алгоритме получения точного двойственного решения оптимизационной задачи.

**Теорема 2.30.** Пусть  $\mathcal{K}$  — разрешимый класс выпуклых рациональных многогранников. Тогда существует полиномиальный алгоритм, который по заданным  $P(n, a_0, T) \in \mathcal{K}$ ,  $c \in E^n$  выдает грани  $(a_i, x) \leq b_i$ ,  $i = 1, \dots, n$ , и рациональные числа  $\lambda_i \geq 0$ ,  $i = 1, \dots, n$ , такие, что  $\sum_{i=1}^n \lambda_i a_i = c$  и  $\sum_{i=1}^n \lambda_i b_i = \max \{(c, x) : x \in P\}$ .

Теоремы 2.26 — 2.30 приведены в [141]. Их доказательство основаны на оценках, указанных в теоремах 2.24, 2.25 и показывающих, с какой точностью нужно решать возникающие задачи оптимизации и отделения, чтобы результат можно было довести до точного рационального решения.

Подытожим совокупность приведенных нами результатов. Они дают единообразный подход к исследованию сложности комбинаторных задач. Этот подход состоит в следующем:

1) попытаться в явном и достаточно компактном виде представить существенные грани многогранника  $K$ , являющегося выпуклым замыканием допустимых (целочисленных) решений задачи (в булевом случае допустимые решения составляют часть вершин многогранника  $K$ );

2) рассмотреть для полученной задачи линейного программирования (как правило, число ограничений в этой задаче будет экспоненциальным по отношению к размерности задачи  $n$ ) задачу отделения (в слабом смысле). Если эта задача является  $NP$ -трудной, то и первоначальная задача  $NP$ -трудная. Если задача отделения полиномиально разрешима относительно  $n$ ,  $|\log \varepsilon|$  для рассматриваемого класса  $\mathcal{K}$  выпуклых целочисленных многогранников, т. е. класс  $\mathcal{K}$  является разрешимым, тогда полиномиально разрешимы и первоначальная задача, а также двойственная к ней, и задача представления точки многогранника  $K \in \mathcal{K}$  в виде выпуклой комбинации его вершин (см. теоремы 2.25 — 2.30).

Как продемонстрировано в [141], такой подход позволяет сравнительно просто установить полиномиальную разрешимость почти всех классов известных задач, для которых она была установлена другими способами, а также получить новые результаты о полиномиальной разрешимости или  $NP$ -трудности других классов задач.

Так как доказательство упомянутых теорем 2.24 — 2.30 является конструктивным, причем в основу алгоритмов положен несколько модифицированный метод эллипсоидов, то при получении результатов о полиномиальной разрешимости той или иной проблемы мы сразу же получаем и некоторый полиномиальный алгоритм решения соответствующего класса задач, основанный на методе эллипсоидов. Как показывает анализ трудоемкости вычислений для приведенных нами трех примеров, эти алгоритмы оказываются более трудоемкими по сравнению с лучшими комбинаторными алгоритмами, специально разработанными для данного класса задач. Поэтому метод эллипсоидов является скорее средством открытия полиномиально разрешимых классов комбинаторных задач, чем средством их эффективного в практическом смысле решения.

В тех случаях, когда для многогранного множества  $P$  сложно выписать полную систему его существенных граней, для приближенного решения задачи можно с успехом использовать и достаточно большое хорошо описываемое подмножество граней. В качестве иллюстрации к сказанному рассмотрим задачу о коммивояжере [166].

Симметричная задача о коммивояжере формулируется следующим образом. Дан граф  $G(V, E)$  с множеством вершин  $V$ ,  $|V| = n$ , и множеством ребер  $E$ . Для ребер  $(i, j) \in E$  задан вес  $c_{ij} \geq 0$ , причем  $c_{ij} = c_{ji}$ . Обозначим через  $M$  множество гамильтоновых циклов графа  $G$ , т. е. простых циклов длины  $n$ . Пусть  $x(t) = \{x_{ij}(t)\}$ ,  $(i, j) \in E$ , — вектор размерности  $|E|$ , принимающий значение 0, если ребро  $(i, j)$  не входит в гамильтонов цикл (маршрут)  $t \in M$ , и 1, если

ребро  $(i, j)$  входит в  $t$ ,  $c(t) = \sum_{(i,j) \in E} c_{ij} x_{ij}(t)$  — длина маршрута  $t$ . Задача коммивояжера заключается в нахождении маршрута минимальной длины, т. е.  $\min_{t \in M} c(t)$ .

Без ограничения общности задачу коммивояжера можно рассматривать на полном графе, вводя недостающие ребра с большими весами. Если зафиксировать вершину  $n$  как начало и конец кольцевого маршрута, то задача коммивояжера сведется к выбору перестановки  $i_1, \dots, i_{n-1}$  вершин  $1, 2, \dots, n-1$  такой, что путь  $(n, i_1, i_2, \dots, i_{n-1}, n)$  является кратчайшим. Таким образом, в полном графе имеется  $(n-1)!$  маршрутов  $t \in M$ . Каждому такому маршруту можно сопоставить вершину  $x(t) = \{x_{ij}(t)\}$  единичного  $n(n-1)/2$ -мерного куба. Пусть  $Q^n$  — многогранник, являющийся выпуклой оболочкой точек  $x(t)$ ,  $t \in M$ . Дальнейшее изложение опирается на обзорную статью [166]. Справедливы следующие результаты.

1. Размерность  $Q^n$  равна  $n(n-3)/2$  для  $n \geq 3$ . Это легко вытекает из того факта, что справедливы  $n$  линейно независимых равенств

$$\sum_{\{j: i \neq j\}} x_{ij}(t) = 2, \quad i = 1, \dots, n. \quad (2.130)$$

2. Для всех  $n \geq 5$  неравенства  $x_{ij} \geq 0$  для различных пар  $(i, j)$  индуцируют различные существенные грани  $Q^n$ . Эти грани называются тривиальными гранями  $Q^n$ .

3. Для любого подмножества  $S \subset V$ , удовлетворяющего условию  $2 \leq |S| \leq n-2$ , можно ввести ограничение, исключающее короткие замкнутые циклы:

$$\sum_{(i,j) \in S} x_{ij} \leq |S| - 1. \quad (2.131)$$

Тогда для всех  $n \geq 4$  и  $S \subset V$ , удовлетворяющих  $2 \leq |S| \leq \lfloor n/2 \rfloor$ , индуцируются различные существенные грани  $Q^n$ . Ограничения (2.131) могут быть переформулированы в форме «разрезных»:

$$\sum_{(i,j) \in \delta(S)} x_{ij} \geq 2 \quad \forall \delta(S), \quad (2.132)$$

где  $\delta(S)$  — множество ребер, имеющих в точности одну вершину в  $S$ .

В [166] рассмотрены достаточно сложные конфигурации из вершин и ребер (так называемые «гребенки» (combs) и кликовые деревья), для каждой из которой выписываются определенные неравенства, являющиеся существенными гранями  $Q^n$ . Несмотря на то что эти классы неравенств имеют большую мощность, они не исчерпывают всех существенных граней  $Q^n$ . Однако они в большинстве случаев являются достаточными для получения оптимальных или близких к оптимальным решений реальных задач.

Для возможности успешного решения задач линейного программирования с экспоненциально растущим числом неравенств методом эллипсоидов нужно уметь решать задачу отделения, т. е. указания того неравенства, которое в данной точке не выполняется. В настоящее время известны эффективные алгоритмы решения задачи отделения лишь для некоторых классов неравенств, связанных с задачей коммивояжера. Так, для класса неравенств вида (2.132) можно использовать алгоритм нахождения минимального разреза в графе, в котором вес ребра  $(i, j)$  равен  $x_{ij}$ . Если значение минимального разреза меньше 2, то ему соответствует неравенство вида (2.132), которое в данной точке не выполняется.

На основе алгоритма Падберга — Рао [164] решения задачи отделения для  $b$ -паросочетаний можно построить алгоритм отделения для некоторого подкласса «гребеночных» неравенств. Однако более глубоких результатов пока нет.

Поэтому в настоящее время для решения задач коммивояжера при больших  $n$  применяют различные модификации известного подхода. Суть его состоит в том, что решается задача нахождения

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2.133)$$

при ограничениях

$$0 \leq x_{ij} \leq 1, \quad (i, j) \in E, \quad (2.134)$$

$$\sum_{\{j: i \neq j\}} x_{ij} = 2, \quad i \in V. \quad (2.135)$$

Если решению  $x_i^*$  этой задачи не соответствует допустимый маршрут, то добавляют невыполняющиеся ограничения вида (2.132) или более сложные, решают более сложную задачу линейного программирования и получают новое решение. Если оно недопустимо, добавляют новые ограничения и т. д. с использованием этого подхода в 1980 г. было получено точное решение задачи коммивояжера при  $n = 120$ ; к ограничениям вида (2.134), (2.135) было добавлено 36 неравенств вида (2.132) и 60 «гребеночных» неравенств. При этом последовательно было решено 13 задач линейного программирования. При построении дополнительных неравенств использовались эвристические методы. Затем метод добавления неравенств был автоматизирован, и в 1980 г. было получено близкое к оптимальному решение при  $n = 318$ . Отыскание новых классов существенных неравенств для задачи коммивояжера, для которых проблема делимости полиномиально разрешима, представляет значительный интерес.

## ЛИНЕЙНЫЕ И ВЫПУКЛЫЕ ЗАДАЧИ ПРОИЗВОДСТВЕННО-ТРАНСПОРТНОГО ПЛАНИРОВАНИЯ

---

### § 1. Введение

Благодатным полем для применения методов недифференцируемой оптимизации служат задачи производственно-транспортного планирования. Многие из них описываются моделями линейного и выпуклого программирования большой размерности с ярко выраженными структурными особенностями, позволяющими применить рациональные схемы декомпозиции и свести их к задачам минимизации негладких выпуклых функций небольшой и средней размерности. В данной главе описываются конкретные модели оптимизационных задач производственно-транспортного планирования, приемы декомпозиции, вытекающие из специфики конкретной задачи, алгоритмы решения и опыт практического применения. Следует отметить, что большая часть рассматриваемых задач имела конкретных заказчиков и результаты оптимизационных расчетов использовались при решении вопросов текущего и перспективного планирования в таких отраслях, как различные виды транспорта, материально-техническое снабжение, строительство, черная металлургия, гражданская авиация и др.

Вначале рассмотрены транспортные задачи линейного программирования в матричной и сетевой формах, а также многопродуктовые задачи на сети с ограниченными пропускными способностями. Для этих задач известно большое число конечношаговых и итеративных алгоритмов, причем наибольшую эффективность показывают программы, основанные на методе потенциалов со специальной организацией просмотра информации. В данной монографии для транспортных задач рассматриваются только алгоритмы, основанные на методах недифференцируемой оптимизации. Дело в том, что именно на сетевых транспортных задачах была впервые продемонстрирована эффективность методов субградиентного спуска при решении задач большой размерности [5,82]. Опыт, полученный при решении транспортных задач, послужил основой создания быстро сходящихся алгоритмов решения гораздо более сложных обобщенных распределительных, стохастических и различных типов динамических распределительных задач, для которых уже нет специальных эффективных конечношаговых алгоритмов, а стандартные процедуры симплекс-метода оказываются неэффективными из-за очень большой размерности задач.

Именно такого рода задачам посвящен основной материал данной главы. Специфика задач производственно-транспортного плани-

рования позволяет сравнительно легко находить приближенные значения двойственных переменных. Для того чтобы перейти от близкого к оптимальному решению двойственной задачи к решению прямой задачи, необходимы специальные дополнительные процедуры (см. гл. 1) или модификации алгоритма. Одной из таких модификаций — приему  $\varepsilon$ -квадратичного сглаживания — посвящен § 5 данной главы.

## § 2. Транспортные задачи в матричной и сетевой формах

Транспортная задача в матричной форме состоит в следующем. Задано  $m$  пунктов производства  $P_i$  с мощностями  $a_i > 0$  ( $i = 1, \dots, m$ ) и  $n$  пунктов потребления  $S_j$  с потребностями  $b_j$  ( $j = 1, \dots, n$ ), при этом предполагается, что  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$  (сбалансированный случай). Кроме этого, задана матрица  $C = \{c_{ij}\}_{i,j=1}^{m,n}$  размера  $m \times n$ , где  $c_{ij}$  интерпретируется как стоимость перевозок (расстояние) между пунктами  $P_i$  и  $S_j$ . Следует определить переменные  $x_{ij}$  ( $i = 1, \dots, m$ ;  $j = 1, \dots, n$ ) (количество продукции, поставляемой из пункта  $P_i$  в пункт  $S_j$ ) таким образом, чтобы обеспечить потребности в каждом пункте потребления с учетом ограничений на мощности в каждом пункте производства с минимальными общими затратами на перевозку продукции.

Математическая формулировка задачи следующая: найти

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

при ограничениях

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = 1, \dots, m, \quad (3.2)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n, \quad (3.3)$$

$$x_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (3.4)$$

Задача (3.1) — (3.4) — специальная задача линейного программирования с  $mn$  неизвестными и  $(m + n)$  нетривиальными ограничениями, из которых одно является следствием остальных, так что независимых ограничений в форме равенства —  $m + n - 1$ .

Матрица коэффициентов ограничений транспортной задачи, рассматриваемой как задача линейного программирования, обладает замечательным свойством: любой минор этой матрицы принимает значения  $\pm 1$  или 0. Такая матрица называется вполне унимодулярной. Благодаря этому свойству любое опорное решение задачи (3.1) — (3.4) при целочисленном векторе правых частей ограничений является целочисленным, в том числе и оптимальное опорное решение.

Если  $m = n$  и все  $a_i$  и  $b_j$  равны 1, то получаем частный случай транспортной задачи — задачу о назначениях. Для ее решения

требуется  $O(n^3)$  арифметических операций; эта оценка достигается на так называемом венгерском алгоритме [24].

Транспортная задача с целыми правыми частями ограничений легко может быть сведена к задаче о назначениях размера  $\sum_{i=1}^m a_i$ . На основе этой идеи в [126] разработан точный алгоритм решения транспортной задачи с оценкой числа арифметических операций

$$O(mn^2(\lg A + 1)), \quad (3.5)$$

где  $A = \max_{i,j} \{a_i, b_j\}$ , ( $a_i, b_j$  — целые). Пока не удалось построить полиномиальный алгоритм, у которого число операций определяется только  $m$  и  $n$  и не зависит от правых частей ограничений.

Практически в настоящее время признаются наиболее эффективными алгоритмы, основанные на методе потенциалов (модификация прямого симплекс-метода для транспортной задачи) со специальной организацией памяти и вычислений [3]. Об этом говорят, в частности, результаты проведенного в 1984 г. конкурса программ решения транспортной задачи, однако для числа итераций в методе потенциалов нет полиномиальных оценок. Поэтому интерес к итеративным методам решения транспортных задач при большой размерности сохраняется.

Опишем, как могут быть применены для решения транспортной задачи методы субградиентного типа. Задача, двойственная к транспортной, имеет следующий вид: найти

$$\max \left( \sum_{j=1}^n b_j v_j - \sum_{i=1}^m a_i u_i \right) \quad (3.6)$$

при ограничениях

$$v_j - u_i \leq c_{ij}, \quad i = 1, \dots, m; \quad j = 1, \dots, n; \quad (3.7)$$

здесь  $v_j$  — двойственные переменные (потенциалы), соответствующие ограничениям (3.3) для пунктов потребления,  $u_i$  — двойственные переменные (потенциалы), соответствующие ограничениям (3.2) для пунктов производства. При фиксированных  $\{u_i\}_{i=1}^m$  каждое  $v_j$  ограничено сверху соотношениями

$$v_j \leq u_i + c_{ij}, \quad i = 1, \dots, m.$$

Так как  $b_j > 0$ , для получения максимума по  $v_j$  в (3.6) при фиксированных  $\{u_i\}_{i=1}^m$  должно выполняться соотношение

$$v_j = \min_{1 \leq i \leq m} (u_i + c_{ij}). \quad (3.8)$$

Таковыми соотношениями должны быть связаны потенциалы  $\{u_i\}_{i=1}^m$  и  $\{v_j\}_{j=1}^n$  в оптимальном плане двойственной транспортной задачи, при этом все ограничения (3.7) выполняются автоматически. Отсюда вытекает, что задача, двойственная к транспортной, сводится к следующей

задаче безусловной максимизации: найти

$$\max_{\{u_i\}} \varphi(u) = \max_{\{u_i\}} \left( \sum_{j=1}^n b_j \min_{1 \leq i \leq m} (u_i + c_{ij}) - \sum_{i=1}^m a_i u_i \right), \quad (3.9)$$

где  $\varphi(u)$  — функция, получающаяся в результате подстановки в (3.6) вместо  $v_j$  его выражения (3.8).

Функция  $\varphi(u)$  определена для любого  $u \in E^m$ , кусочно линейна и вогнута. Разрывы градиента функции  $\varphi(u)$  соответствуют разрывам градиентов одной из функций  $\psi_j(u) = \min_{1 \leq i \leq m} (u_i + c_{ij})$ , которые происходят при таких значениях  $u$ , когда минимум в выражении для  $\psi_j(u)$  достигается сразу при двух или большем числе значений  $i$ . Число кусков гладкости функции  $\psi_j(u)$  равно  $m$ , поэтому верхней границей для числа кусков гладкости функции  $\varphi(u)$  является  $m^2$ .

Задача (3.9), вытекающая из транспортной задачи, фактически явилась той первой моделью, на которой отработывались методы субградиентного спуска и была показана их практическая эффективность при решении задач большой размерности [5]. Решение задачи (3.9) может быть получено одним из описанных в гл. 2 методов субградиентного типа. Эффективность этих методов увеличивается, когда  $m \ll n$ . Если решать задачу (3.9) с помощью  $r$ -алгоритма, то на каждой итерации придется делать  $O(m^2 + mn)$  арифметических операций, а эмпирическая оценка для числа итерации —  $O(m \ln 1/\varepsilon)$ , где  $\varepsilon$  — относительная погрешность определения максимального значения функционала в (3.9). Таким образом, общее число операций оценивается величиной  $O(m^2(m + n) \ln 1/\varepsilon)$ , где  $m \leq n$ , что сравнимо с (3.5).

Решив с достаточной точностью задачу (3.9), получаем значения  $\bar{u}^*$ ,  $\bar{v}^*$ , близкие к оптимальным значениям переменных двойственной задачи  $u^*$ ,  $v^*$ .

Рассмотрим двудольный граф  $D$  с вершинами  $i$  ( $i = 1, \dots, m$ ), соответствующими поставщикам, и  $j$  ( $j = 1, \dots, n$ ), соответствующими потребителям. Ребрами этого графа служат пары  $(i, j)$ , выделенные на основе приближенного выполнения условия оптимальности при некотором  $\varepsilon > 0$

$$\{(i, j) : \bar{v}_j - \min_{1 \leq k \leq m} (\bar{u}_k + c_{kj}) \geq \bar{u}_i + c_{ij} - \varepsilon, i = 1, \dots, m, j = 1, \dots, n\}.$$

Ребрам графа  $D$  соответствуют переменные  $x_{ij}$ , которые в оптимальном решении могут принимать ненулевые значения. Если полученный граф несвязный, то каждую его компоненту связности нужно рассматривать отдельно. Поэтому будем считать, что граф  $D$  связный. Если транспортная задача имеет единственное решение, то при определении  $\bar{u}^*$  с высокой точностью и достаточно малом  $\varepsilon > 0$   $D$  будет представлять собой дерево. Переменные  $x_{ij}$ , соответствующие висячим ребрам дерева, определяются однозначно. После отбрасывания висячих ребер получаем редуцированное дерево  $D^1$ , висячим ребрам которого соответствуют однозначно получаемые переменные, и т. д.



Таким способом однозначно восстанавливается единственное оптимальное решение. В случае когда транспортная задача имеет не единственное оптимальное решение, количество ребер в графе  $D$  превосходит  $m + n - 1$ , и в нем появляются циклы. При наличии висячих ребер часть переменных однозначно определяется описанным выше способом. Для получения решения небольшой, как правило, транспортной задачи, которой соответствует редуцированный подграф без висячих ребер, может быть применен алгоритм со сглаживанием (см. § 5 гл. 3).

Перейдем к рассмотрению транспортной задачи в сетевой форме. При решении транспортных задач большой размерности на сети железных или автомобильных дорог удобно вместо вычисления огромных таблиц кратчайших расстояний (или удельных стоимостей) перевозок  $\{c_{ij}\}_{i,j=1}^{m,n}$  задавать информацию об этих величинах в сетевой форме. Это значит, что соответствующая транспортная сеть задается в форме графа  $G(V, R)$ ,  $V$  — множество вершин,  $R$  — множество ребер. Для простоты изложения примем, что  $G$  — неориентированный связный граф без петель, что соответствует возможности двустороннего движения по каждому ребру с одинаковыми эксплуатационными характеристиками в обоих направлениях.

Каждое ребро  $r_{ij} \in R$ , соединяющее пару вершин  $(i, j)$ , имеет длину  $d_{ij}$ . Выделены вершины  $\omega_1, \dots, \omega_m \in V$ , соответствующие пунктам производства, и  $\omega^1, \dots, \omega^n \in V$ , соответствующие пунктам потребления.

Заданы мощности  $a_1, \dots, a_m$  в пунктах производства и потребности  $b_1, \dots, b_n$  в пунктах потребления, причем  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ . Нужно оп-  
ределить величины  $x_{ij}$ , соответствующие задаче (3.1) — (3.4), предполагая, что расстояния  $c_{ij}$  принимаются равными кратчайшим расстояниям от  $\omega_i$  к  $\omega^j$ .

При решении транспортной задачи в матричной форме матрица  $C = \{c_{ij}\}_{i,j=1}^{m,n}$  используется на каждой итерации при нахождении  $v_j^{(k)} = \min_{1 \leq i \leq m} (u_i^{(k)} + c_{ij})$ ,  $k$  — номер итерации. В сетевой форме матрица  $C$  задана неявно, однако приведенная выше операция определения  $v_j^{(k)}$  ( $j = 1, \dots, n$ ) осуществляется достаточно эффективно. Для этого достаточно к графу  $G$  добавить вершину  $v_0$ , соединить ее ребрами  $r_{0i}$  с вершинами-поставщиками  $\omega_i$  ( $i = 1, \dots, m$ ) и придать каждому ребру  $r_{0i}$  длину  $u_i^{(k)}$ . Тогда, очевидно,  $v_j^{(k)}$  будет равно кратчайшему расстоянию от вершины  $\omega^j$  до  $v_0$ , при этом считается, что если кратчайший путь на  $k$ -й итерации от  $\omega^j$  к  $v_0$  идет через ребро  $r_{0i}$ , то на этой итерации  $j$ -й потребитель «прикрепляется» к  $i$ -му поставщику. Так как существуют простые эффективные алгоритмы построения кратчайших путей от одной вершины графа до всех остальных [82], то реализация алгоритма субградиентного типа для транспортной задачи в сетевой форме тоже в достаточной степени эффективна.

Другой способ решения сетевой транспортной задачи — использование так называемых контурных переменных. Сформулируем сете-

вую транспортную задачу в виде задачи о потоке. Пусть задан граф  $G(V, R)$  с вершинами  $i = 1, \dots, N$  и ребрами  $(i, j) \in R$ , соединяющими вершины  $i$  и  $j$ ,  $\rho(i)$  — множество вершин, инцидентных вершине  $i$ . Каждому ребру  $(i, j) \in R$  сопоставим длину  $d_{ij}$  и потоковые переменные  $y_{ij}$ . Пусть  $P \subset V$  — множество вершин-источников (поставщиков); каждому  $i \in P$  сопоставим число  $a_i > 0$  (мощность). Далее,  $S \subset V$  — множество вершин-стоков; каждому  $j \in S$  сопоставим число  $b_j > 0$ , причем  $\sum_{i \in P} a_i = \sum_{j \in S} b_j$ . Тогда сетевая транспортная задача принимает следующую форму: определить поток  $Y = \{y_{ij}\}$ ,  $(i, j) \in R$  так, чтобы выполнялись следующие условия:

$$y_{ij} = -y_{ji}; \quad (i, j) \in R; \quad (3.10)$$

$$\sum_{j \in \rho(i)} y_{ij} = a_i; \quad i \in P; \quad (3.11)$$

$$\sum_{i \in \rho(j)} y_{ij} = b_j; \quad j \in S; \quad (3.12)$$

$$\sum_{j \in \rho(i)} y_{ij} = 0; \quad i \in V \setminus (S \cup P), \quad (3.13)$$

и при этом получить

$$\min C(Y) = \min \sum_{(i,j) \in R} d_{ij} |y_{ij}|. \quad (3.14)$$

Будем предполагать, что граф  $G(V, R)$  связный. Последовательность ребер  $(i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)$  назовем простым циклом, если  $i_1 = i_k$ , а среди вершин  $i_1, \dots, i_{k-1}$  нет совпадающих. Если связный граф  $G(V, R)$  не имеет циклов, то он называется деревом, при этом  $|V| = |R| + 1$ . В общем случае величина  $|R| + 1 - |V|$  неотрицательна и называется цикломатическим числом  $\lambda(G)$ . Если  $\lambda(G) > 0$ , то, удаляя последовательно  $\lambda(G)$  ребер, можно «разорвать» все циклы и превратить граф в остовное дерево. Система уравнений (3.10) — (3.13) для графа-дерева решается однозначно. В общем случае обозначим через  $Q$  совокупность ребер  $(i, j)$ , удаление которых превращает граф  $G$  в остовное дерево. Если принять  $y_{ij} = 0$ ,  $(i, j) \in Q$ , то уравнения (3.10) — (3.13) относительно остальных переменных решаются однозначно, и этому решению соответствует некоторый допустимый поток  $Y_0$ . Упорядочим определенным образом ребра  $(i, j) \in Q$ , дав им номера  $1, 2, \dots, |Q|$ , и пусть соответствующие переменные  $y_{ij}$  образуют вектор  $z = \{z_1, \dots, z_{|Q|}\}$ . После этого любая переменная  $y_{ij}$ ,  $(i, j) \in R$ , однозначно представляется как линейная функция  $l_{ij}(z)$ . Выражение  $C(Y)$  принимает следующий вид:

$$C(Y) = \bar{C}(z) = \sum_{(i,j) \in R} d_{ij} |l_{ij}(z)|, \quad (3.15)$$

при этом «контурные» переменные  $z$  являются свободными. Таким образом, сетевая транспортная задача сводится к задаче безусловной минимизации кусочно линейной выпуклой функции (3.15), причем размерность задачи определяется цикломатическим числом графа  $G$ .

Этот же прием можно использовать для решения нелинейных сетевых транспортных задач, когда вместо целевой функции (3.14) используется целевая функция более общего вида

$$C_1(Y) = \sum_{(i,j) \in R} \varphi_{ij}(y_{ij}).$$

Если все функции  $\varphi_{ij}$  являются выпуклыми, то получаем задачу выпуклой минимизации: найти

$$\min \sum_{(i,j) \in R} \varphi_{ij}(l_{ij}(z)), \quad (3.16)$$

для решения которой можно использовать эффективные методы субградиентного типа.

Если не все  $\varphi_{ij}$  являются выпуклыми функциями, то задача (3.16) может стать многоэкстремальной, однако для получения локальных экстремумов можно также использовать достаточно эффективно описанные в предыдущей главе методы недифференцируемой оптимизации, в частности  $r$ -алгоритм. Такой подход достаточно успешно использовался при решении нелинейных транспортных задач определения оптимального распределения потоков газа в Единой газоснабжающей системе СССР как в определенный год, так и с учетом динамики ее развития. Более подробное описание этой задачи можно найти в монографии [15].

### § 3. Многопродуктовые сетевые транспортные задачи

При решении вопросов реконструкции транспортной сети, увеличения пропускной способности ее узлов и участков необходимо рассматривать одновременно перевозки различных видов грузов. При этом возникают многопродуктовые транспортные сетевые задачи с ограничениями на пропускные способности элементов сети. Опишем соответствующую математическую модель.

Пусть  $G(V, R)$  — связный граф. Под путем  $\pi_{ij}$ , соединяющим вершины  $i$  и  $j$ , будем понимать цепочку ребер вида  $(i, k_1), (k_1, k_2) \dots (k_l, j)$ , причем среди этих ребер нет совпадающих. Будем говорить, что ребро  $(t_1, t_2)$  принадлежит пути  $\pi_{ij}$ , если оно совпадает с одним из ребер цепочки, определяющей путь  $\pi_{ij}$ . Совокупность путей, соединяющих  $i$  с  $j$ , будем обозначать  $\Pi_{ij}$ . На сети перевозится  $N$  видов продуктов. Множества вершин  $P^k$  и  $S^k$  являются соответственно поставщиками и потребителями  $k$ -го продукта, причем для любой вершины  $i \in P^k$  известна мощность производства  $a_i^k$ , а для любой

вершины  $j \in S^k$  — объем потребления  $b_j^k$   $k$ -го продукта, и  $\sum_{i \in P^k} a_i^k = \sum_{i \in S^k} b_i^k$  ( $k = 1, \dots, N$ ). Каждому ребру  $(i, j) \in R$  сопоставлены длина  $d_{ij}$  и пропускная способность  $\rho_{ij}$ . Пусть  $x_{ij}^k(\pi)$  — величина перевозок из  $i$ -го пункта производства  $k$ -го продукта в  $j$ -й пункт его потребления по пути  $\pi$ . Тогда задача об оптимальной транспортировке  $N$  продуктов при ограниченной пропускной способности ребер сети формулируется следующим образом: найти

$$\min \sum_{k=1}^N \sum_{i \in P^k} \sum_{j \in S^k} \sum_{\pi \in \Pi_{ij}} x_{ij}^k(\pi) d(\pi) \quad (3.17)$$

при ограничениях

$$\sum_{i \in S^k} \sum_{\pi \in \Pi_{ij}} x_{ij}^k(\pi) = a_i^k, \quad i \in P^k, \quad k = 1, \dots, N; \quad (3.18)$$

$$\sum_{i \in P^k} \sum_{\pi \in \Pi_{ij}} x_{ij}^k(\pi) = b_j^k, \quad j \in S^k, \quad k = 1, \dots, N; \quad (3.19)$$

$$\sum_{k=1}^N \sum_{i \in P^k} \sum_{j \in S^k} \sum_{\pi \in A(\alpha, \beta)} x_{ij}^k(\pi) \leq \rho_{\alpha\beta}, \quad (\alpha, \beta) \in R; \quad (3.20)$$

$$x_{ij}^k(\pi) \geq 0, \quad k = 1, \dots, N, \quad i \in P^k, \quad j \in S^k, \quad \pi \in \Pi_{ij}. \quad (3.21)$$

Здесь  $d(\pi) = \sum_{(\alpha, \beta) \in \pi} d_{\alpha\beta}$  (суммирование идет по ребрам, принадлежащим пути  $\pi$ ),  $A(\alpha, \beta)$  — совокупность путей, проходящих через ребро  $(\alpha, \beta)$ .

Можно предложить три способа решения задачи (3.17) — (3.21), основанные на методах недифференцируемой оптимизации.

Первый способ основан на схеме декомпозиции по ограничениям. Ограничения (3.20) с неопределенными множителями Лагранжа  $u = \{u_{\alpha\beta}\}$  переносятся в целевую функцию. При фиксированных  $\bar{u} \geq 0$  задача (3.17) — (3.19), (3.21) распадается на  $N$  транспортных сетевых задач, причем длина ребер сети  $(\alpha, \beta)$  равна  $\bar{d}_{\alpha\beta} = d_{\alpha\beta} + \bar{u}_{\alpha\beta}$ . Предположим, что имеется достаточно эффективная программа решения сетевых транспортных задач, использующая конечношаговый метод, например метод потенциалов [25]. Решив при заданном  $\bar{u}$   $N$  транспортных задач, получим суммарную целевую функцию затрат  $\psi(\bar{u})$  и суммарные потоки вдоль каждого ребра  $(\alpha, \beta) \in R$   $y_{\alpha\beta}^*(\bar{u}) = \sum_{k=1}^N |y_{\alpha\beta}^k(\bar{u})|$ ,

где  $y_{\alpha\beta}^k(\bar{u})$  — оптимальный поток по ребру  $(\alpha, \beta)$  для  $k$ -й транспортной задачи. В соответствии с общей схемой декомпозиции по ограничениям вектор  $\{y_{\alpha\beta}^*(\bar{u}) - \rho_{\alpha\beta}\}_{(\alpha, \beta) \in R}$  определяет обобщенный градиент функции

$\psi(u)$  в точке  $\bar{u}$ . Теперь мы можем применить для максимизации  $\psi(u)$  при  $u \geq 0$  один из методов недифференцируемой оптимизации субградиентного типа. Если  $\sup \psi(u) = +\infty$ , то задача (3.17) — (3.21) несовместна. В противном случае получим оптимальные значения двойственных переменных  $u^*$ , а затем можно воспользоваться одним из способов получения решения прямой задачи линейного программирования по оптимальному решению двойственной, описанному в § 4 гл. 1. Заметим, что если сетевые транспортные задачи решаются методом потенциалов, то, учитывая, что при изменении  $u$  меняются лишь коэффициенты целевых функций транспортных задач, целесообразно использовать на очередной итерации при решении транспортных задач в качестве начального приближения оптимальное решение, полученное на предыдущей итерации. Это может существенно сократить время счета.

Второй способ также основан на схеме декомпозиции по ограничениям, однако декомпозиция производится другим способом: ограничения (3.18) и (3.20) с неопределенными множителями Лагранжа  $\omega = \{\omega_i^k, \omega_{\alpha\beta}\}$  переносятся в целевую функцию. После этого задача при каждом фиксированном  $\omega = \bar{\omega}$  распадается на  $\sum_{k=1}^N |S^k|$  блоков вида (3.1), (3.3), (3.4), для получения решения которых нужно  $N$  раз использовать процедуру нахождения кратчайших расстояний, подобную той, которая предлагалась для решения сетевой транспортной задачи на с. 112. Пусть  $\varphi(\omega)$  — суммарное значение функции Лагранжа при заданном  $\omega$ . Невязки в ограничениях (3.18), (3.20) дадут нам компоненты обобщенного градиента функции  $\varphi(\omega)$ . Максимизация этой функции может производиться одним из методов недифференцируемой оптимизации, что дает возможность в случае совместности ограничений (3.18) — (3.21) получить оптимальные значения двойственных переменных  $\omega^*$ .

Второй способ решения задачи (3.17) — (3.21) — использование «контурных» переменных подобно тому, как это предлагалось для решения сетевой транспортной задачи (см. с. 113). Выделим в графе  $G(V, R)$  остовное дерево, тогда поток  $k$ -го продукта ( $k = 1, \dots, N$ ) по любому ребру  $(\alpha, \beta)$   $y_{\alpha\beta}^k$  будет линейной функцией от потоков по ребрам, не входящих в остовное дерево. Вектор этих потоков обозначим через  $z^k$ ; тогда поток  $l_{\alpha\beta}(z^k)$  вдоль ребра  $(\alpha, \beta)$  в  $k$ -й задаче является линейной функцией  $z^k$  и обладает свойством  $l_{\alpha\beta}(z^k) = -l_{\beta\alpha}(z^k)$ . В терминах переменных  $z^k$  задача (3.17) — (3.21) принимает следующий вид: найти

$$\min \sum_{k=1}^N \sum_{(\alpha, \beta) \in R} d_{\alpha\beta} |l_{\alpha\beta}(z^k)|, \quad (3.22)$$

при ограничениях

$$\sum_{k=1}^N |l_{\alpha\beta}(z^k)| \leq \rho_{\alpha\beta}, \quad (\alpha, \beta) \in R. \quad (3.23)$$

Относительно  $z = \{z^k\}_{k=1}^N$  получаем задачу выпуклого кусочно линейного программирования. Используя метод негладких штрафных функций, задачу (3.22), (3.23) легко свести к задаче безусловной минимизации негладкой выпуклой функции от  $N\lambda(G)$  переменных ( $\lambda(G)$  — цикломатическое число графа  $G$ ).

Оценивая эффективность трех указанных способов решения многопродуктовой транспортной задачи, следует заметить, что в первом способе координирующая задача имеет наименьшую размерность в худшем случае —  $|R|$  (практически пропускные способности ограничиваются не на всех ребрах) — однако внутренняя задача требует использования хорошего модуля решения транспортной задачи; во втором способе размерность координирующей задачи равна  $\sum_{k=1}^N |P^k| + |R|$ , зато внутренняя задача решается быстро.

Третий способ эффективен, когда цикломатическое число графа невелико. Кроме того, используя третий способ, мы сразу получаем решение прямой задачи, в то время как первый и второй способы требуют дополнительной процедуры получения решения прямой задачи по оптимальному решению двойственной.

Отметим, что предложенные алгоритмы решения многопродуктовой сетевой задачи легко обобщаются на случай ориентированных сетей и мультисетей (когда имеются параллельные дуги или ребра), а также когда заданы ограничения на пропускные способности в узлах. Третий способ решения легко обобщается на задачи с нелинейными выпуклыми функциями затрат вдоль ребер.

#### § 4. Динамические распределительные задачи

Планирование закупок долговременно работающего оборудования и машин, согласование во времени поставок материалов с учетом неравномерности их производства и потребления и необходимости сохранения по возможности постоянства связей «поставщик — потребитель», долгосрочное планирование развития энергетических и трубопроводных систем приводят к различного рода динамическим распределительным задачам, как линейным, так и нелинейным, и, как правило, высокой размерности. Методы негладкой оптимизации в сочетании со схемами декомпозиции и использованием точных негладких функций штрафа позволяют успешно решать многие задачи такого типа. Отметим, что динамические распределительные задачи довольно разнообразны по своей структуре, поэтому мы приведем несколько характерных моделей и для каждой из них предложим свой специфический подход к решению, который кажется нам наиболее рациональным. Наименования, которые мы будем давать конкретным моделям, несколько условны в том смысле, что эти модели, конечно, далеко не в полной мере отражают все существенные технико-экономические подробности и сложности, присущие реальным задачам, которым соответствуют эти названия.

## 1. Задача об оптимальной структуре машинно-тракторного парка.

Для сельскохозяйственного производства характерным является сезонность, резкая неоднородность во времени выполняемых работ, наличие как достаточно универсальной техники, так и специализированных и высокопроизводительных машин и агрегатов. При оценке эффективности той или иной машины важное значение имеет не только ее производительность, но и на каких работах и какое время она используется в течение планового периода. В силу этого задача определения оптимальной структуры машинно-тракторного парка требует комплексного учета как объемов, так и сроков проведения всех основных работ сельскохозяйственного предприятия.

Ниже мы рассмотрим одну из наиболее часто применяемых линейных моделей для расчета оптимальной структуры машинно-тракторного парка сельскохозяйственного предприятия по критерию минимума приведенных затрат на закупку и эксплуатацию машинно-тракторного парка при заданных объемах работ по периодам: найти

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T c_{ijt} x_{ijt} + \sum_{i=1}^m k_i y_i \right) \quad (3.24)$$

при ограничениях

$$\sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T p_{ijt} x_{ijt} = b_{jt}, \quad j = 1, \dots, n, \quad t = 1, \dots, T, \quad (3.25)$$

$$\sum_{j=1}^n x_{ijt} \leq d_i + y_i, \quad i = 1, \dots, m, \quad t = 1, \dots, T, \quad (3.26)$$

$$x_{ijt}, y_i \geq 0, \quad j = 1, \dots, n, \quad i = 1, \dots, m, \quad t = 1, \dots, T, \quad (3.27)$$

где  $x_{ijt}$  — количество машин  $i$ -го типа, занятых в период  $t$  на работе  $j$ ,  $y_i$  — количество дополнительно покупаемых машин  $i$ -го типа (ввиду большой размерности рассматриваемой задачи мы не накладываем на переменные  $y_i$  требование целочисленности; это требование учитывается на заключительной стадии расчетов с использованием эвристических соображений),  $d_i$  — наличное количество машин  $i$ -го типа перед началом планового периода,  $c_{ijt}$  — приведенные удельные эксплуатационные затраты при использовании машин  $i$ -го типа в  $t$ -й период на  $j$ -й работе,  $p_{ijt}$  — производительность  $i$ -й машины на  $j$ -й работе в  $t$ -й период,  $b_{jt}$  — объем  $j$ -й работы в  $t$ -й период.

Характерные размеры задачи таковы: количество периодов  $T$  — от нескольких десятков до сотни, общее количество планируемых работ по всем периодам — порядка  $10^3$ , каждая работа может выполняться 5—10 технологическими способами. Таким образом, приведенная выше задача линейного программирования может иметь порядка  $5 \times 10^3$  переменных и порядка  $10^3$  ограничений. Решение задач такого размера на машинах среднего класса стандартными средствами, использующими симплекс-метод, вызывает определенные трудности, особенно при массовом решении задач, так как требует солидного времени счета.

Задача (3.24) — (3.27) может быть несколькими способами сведена к задаче недифференцируемой оптимизации.

а) Легко увидеть, что при фиксации вектора  $y = \{y_i\}_{i=1}^m$  задача (3.24) — (3.27) распадается на  $T$  распределительных задач сравнительно небольшого размера. Поэтому естественно попытаться использовать схему декомпозиции по переменным. Пусть  $\Phi(y)$  — значение целевой функции задачи (3.24) — (3.27) при фиксированном  $y$ . Для того чтобы эта функция была определена для любого  $y \geq 0$ , нужно, чтобы при любом  $y \geq 0$  блоки распределительных задач для каждого периода имели допустимое решение. Этого можно добиться, несколько модифицировав задачу, — добавив в каждое ограничение (3.26) переменное  $z_{jt}$  и введя его в целевую функцию с достаточно большим положительным коэффициентом  $c_{jt}$  (это соответствует введению дополнительной «супермашины», которая может выполнять любую работу, но с очень большими удельными эксплуатационными затратами). Ниже речь будет идти о модифицированной таким образом задаче. Оптимальный план модифицированной задачи при достаточно больших  $c_{jt}$  будет совпадать с оптимальным планом первоначальной задачи.

Мы свели задачу (3.24) — (3.27) к задаче нахождения  $\min_{y \geq 0} \Phi(y)$ . В соответствии с теоремой 1.12  $\Phi(y)$  — выпуклая кусочно линейная функция. Обозначим оптимальные значения двойственных переменных (потенциалы), соответствующих ограничениям (3.26) при фиксированном  $y$ , через  $\{u_{it}(y)\}$ . Тогда из формулы (1.25) вытекает следующее выражение для субградиента  $\Phi(y)$ :

$$g_{\Phi}(y) = \left\{ k_i - \sum_{t=1}^T u_{it} \right\}_{i=1}^m.$$

Теперь у нас есть все необходимое для отыскания  $\min_{y \geq 0} \Phi(y)$  одним из субградиентных процессов. Так как размерность вектора  $y$  невелика (порядка нескольких десятков), можно ожидать, что при использовании  $r$ -алгоритма для получения решения с достаточно высокой точностью потребуется порядка 100—150 итераций.

Так как каждая итерация сопряжена с решением  $T$  распределительных задач, для эффективной реализации предлагаемого подхода требуется быстрый конечношаговый алгоритм решения распределительных задач небольшой размерности. Отметим, что при переходе от одной итерации к другой изменяются правые части ограничений распределительных задач. Поэтому, если на последующем шаге для ускорения процесса решения распределительных задач мы хотим использовать в качестве начального приближения решение, полученное на данной итерации, желательно использовать некоторую модификацию двойственного симплекс-метода (конечно, учитывающую специфику распределительной задачи).

Указанная схема кажется достаточно привлекательной, однако практически она пока не реализована.

б) Пусть зафиксирована группа неотрицательных переменных  $\{x_{ijt}\}$ ,  $i = 1, \dots, m$ ;  $j = 1, \dots, n$ ;  $t = 1, \dots, T$ , удовлетворяющих ограниче-



ниями (3.25), (3.26), и справедливы неравенства

$$y_i \geq d_{it} - \sum_{j=1}^n x_{ijt}, \quad t = 1, \dots, T, \quad i = 1, \dots, m.$$

Отсюда вытекает, что в оптимальном плане задачи (3.24) — (3.27)

$$y_i = \bar{y}_i = \max \left\{ 0; \max_{1 \leq t \leq T} \left( d_{it} - \sum_{j=1}^n x_{ijt} \right) \right\}, \quad i = 1, \dots, m. \quad (3.28)$$

Подставив полученное выражение для  $y_i$  в (3.24), получаем задачу минимизации кусочно линейной выпуклой функции

$$f(\{x_{ijt}\}) = \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T c_{ijt} x_{ijt} + \sum_{i=1}^m k_i \max_{1 \leq t \leq T} \left\{ 0; d_{it} - \sum_{j=1}^n x_{ijt} \right\} \quad (3.29)$$

при распадающихся ограничениях

$$\sum_{i=1}^m p_{ijt} x_{ijt} = b_{jt}, \quad x_{ijt} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 1, \dots, T, \quad (3.30)$$

каждое из которых геометрически соответствует  $(k_{jt} - 1)$ -мерному симплексу  $S_{jt}$  ( $k_{jt}$  — количество переменных, входящих с ненулевыми коэффициентами в ограничение (3.25)). Наиболее простой алгоритм решения задачи (3.29), (3.30) — субградиентный процесс с проектированием на множество ограничений по схеме

$$x_{k+1} = \text{Пр}_D \{x_k - h_k g_f(x_k)\}, \quad (3.31)$$

где  $D = \prod_{j=1}^n \prod_{t=1}^T S_{jt}$  — прямое произведение симплексов. Как известно, проектирование на прямое произведение выпуклых тел сводится к последовательному проектированию на каждое из них. Задача проектирования точки  $a$  на симплекс сводится к простой задаче квадратичного программирования следующего вида: найти

$$\min_{y \in E^n} \left\{ \|y - a\|^2 : y \geq 0, \quad \sum_{i=1}^n y^i = c, \quad c > 0 \right\},$$

которая легко решается алгоритмом со сложностью  $O(n)$ .

Отметим некоторые особенности процесса (3.31) применительно к нашему случаю. Пусть  $t_i^*(x)$  — одно из значений  $t \in \{1, \dots, T\}$ , на котором достигается максимум в формуле (3.29), либо 0, если

$$\max_{1 \leq t \leq T} \left\{ d_{it} - \sum_{j=1}^n x_{ijt} \right\} < 0.$$

Тогда  $g_f^{ijt}(x)$  —  $(i, j, t)$ -я компонента субградиента  $g_f(x)$  — вычисляется по формуле

$$g_f^{ijt}(x) = \begin{cases} c_{ijk}, & \text{если } t_i^*(x) \neq t, \\ c_{ijk} + k_i, & \text{если } t_i^*(x) = t. \end{cases}$$

Назовем период  $t$   $i$ -критическим, если  $t_i^*(x) = t$ , и  $(\varepsilon, i)$ -критическим, если  $d_{it} - \sum_{j=1}^n x_{ijt}$  отличается от  $\bar{y}_i$  на величину, не превышающую  $\varepsilon$ . Так как скачок  $g_{ijt}(x)$  происходит лишь при изменении  $t_i^*$ , то на каждой итерации из нескольких тысяч составляющих субградиента изменяются лишь несколько десятков. Этот факт можно использовать для более быстрого вычисления субградиента, выделяя лишь те составляющие, которые близки к скачку (с учетом величины шагового множителя). В силу этой же причины большинство составляющих вектора  $x$  также, как правило, остаются неизменными на данном шаге. Мы можем заранее выделять те группы работ, на которых возможно перераспределение технологии их выполнения на ближайших итерациях, и ограничивать свои вычисления только этими работами.

Указанные приемы могут значительно сократить время одной итерации. Это весьма существенно, так как размерность вектора  $x = \{x_{ijk}\}$  велика; поэтому трудно применять ускоряющие процедуры и приходится ограничиваться простым алгоритмом (3.31), который обладает медленной сходимостью и требует для получения приемлемой точности большого числа итераций.

Указанный алгоритм с небольшими модификациями можно применять и в том случае, когда нет четкого разбиения на периоды, а просто задается перечень работ, объемы и интервалы выполнения каждой работы.

**2. Задача о закупке машин и оборудования на долговременный период.** Ставится задача о закупке и использовании в течение длительного периода определенного набора взаимозаменяемых машин и оборудования для производства заданного списка работ в заданных объемах в определенные промежутки времени. Пусть  $t \in \{1, 2, \dots, T\}$  — промежуток времени,  $j$  — индекс работы,  $j = 1, \dots, n$ ,  $i$  — индекс машины (оборудования),  $i = 1, \dots, m$ ;  $x_{ijt}$  — количество машин  $i$ -го типа, выполняющих  $j$ -ю работу в  $t$ -й период времени;  $y_{it}$  — количество машин типа  $i$ , закупаемых к началу периода  $t$ ;  $a_{it}$  — количество старых машин  $i$ -го типа, остающихся в эксплуатации к началу периода  $t$ ;  $p_{ijt}$  — производительность  $i$ -й машины на  $j$ -й работе в  $t$ -й период;  $p_{ijt}^\alpha$  — технологические коэффициенты расходования ресурсов  $\alpha$ -го типа при использовании  $i$ -й машины на  $j$ -й работе в  $t$ -й период;  $b_{jt}$ ,  $d_t^\alpha$  — соответственно объемы работ  $j$ -го типа и ограничения по ресурсам  $\alpha$  в  $t$ -й период;  $c_{ijt}$  — приведенные эксплуатационные затраты, связанные с использованием машины  $i$ -го типа на  $j$ -й работе в  $t$ -й период;  $c_{it}$  — приведенные затраты по закупке  $i$ -й машины перед началом  $t$ -го периода (с учетом остаточной стоимости к концу планируемого промежутка времени (к концу  $T$ -го периода)).

Математическая модель задачи имеет следующий вид: найти

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T c_{ijt} x_{ijt} + \sum_{i=1}^m \sum_{t=1}^T c_{it} y_{it} \right) \quad (3.32)$$

при ограничениях

$$\sum_{i=1}^m p_{ijt} x_{ijt} = b_{jt}, \quad j = 1, \dots, n, \quad t = 1, \dots, T, \quad (3.33)$$

$$\sum_{i=1}^m x_{ijt} \leq a_{it} + \sum_{\tau=1}^t y_{i\tau}, \quad i = 1, \dots, m, \quad t = 1, \dots, T; \quad (3.34)$$

$$\sum_{i=1}^m \sum_{j=1}^n p_{ijt}^{\alpha} x_{ijt} \leq d_i^{\alpha}, \quad \alpha \in A, \quad t = 1, \dots, T, \quad (3.35)$$

где  $A$  — множество видов ресурсов;

$$x_{ijt} \geq 0, \quad y_{it} \geq 0, \quad i = 1, \dots, m, \quad (3.36)$$

$$j = 1, \dots, n, \quad t = 1, \dots, T$$

Для решения этой задачи применим схему декомпозиции по ограничениям. Рассмотрим функцию Лагранжа

$$\begin{aligned} \mathcal{L}(x, y, u) = & \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T c_{ijt} x_{ijt} + \sum_{i=1}^m \sum_{t=1}^T c_{it} y_{it} + \\ & + \sum_{i=1}^m \sum_{t=1}^T u_{it} \left( \sum_{j=1}^n x_{ijt} - a_{it} - \sum_{\tau=1}^t y_{i\tau} \right) + \\ & + \sum_{t=1}^T \sum_{\alpha \in A} u_i^{\alpha} \left( \sum_{i=1}^m \sum_{j=1}^n p_{ijt}^{\alpha} x_{ijt} - d_i^{\alpha} \right) \end{aligned}$$

и внутреннюю задачу определения

$$\Psi(u) = \min_{\{x, y\}} \mathcal{L}(x, y, u) \quad (3.37)$$

при ограничениях

$$\sum_{i=1}^m p_{ijt} x_{ijt} = b_{jt}, \quad j = 1, \dots, n, \quad t = 1, \dots, T, \quad (3.38)$$

$$x_{ijt} \geq 0, \quad y_{it} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 1, \dots, T. \quad (3.39)$$

В выражении для  $\mathcal{L}(x, y, u)$  приведем подобные члены:

$$\begin{aligned} \mathcal{L}(x, y, u) = & \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T \left( c_{ijt} + u_{it} + \sum_{\alpha \in A} u_i^{\alpha} p_{ijt}^{\alpha} \right) x_{ijt} + \\ & + \left( c_{it} - \sum_{\tau=t}^T u_{i\tau} \right) y_{it} - \sum_{i=1}^m \sum_{t=1}^T a_{it} u_{it} - \sum_{t=1}^T \sum_{\alpha \in A} u_i^{\alpha} d_i^{\alpha}. \quad (3.40) \end{aligned}$$

При любом  $u \geq 0$   $\inf_{x,y} \mathcal{L}(x, y, u)$  дает оценку снизу для оптимального значения функции цели, однако нетривиальные оценки, не равные  $-\infty$ , будут получаться для значений  $u$ , удовлетворяющих неравенствам

$$c_{it} - \sum_{\tau=1}^T u_{i\tau} \geq 0, \quad i = 1, \dots, m, \quad t = 1, \dots, T. \quad (3.41)$$

Функция  $\psi(u)$  определена и конечна при выполнении ограничений  $u \geq 0$  и (3.41). В самом деле, при выполнении (3.41) оптимальные значения  $y = \{y_{it}\}$  в задаче (3.37) — (3.39) есть  $y = 0$ . По переменным  $x(u) = \{x_{ijt}(u)\}$  ограничения (3.38) распадаются на однострочные блоки, поэтому  $x(u)$  находится весьма просто путем решения следующего семейства задач: найти

$$\min \sum_{i=1}^m \left( c_{ijt} + u_{it} + \sum_{\alpha \in A} p_{ijt}^{\alpha} u_i^{\alpha} \right) x_{ijt}$$

при ограничениях

$$\sum_{i=1}^m p_{ijt} x_{ijt} = b_{jt}, \quad j = 1, \dots, n, \quad t = 1, \dots, T,$$

$$x_{ijt} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 1, \dots, T.$$

Обозначим  $r_{ijt}(u) = c_{ijt} + u_{it} + \sum_{\alpha \in A} p_{ijt}^{\alpha} u_i^{\alpha}$ , и пусть  $s_{jt}(u) = \min_{1 \leq i \leq m} (r_{ijt}(u)/p_{ijt}(u))$ ,  $i^* = i_j^*(u)$  — минимальный номер, при котором  $r_{i^*jt}/p_{i^*jt} = s_{jt}(u)$ . Тогда

$$x_{ijt}(u) = \begin{cases} b_{jt}/p_{ijt}, & \text{если } i = i_j^*(u), \\ 0, & \text{если } i \neq i_j^*(u). \end{cases} \quad (3.42)$$

Подставив значения  $x_{ijt}(u)$  в (3.40), получаем следующую задачу конечно гладкой оптимизации: найти

$$\max_{u \in \bar{U}} \psi(u) = \max_{u \in \bar{U}} \left( \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T \left( c_{ijt} + u_{it} + \sum_{\alpha \in A} u_i^{\alpha} p_{ijt}^{\alpha} \right) x_{ijt}(u) - \sum_{i=1}^m \sum_{t=1}^T a_{it} u_{it} - \sum_{i=1}^I \sum_{\alpha \in A} d_i^{\alpha} u_i^{\alpha} \right), \quad (3.43)$$

где

$$\bar{U} = \left\{ u : u \geq 0, \quad c_{it} - \sum_{\tau=1}^T u_{i\tau} \geq 0, \quad \tau = 1, \dots, T, \quad i = 1, \dots, m \right\}. \quad (3.44)$$

Для того чтобы учесть нетривиальные ограничения в (3.41), используем метод негладких штрафных функций, при этом задача (3.43)

сводится к следующей: найти

$$\max_{u \geq 0} \left[ \sum_{i=1}^m \sum_{t=1}^T u_{it} \left( \sum_{j=1}^n x_{ijt}(u) - a_{it} \right) + \right. \\ \left. + \sum_{t=1}^T \sum_{\alpha \in A} u_t^\alpha \left( \sum_{j=1}^n x_{ijt}(u) p_{ijt}^\alpha - d_t^\alpha \right) + \right. \\ \left. + \sum_{i=1}^m \sum_{t=1}^T s_{it} \left( c_{it} - \sum_{\tau=1}^T u_{i\tau} \right)^+ \right], \quad (3.45)$$

где  $s_{it}$  — достаточно большие положительные числа (штрафные множители) (см. теорему 1.11).

Для решения задачи (3.45) с успехом применяется  $r$ -алгоритм, в котором ограничения  $u \geq 0$  учитываются по схеме отражения, как описано в [15].

Математическая модель, подобная (3.32) — (3.36), использовалась для решения задачи об оптимальной поставке пассажирских и грузовых самолетов Министерству гражданской авиации при долгосрочном планировании (5—10 лет). Подробное описание модели, алгоритма решения этой задачи и анализ результатов реальных расчетов содержится в [36].

**3. Динамическая транспортная задача.** При решении вопросов материально-технического снабжения большое значение имеет постоянство связей между поставщиками и потребителями в течение достаточно длительного периода. В условиях неравномерности потребления и (или) производства планирование распределения продукции между потребителями с учетом постоянства связей наталкивается на серьезные трудности: необходимость создания запасов продукции, невозможность независимого планирования распределения продукции по отдельным небольшим периодам. Требование постоянства связей при строгом его учете вносит в математическую модель распределительной задачи булевы переменные, что весьма затрудняет процесс получения оптимального плана. В работе [7] была предложена специальная модель линейного программирования (без булевых переменных), которая достаточно хорошо соответствует реальным требованиям планирования поставок однородной продукции во времени.

Имеется  $m$  поставщиков (предприятий) и  $n$  потребителей (строек). Планируемый период разбит на  $T$  интервалов. Для краткости интервал планирования будем называть кварталом. Заданы:  $a_{it}$  — объем продукции, производимой  $i$ -м предприятием за квартал  $t$ ,  $t =$

$$= 1, \dots, T, a_i = \sum_{t=1}^T a_{it} — \text{объем продукции, производимой } i\text{-м пред-}$$

приятием за планируемый период,  $b_{jt}$  — объем спроса на продукцию  $j$ -го потребителя за квартал  $t$ ,  $t = 1, \dots, T$ ,  $b_j$  — потребность  $j$ -й

стройки за планируемый период,  $b_j = \sum_{t=1}^T b_{jt}$ ,  $c_{ij}$  — стоимость перевозки

единицы продукции от  $i$ -го предприятия к  $j$ -му потребителю. Транспортные издержки  $c_{ij}$  принимаются независимыми от времени доставки. Предполагается, что общее производство и потребление сбалансированы:

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

однако поквартальный баланс, вообще говоря, не имеет места:  $\sum_{i=1}^m a_{it}$

не обязательно равна  $\sum_{j=1}^n b_{jt}$  ( $t = 1, \dots, T$ ).

На интуитивном уровне задача заключается в определении такого распределения продукции между потребителями, чтобы при этом выполнялись условия:

а) для большинства предприятий или, по крайней мере, для предприятий с большим приоритетом удовлетворяется их потребность по кварталам;

б) объем преждевременно поставленной продукции должен быть по возможности меньшим (это условие можно интерпретировать как минимизацию затрат на хранение преждевременно произведенной продукции);

в) транспортные издержки минимальны по сравнению с другими распределениями, удовлетворяющими первым двум условиям;

г) потребители в течение всего планируемого периода должны по возможности прикрепляться к одному поставщику.

Введем обозначения:  $\beta_{jt}$  — объем продукции, являющейся спросом  $j$ -го потребителя за первые  $t$  кварталов,

$$\beta_{jt} = \sum_{\tau=1}^t b_{j\tau}, \quad t = 1, \dots, T;$$

$\alpha_{it}$  — объем продукции, производимой  $i$ -м предприятием за первые  $t$  кварталов,

$$\alpha_{it} = \sum_{\tau=1}^t a_{i\tau}, \quad t = 1, \dots, T;$$

$x_{ijt}$  — объем продукции, поставляемой  $i$ -м предприятием  $j$ -му потребителю за первые  $t$  кварталов. Предположим, что все  $\beta_{jt} \neq 0$ , однако отметим, что это ограничение, как будет видно из дальнейшего, не является существенным и введено лишь для простоты изложения. Окончательные результаты справедливы для случая  $\beta_{jt} \geq 0$ . Величину  $x_{ijt}/\beta_{jt}$  назовем степенью обеспеченности  $i$ -го потребителя  $j$ -м предприятием за первые  $t$  кварталов.

При построении математической модели дополним постановку задачи, не изменяя существенно ее экономической сути, следующими условиями:

1) преждевременно произведенная продукция предприятия хранится на его складах, т. е. потребители не получают излишки

продукции:

$$\sum_{i=1}^m x_{ijt} \leq \beta_{jt}, \quad t = 1, \dots, T-1;$$

2) степень обеспеченности потребителя не убывает со временем:

$$x_{ijt}/\beta_{jt} \geq x_{ij,t-1}/\beta_{j,t-1}, \quad t = 2, \dots, T. \quad (3.46)$$

Введем штрафные коэффициенты:  $R_{jt}$  — величина штрафа за недопоставку единицы продукции  $j$ -му потребителю в конце  $t$ -го квартала;  $l_{it}$  — величина штрафа за наличие единицы нераспределенной продукции на  $i$ -м предприятии в конце  $t$ -го квартала. Через  $w_{it}$  обозначим объем нераспределенной продукции на  $i$ -м предприятии в конце  $t$ -го квартала.

Математическая модель поставленной задачи описывается следующей задачей линейного программирования: найти

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ijt} + \sum_{i=1}^m \sum_{t=1}^{T-1} R_{jt} \left( \beta_{jt} - \sum_{i=1}^m x_{ijt} \right) + \sum_{i=1}^m \sum_{t=1}^{T-1} l_{it} w_{it} \right) \quad (3.47)$$

при ограничениях

$$\sum_{i=1}^m x_{ijt} = b_j, \quad j = 1, \dots, n, \quad (3.48)$$

$$\sum_{j=1}^n x_{ijt} = a_i, \quad i = 1, \dots, m, \quad (3.49)$$

$$\beta_{j,t-1} x_{ijt} \geq \beta_{jt} x_{ij,t-1}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 2, \dots, T, \quad (3.50)$$

$$\sum_{i=1}^n x_{ijt} + w_{it} = \alpha_{it}, \quad i = 1, \dots, m, \quad t = 1, \dots, T-1, \quad (3.51)$$

$$\sum_{i=1}^m x_{ijt} \leq \beta_{jt}, \quad j = 1, \dots, n, \quad t = 1, \dots, T-1, \quad (3.52)$$

$$x_{ijt} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 1, \dots, T, \quad (3.53)$$

$$w_{it} \geq 0, \quad i = 1, \dots, m, \quad t = 1, \dots, T-1. \quad (3.54)$$

З а м е ч а н и я. 1. Ограничения (3.50), имеющие смысл и для  $\beta_{jt} = 0$ , соответствуют условию (3.46).

2. Хотя естественное условие  $x_{ij,t+1} \geq x_{ijt}$  не введено формально в список ограничений задачи, оно является следствием ограничений (3.50), (3.52) и (3.53), поскольку  $\beta_{j,t+1} \geq \beta_{jt}$ .

3. Отметим, что условие (3.50) запрещает прерывать обслуживание потребителя данным поставщиком, что «способствует» выполнению условия г). Более точная формулировка требования г) приводила бы к задаче целочисленного программирования, получение удовлетворительного решения которой было бы весьма проблематичным в случае большой размерности задачи.

Опишем метод решения поставленной задачи. Решение задачи (3.47) — (3.54) будет проводиться в два этапа. На первом этапе определяется схема оптимального прикрепления потребителей к поставщикам, на втором — объемы поквартальных поставок  $x_{ijt}$ .

Нахождение схемы оптимального прикрепления потребителей сводится, как и в случае классической транспортной задачи, к решению двойственной задачи.

**З а м е ч а н и е.** Под схемой оптимального прикрепления мы понимаем выделение пар индексов  $(i, j)$ , для которых  $x_{ijt} \geq 0$ .

Прежде чем переходить к двойственной задаче, запишем задачу (3.47) — (3.54) в другой эквивалентной форме: найти

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n x_{ijt} c_{ij} + \sum_{j=1}^n \sum_{t=1}^{T-1} \left( \sum_{\tau=1}^t \beta_{j\tau} R_{j\tau} \right) \left( \sum_{i=1}^m y_{ijt} \right) + \sum_{i=1}^m \sum_{t=1}^{T-1} l_{it} w_{it} \right) \quad (3.55)$$

при ограничениях

$$\sum_{i=1}^m x_{ijt} = b_j, \quad j = 1, \dots, n, \quad (3.56)$$

$$\sum_{j=1}^n x_{ijt} = a_i, \quad i = 1, \dots, m, \quad (3.57)$$

$$\sum_{j=1}^n \beta_{jt} \sum_{\tau=0}^{t-1} y_{ijt} + w_{it} = \alpha_{it}, \quad i = 1, \dots, m, \quad t = 1, \dots, T-1, \quad (3.58)$$

$$-\frac{1}{b_j} x_{ijt} + \sum_{\tau=0}^{T-1} y_{i\tau} = 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (3.59)$$

$$x_{ijt} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (3.60)$$

$$y_{ijt} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 0, \dots, T-1, \quad (3.61)$$

$$w_{it} \geq 0, \quad i = 1, \dots, m, \quad t = 1, \dots, T-1. \quad (3.62)$$

Нетрудно показать, что задача (3.55) — (3.62) эквивалентна задаче (3.47) — (3.54), причем решение  $\{x_{ijt}\}$  задачи (3.47) — (3.54) определяется формулой

$$x_{ijt} = \beta_{jt} \sum_{\tau=0}^{t-1} y_{i\tau}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 1, \dots, T,$$

где  $\{y_{ijt}\}$  — решение задачи (3.55) — (3.62).

Покажем, например, справедливость равенства

$$\sum_{t=1}^{T-1} \left( \sum_{\tau=1}^t \beta_{j\tau} R_{j\tau} \right) \left( \sum_{i=1}^m y_{ijt} \right) = \sum_{\tau=1}^{T-1} R_{j\tau} (\beta_{j\tau} - \tilde{x}_{j\tau}),$$

где

$$\tilde{x}_{j\tau} = \sum_{i=1}^m x_{ijt}.$$



Положив

$$r_{jt} = \sum_{\tau=1}^t \beta_{j\tau} R_{j\tau}, \quad t = 1, \dots, T-1, \quad j = 1, \dots, m,$$

и учитывая (3.59), получаем

$$\begin{aligned} \sum_{t=1}^{T-1} \left( \sum_{\tau=1}^t \beta_{j\tau} R_{j\tau} \right) \left( \sum_{i=1}^m y_{ijt} \right) &= \sum_{t=1}^{T-1} r_{jt} \sum_{i=1}^m \left( \frac{x_{ijt,t+1}}{\beta_{j,t+1}} - \frac{x_{ijt}}{\beta_{jt}} \right) = \\ &= \sum_{t=1}^{T-1} r_{jt} \left( \frac{\tilde{x}_{j,t,t+1} - \beta_{j,t,t+1}}{\beta_{j,t,t+1}} - \frac{\tilde{x}_{jt} - \beta_{jt}}{\beta_{jt}} \right) = \\ &= \sum_{t=1}^{T-1} r_{jt} \left( \frac{\beta_{jt} - \tilde{x}_{jt}}{\beta_{jt}} \right) - \sum_{t=1}^{T-1} r_{jt} \left( \frac{\beta_{j,t,t+1} - \tilde{x}_{j,t,t+1}}{\beta_{j,t,t+1}} \right) = \\ &= \sum_{t=1}^{T-1} r_{jt} \left( \frac{\beta_{jt} - \tilde{x}_{jt}}{\beta_{jt}} \right) - \sum_{t=1}^{T-1} (r_{i,t+1} - \beta_{i,t+1} R_{i,t+1}) \frac{\beta_{j,t+1} - \tilde{x}_{j,t+1}}{\beta_{j,t+1}} = \\ &= r_{j1} \frac{\beta_{j1} - \tilde{x}_{j1}}{\beta_{j1}} + \sum_{t=2}^{T-1} R_{jt} (\beta_{jt} - \tilde{x}_{jt}) = \sum_{t=1}^{T-1} R_{jt} (\beta_{jt} - \tilde{x}_{jt}), \end{aligned}$$

что и требовалось показать.

Обозначим через  $v_j$ ,  $u_{iT}$ ,  $u_{it}$ ,  $z_{ij}$  двойственные переменные, отвечающие соответственно ограничениям (3.56) — (3.59). Тогда задача, двойственная к (3.55) — (3.62), будет состоять в следующем: найти

$$\max \left( \sum_{j=1}^n b_j v_j - \sum_{i=1}^m \sum_{t=1}^T \alpha_{it} u_{it} \right) \quad (3.63)$$

при ограничениях

$$v_j \leq c_{ij} + z_{ij}/b_j + u_{iT}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (3.64)$$

$$z_{ij} \leq \sum_{t=1}^{T-1} \beta_{jt} u_{it}, \quad (3.65)$$

$$z_{ij} \leq \sum_{\tau=1}^t \beta_{j\tau} R_{j\tau} + \sum_{t=t+1}^{T-1} \beta_{jt} u_{i\tau}, \quad (3.66)$$

$$i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 1, \dots, T-2,$$

$$z_{ij} \leq \sum_{t=1}^{T-1} \beta_{jt} R_{jt}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (3.67)$$

$$u_{it} \geq -l_{it}, \quad i = 1, \dots, m, \quad t = 1, \dots, T-1 \quad (3.68)$$

Пусть  $v_j^*$ ,  $u_{it}^*$ ,  $z_{ij}^*$  — решение задачи (3.63) — (3.68). Тогда, очевидно, выполняется соотношение

$$v_j^* = \min_{1 \leq i \leq m} \left( c_{ij} + \frac{1}{b_j} z_{ij}^* + u_{iT}^* \right). \quad (3.69)$$

Обозначим через  $I^*(j)$  множество индексов  $i$ , при которых достигается минимум в правой части формулы (3.69). Поскольку переменные  $x_{ijt}$  задачи (3.47) — (3.54) являются двойственными оценками для системы ограничений (3.64), то  $x_{ijt} > 0$  лишь для  $i \in I^*(j)$ . Поэтому  $I^*(j)$  является множеством номеров предприятий, к которым прикрепляется  $j$ -й потребитель. Таким образом, для определения схемы оптимального прикрепления потребителей к пунктам производства достаточно решить двойственную задачу (3.63) — (3.68). Отметим, что в случае  $R_{jt} = 0$ ,  $l_{it} = 0$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ ,  $t = 1, \dots, T-1$ , (3.63) — (3.68) является двойственной к классической транспортной задаче, определяемой параметрами  $c_{ij}$ ,  $a_i$ ,  $b_j$ .

Для решения задачи (3.63) — (3.68) применим схему декомпозиции по множеству переменных. Зафиксировав переменные  $u_{it}$ , приходим к задаче: найти

$$\max \sum_{j=1}^n b_j v_j \quad (3.70)$$

при ограничениях

$$\begin{aligned} v_j &\leq c_{ij} + z_{ij}/b_j + u_{iT}, & i = 1, \dots, m, & j = 1, \dots, n, \\ z_{ij} &\leq r_{ijt}(u), & i = 1, \dots, m, & j = 1, \dots, n, & t = 0, \dots, T-1, \end{aligned}$$

где

$$\begin{aligned} r_{ij0}(u) &= \sum_{\tau=1}^{T-1} \beta_{j\tau} u_{i\tau}, \\ r_{ijt}(u) &= \sum_{\tau=1}^t \beta_{j\tau} R_{j\tau} + \sum_{\tau=t+1}^{T-1} \beta_{j\tau} u_{i\tau}, & t = 1, \dots, T-2, \\ r_{i,T-1}(u) &= \sum_{\tau=1}^{T-1} \beta_{j\tau} R_{j\tau}. \end{aligned}$$

Решение задачи (3.70) определяется, очевидно, формулами

$$z_{ij}(u) = \min_{0 \leq t \leq T-1} r_{ijt}(u), \quad (3.71)$$

$$v_j(u) = \min_{1 \leq i \leq m} \left( c_{ij} + \frac{1}{b_j} z_{ij}(u) + u_{iT} \right). \quad (3.72)$$

Таким образом, задача (3.63) — (3.68) сводится к задаче максимизации кусочно линейной вогнутой функции  $\tilde{F}(u)$ : найти

$$\max \tilde{F}(u) = \max \left( \sum_{j=1}^n b_j v_j(u) - \sum_{i=1}^m \sum_{t=1}^T \alpha_{it} u_{it} \right) \quad (3.73)$$

при ограничениях

$$u_{it} \geq -l_{it}, \quad i = 1, \dots, m, \quad t = 1, \dots, T-1. \quad (3.74)$$

Рассмотрим также задачу безусловной максимизации кусочно линейной вогнутой функции  $F(u)$ : найти

$$\max_u F(u), \quad (3.75)$$

где

$$F(u) = \sum_{j=1}^n b_j v_j(u) - \sum_{i=1}^m \sum_{t=1}^{T-1} \alpha_{it} \max\{u_{it}, -l_{it}\} - \sum_{i=1}^m \alpha_{iT} u_{iT},$$

а  $v_j(u)$  по-прежнему определяется формулами (3.71), (3.72).

Пусть  $u^*$  — решение задачи (3.75). Тогда нетрудно видеть, что решение  $u^*$  задачи (3.73), (3.74) определяется формулами

$$\tilde{u}_{it}^* = \max\{u_{it}^*, -l_{it}\}, \quad t = 1, \dots, T-1, \quad \tilde{u}_{iT}^* = u_{iT}^*.$$

Итак, задача определения схемы прикрепления потребителей к пунктам производства сводится к задаче (3.75) безусловной максимизации кусочно линейной вогнутой функции, для решения которой можно применить одну из модификаций метода градиентного типа. Как показали экспериментальные расчеты,  $r$ -алгоритм является эффективным средством решения такого рода задач.

Задача определение поставок  $\{x_{ijt}\}$  сводится к проблеме определения решения прямой задачи (3.47) — (3.54) по найденному решению двойственной задачи (3.63) — (3.68).

### § 5. Применение $\varepsilon$ -квадратичного сглаживания при решении задачи линейного программирования путем декомпозиции по ограничениям

Запишем задачу линейного программирования в следующем виде: найти

$$\min(c, x) \quad (3.76)$$

при ограничениях

$$x \in X, \quad (3.77)$$

$$Ax \leq b, \quad (3.78)$$

где  $x, c \in E^n$ ,  $b \in E^m$ ,  $A$  — матрица размеров  $m \times n$ ,  $X$  — некоторое выпуклое полиэдральное множество. Здесь сознательно выделены в отдельную группу ограничения (3.77), хотя они также представимы в виде системы линейных неравенств. Особенность их в том, что трудоемкость решения задачи (3.76), (3.77) сравнительно невелика. Как правило, это связано с тем, что система неравенств, задающих  $X$ , имеет блочно-диагональный вид; при этом отдельные блоки сравнительно малы, хотя их число может быть довольно большим. В таких условиях задача (3.76), (3.77) распадается на ряд независимых

подзадач, что и обеспечивает простоту ее решения. В отличие от «блочных» ограничений (3.77), ограничения (3.78) лишены специфики: это линейные неравенства общего вида, связывающие переменные из различных блоков, так называемые связывающие ограничения. Обозначим через  $\bar{x}$  некоторое оптимальное решение задачи (3.76) — (3.78) (предполагается, что множество оптимальных решений не пусто),  $f^* = (c, \bar{x})$ . Применим к задаче (3.76) — (3.78) схему декомпозиции по ограничениям.

Для аргумента  $u \geq 0$ ,  $u \in E^m$ , рассмотрим функцию

$$\varphi(u) = \min_{x \in X} \{(c, x) + (u, Ax - b)\}. \quad (3.79)$$

Функция  $\varphi(u)$  вогнута, кусочно линейна, границы кусков линейности функции  $\varphi(u)$  состоят из точек  $u \in E^m$ , для которых значения

$$x(u) = \arg \min_{x \in X} \{(c, x) + (u, Ax - b)\} \quad (3.80)$$

определяются неоднозначно.

Из доказанных выше результатов следует, что

$$f^* = (c, \bar{x}) = \varphi(u^*),$$

где  $u^* = \arg \max_{u \geq 0} \varphi(u)$ . Кроме того, компоненты вектора  $u^*$  совпадают с той частью оптимальных двойственных переменных задачи (3.76) — (3.78), которые соответствуют ограничениям (3.78). Таким образом, нахождение  $f^*$  сводится к максимизации по  $u \geq 0$  вогнутой негладкой функции  $\varphi(u)$ . Субградиент функции  $\varphi$  в точке  $u$  вычисляется по формулам

$$g_\varphi(u) = Ax(u) - b, \quad (3.81)$$

где  $x(u)$  — некоторое решение (3.80). Отметим, что трудоемкость вычисления  $\varphi(u)$  и  $g_\varphi(u)$  при блочно-диагональной структуре  $X$  сравнительно невелика.

Возможность сведения задачи (3.76) — (3.78) к задаче максимизации вогнутой функции небольшой размерности особенно ценна, когда задача (3.76) — (3.78) имеет большой объем: количество переменных и ограничений (в том числе и определяющих множество  $X$ ) порядка  $10^5$ — $10^6$ . В этой ситуации применение конечных алгоритмов типа симплекс-метода наталкивается на трудности, связанные со значительным увеличением числа итераций и потерей точности вычислений. А в соответствующей негладкой задаче количество переменных равно  $m$  — количеству связывающих ограничений, которое даже в больших практических задачах редко превышает  $10^3$ . При заданной точности решения число итераций эффективных алгоритмов негладкой оптимизации (например,  $r$ -алгоритма) практически линейно зависит от  $m$ . Кроме того, эти алгоритмы менее чувствительны к накоплению ошибок.

Решив с высокой точностью задачу (3.80), мы получим хорошее приближение к  $f^*$ . Однако  $x(u^*)$  находится неоднозначно, лишь часть значений  $x(u^*)$  удовлетворяет ограничениям (3.78), т. е. яв-

ляется допустимой в задаче (3.76) — (3.78). Нахождение допустимого  $x(u^*)$  при больших размерностях не является тривиальной задачей. Для ее решения можно использовать несколько подходов.

Первый подход основан на теории двойственности в линейном программировании. Он сводится к проверке выполнения (с некоторыми заданными порогами точности) соотношений дополняющей нежесткости. Это позволяет редуцировать задачу (3.76) — (3.78), т. е. выделить несущественные ограничения и переменные, заведомо равные нулю в оптимальном решении. Если полученная редуцированная задача имеет сравнительно небольшой объем, для ее решения можно применить симплекс-метод или подходящую для данной размерности другую эффективную процедуру.

Другой подход связан с применением процедур усреднения для последовательности  $\{x(u_k)\}_{k=1}^{\infty}$  [7]. Если объем исходной задачи велик, оба эти подхода обнаруживают существенные недостатки. При первом подходе объем редуцированной задачи может оказаться еще слишком большим для успешного применения алгоритма с конечным числом итераций. При втором подходе удовлетворительное приближение к  $\bar{x}$  достигается, если для усреднения берется большое число членов последовательности  $\{x(u_k)\}$ ; что требует больших вычислительных затрат.

Следующий способ нахождения  $\bar{x}$  основан на том, чтобы сделать целевую функцию (3.76) строго выпуклой. В этом случае  $x(u^*)$  будет определяться однозначно и автоматически будет допустимым.

Внесем в (3.76) регуляризующую квадратичную добавку. Получим возмущенную задачу: найти

$$\min \left( (c, x) + \frac{\varepsilon}{2} (x, x) \right), \quad \varepsilon > 0, \quad (3.82)$$

при ограничениях

$$x \in X, \quad (3.83)$$

$$Ax - b \leq 0. \quad (3.84)$$

Обозначим через  $x^*(\varepsilon)$  оптимальное решение задачи (3.82) — (3.84). Отметим, что из существования  $\bar{x}$  следует существование  $x^*(\varepsilon)$  и его единственность. Решение задачи (3.82) — (3.84) сводится к максимизации по  $u \geq 0$  строго вогнутой функции

$$\varphi_{\varepsilon}(u) = \min_{x \in X} \left( (c, x) + \frac{\varepsilon}{2} (x, x) + (u, Ax - b) \right).$$

При этом если положить

$$x_{\varepsilon}(u) = \arg \min_{x \in X} \left[ (c, x) + \frac{\varepsilon}{2} (x, x) + (u, Ax - b) \right],$$

$$u_{\varepsilon}^* = \arg \max_{u \geq 0} \varphi_{\varepsilon}(u),$$

то  $x^*(\varepsilon) = x_{\varepsilon}(u_{\varepsilon}^*)$ .

Оказывается, что при определенных условиях  $x^*(\varepsilon)$  совпадает или в определенном смысле близок к  $\bar{x}$ . Для исследования этого вопроса запишем задачи (3.76) — (3.78) и (3.82) — (3.84) в следующем

виде: найти

$$\min(c, x)$$

при ограничениях

$$(a_i, x) - b_i \leq 0, \quad i = 1, \dots, N, \quad x \in E^n;$$

найти

$$\min\left((c, x) + \frac{\varepsilon}{2}(x, x)\right)$$

при ограничениях

$$(a_i, x) - b_i \leq 0, \quad i = 1, \dots, N, \quad x \in E^n$$

(здесь  $a_i \in E^n$  и  $N > n$ , так как вместо ограничений  $x \in X$  в явном виде выписана система линейных неравенств, определяющих  $X$ ). Пусть оптимальное решение  $\bar{x}$  задачи (3.76) — (3.78) не вырождено и единственно. Без ограничения общности можно считать, что векторы  $a_1, \dots, a_n$  линейно независимы и

$$(a_i, \bar{x}) = b_i, \quad i = 1, \dots, n, \quad (3.85)$$

$$(a_i, \bar{x}) < b_i, \quad i > n. \quad (3.86)$$

Пусть  $\bar{u} = \{\bar{u}_1, \dots, \bar{u}_n, \dots, \bar{u}_N\} \in E^N$  — вектор оптимальных двойственных переменных задачи (3.76) — (3.78). Из предположений о свойствах  $\bar{x}$  следует, что

$$\bar{u}_i > 0, \quad i = 1, \dots, n, \quad \bar{u}_i = 0, \quad i = n + 1, \dots, N. \quad (3.87)$$

Введем в рассмотрение величины

$$D = \det \Gamma(a_1, a_2, \dots, a_n) > 0;$$

$$D_a^i = \det \Gamma(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n),$$

$$D_b^i = \left\| \begin{array}{c|c} \Gamma(a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_n) & \begin{matrix} b_1 \\ \vdots \\ b_n \end{matrix} \\ \hline (a_i, a_1) \quad (a_i, a_2) \quad \dots \quad (a_i, a_n) & b_i \end{array} \right\|,$$

где  $\Gamma(r_1, \dots, r_p)$  — матрица Грама векторов  $(r_1, \dots, r_p)$ . Соотношение между  $x$  и  $x^*(\varepsilon)$  описывает

Теорема 3.1 [22]. Пусть  $\varepsilon_0$  — максимальное (не обязательно конечное) решение системы неравенств

$$\varepsilon D_b^i \leq \bar{u}_i D, \quad i = 1, \dots, n. \quad (3.88)$$

Тогда:

- а) при  $\varepsilon \in [0, \varepsilon_0]$   $x^*(\varepsilon)$  не зависит от  $\varepsilon$  и  $x^*(\varepsilon) = \bar{x}$ ;
- б) если  $\varepsilon_0 < \infty$  и номер  $i_0$  такой, что  $\varepsilon_0 D_b^{i_0} = \bar{u}_{i_0} D$ , единственный, то существует (не обязательно конечное)  $\varepsilon_1 > \varepsilon_0$  такое, что при  $\varepsilon \in [\varepsilon_0, \varepsilon_1]$

$$(c, x^*(\varepsilon)) - (c, \bar{x}) = \bar{u}_{i_0} D_b^{i_0} / D_0^{i_0} (1 - \varepsilon_0 / \varepsilon).$$

Доказательство. Из (3.85) — (3.87) в соответствии с теоремой Куна — Таккера имеем

$$-c = \sum_{i=1}^n \bar{u}_i a_i, \quad \bar{u}_i > 0, \quad i = 1, \dots, n,$$

г. е.

$$-c \in \text{int } K(a_1, \dots, a_n), \quad (3.89)$$

где  $K(a_1, \dots, a_n)$  — конус, определяемый векторами  $a_1, \dots, a_n$ . Из (3.89) вытекает, что существует  $\varepsilon_0 > 0$  такое, что при  $\varepsilon \in [0, \varepsilon_0]$  справедливо включение

$$-(c + \varepsilon \bar{x}) \in K(a_1, \dots, a_n). \quad (3.90)$$

Последнее означает, что  $\bar{x}$  является оптимальным решением задачи (3.82) — (3.84). Таким образом, при  $\varepsilon \in [0, \varepsilon_0]$   $x^*(\varepsilon)$  не зависит от  $\varepsilon$ ,  $x^*(\varepsilon) = \bar{x}$ . Величина  $\varepsilon_0$  может быть найдена как максимальное  $\varepsilon$ , при котором выполняется включение (3.90). Для явного описания конуса  $K(a_1, \dots, a_n)$  воспользуемся следующим утверждением.

Л е м м а 3.1. Пусть векторы  $a_1, a_2, \dots, a_l$  из  $E^n$  линейно независимы. Положим (см. с. 80)

$$p_i = -N(a_i, L(a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_l)), \quad i = 1, \dots, l, \quad (3.91)$$

тогда имеет место представление

$$K(a_1, \dots, a_l) = \{y: y \in L(a_1, \dots, a_l), (p_i, y) \leq 0, i = 1 \dots l\}.$$

Доказательство. Обозначим

$$Y = \{y: y \in L(a_1, \dots, a_l), (p_i, y) \leq 0, i = 1, \dots, l\}.$$

Рассмотрим произвольный вектор  $y \in Y$ . Этот вектор однозначно представим в виде  $y = \sum_{j=1}^l \lambda_j a_j$ . При  $i = 1, \dots, l$  с учетом (3.91), (2.84)

имеем

$$0 \geq (p_i, y) = \sum_{j=1}^l \lambda_j (p_i, a_j) = \lambda_i (p_i, a_i) = -\lambda_i \frac{|\Gamma(a_1, \dots, a_l)|}{|\Gamma(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_l)|}.$$

Отсюда  $\lambda_i \geq 0$ ,  $i = 1, \dots, l$ , т. е.  $y \in K(a_1, \dots, a_l)$ . Следовательно,  $Y \subseteq K(a_1, \dots, a_l)$ . Обратное включение проверяется непосредственно. Лемма доказана.

Продолжим доказательство теоремы. Лемма 3.1 позволяет утверждать, что  $\varepsilon_0$  может быть найдено как максимальное решение системы неравенств

$$-(p_i, c + \varepsilon \bar{x}) \leq 0, \quad i = 1, \dots, n. \quad (3.92)$$

Имеем при  $i = 1, \dots, n$ :

$$(p_i, c) = - \frac{\left| \begin{array}{c} \Gamma(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n) \\ \hline (a_i, a_1) \dots (a_i, a_{i-1}) (a_i, a_{i+1}) \dots (a_i, a_n) \end{array} \right|}{|\Gamma(a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_n)|} \begin{array}{c} (a_i, c) \\ \vdots \\ (a_i, c) \end{array}. \quad (3.93)$$

Подставляя сюда выражение для  $c$  и разлагая детерминант в числителе в сумму  $n$  детерминантов (из которых  $(n - 1)$  равны 0), получим:

$$(p_i, c) = \frac{\bar{u}_i |\Gamma(a_1, \dots, a_n)|}{|\Gamma(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)|} = \bar{u}_i \frac{D}{D_a^i}, \quad i = 1, \dots, n. \quad (3.94)$$

Далее, при  $i = 1, \dots, n$  аналогично вычисляем  $-(p_i, \bar{x})$ . С учетом (3.85) имеем:

$$-(p_i, \bar{x}) = D_b^i / D_a^i. \quad (3.95)$$

Таким образом, неравенство (3.92) эквивалентно неравенству  $\varepsilon D_b^i \leq \bar{u}_i D$ ,  $i = 1, \dots, n$ , что завершает доказательство пункта а) теоремы.

Пусть  $\varepsilon_0$  оказалось конечным, а номер  $i_0$  такой, что  $\varepsilon_0 D_b^{i_0} = \bar{u}_{i_0} D$ , — единственным. Без ограничения общности можно считать, что  $i_0 = n$ . Это означает, что

$$-(p_i, c + \varepsilon_0 \bar{x}) < 0, \quad i = 1, \dots, n - 1; \quad -(p_n, c + \varepsilon_0 \bar{x}) = 0. \quad (3.96)$$

Рассмотрим точки вида  $x(\alpha) = \bar{x} + \alpha p_n / \|p_n\|$ . С учетом (3.91) — (3.93) имеем:

$$\begin{aligned} (a_i, x(\alpha)) - b_i &= (\alpha / \|p_n\|) (a_i, p_n) = d_i, \\ d_i &= 0, \quad i = 1, \dots, n - 1, \quad d_n < 0. \end{aligned}$$

Из (3.86) ясно, что существует (не обязательно конечное)  $\bar{\alpha} > 0$  такое, что при  $\alpha \in [0, \bar{\alpha})$

$$(a_i, x(\alpha)) - b_i = ((a_i, \bar{x}) - b_i) + (\alpha / \|p_n\|) (a_i, p_n) < 0, \quad i > n.$$

Таким образом, при  $\alpha \in [0, \bar{\alpha})$  точки вида  $x(\alpha)$  принадлежат допустимой области задач (3.76) — (3.78) и (3.82) — (3.84).

Введем в рассмотрение функцию  $\alpha(\varepsilon)$ :

$$\alpha(\varepsilon) = -\frac{1}{\varepsilon} (p_n / \|p_n\|, c + \varepsilon \bar{x}).$$

Ее производная

$$\alpha'(\varepsilon) = \frac{1}{\varepsilon^2} \frac{(p_n, c)}{\|p_n\|} > 0.$$

Значит,  $\alpha(\varepsilon)$  монотонно возрастает. Отсюда  $\alpha(\varepsilon) = 0$ , если  $\varepsilon = \varepsilon_0$ ,  $\alpha(\varepsilon) > 0$  при  $\varepsilon > \varepsilon_0$ . Положим

$$x_\varepsilon = \bar{x} + \alpha(\varepsilon) p_n / \|p_n\| = \bar{x} - \frac{1}{\varepsilon} (p_n, c + \varepsilon \bar{x}) p_n / \|p_n\|^2.$$

Покажем, что для задачи (3.82) — (3.84) при  $\varepsilon \in [\varepsilon_0, \varepsilon_1)$   $x_\varepsilon$  является оптимальным решением. Пусть  $\bar{\varepsilon} > 0$  таково, что  $\alpha(\bar{\varepsilon}) = \bar{\alpha}$ . Из определения  $\bar{\alpha}$  следует, что при  $\varepsilon \in [\varepsilon_0, \bar{\varepsilon})$   $x(\varepsilon)$  является допустимым решением задачи (3.82) — (3.84), при этом

$$(a_i, x_\varepsilon) - b_i = 0, \quad i = 1, \dots, n - 1; \quad (a_i, x_\varepsilon) - b_i < 0, \quad i \geq n.$$



Далее имеем:

$$-(p_n, c + \varepsilon x_\varepsilon) = -(p_n, c + \varepsilon \bar{x} - (p_n + \varepsilon \bar{x}) p_n / \|p_n\|^2) = 0.$$

Отсюда  $-(c + \varepsilon x_\varepsilon) \in L(a_1, \dots, a_{n-1})$ . Далее, при  $i = 1, \dots, n-1$

$$-(p_i, c + \varepsilon x_\varepsilon) = -(p_i, c + \varepsilon_0 \bar{x}) - (\varepsilon - \varepsilon_0) ((p_i, \bar{x}) + (p_i, p_n) (p_n, \bar{x}) / \|p_n\|^2).$$

Первое слагаемое здесь строго отрицательно, поэтому существует  $\varepsilon > \varepsilon_0$  такое, что при  $\varepsilon \in [\varepsilon_0, \varepsilon)$  выполняется неравенство

$$-(p_i, c + \varepsilon x_\varepsilon) \leq 0, \quad i = 1, \dots, n-1.$$

Лемма 3.1 позволяет заключить, что при  $\varepsilon \in [\varepsilon_0, \varepsilon) - (c + \varepsilon x_\varepsilon) \in K(a_1, \dots, a_{n-1})$ . Полагая  $\varepsilon_1 = \min\{\varepsilon, \varepsilon^2\}$ , получим, что при  $\varepsilon \in [\varepsilon_0, \varepsilon_1)$   $x_\varepsilon$  удовлетворяет условиям оптимальности для задачи (3.82)–(3.84), т. е.  $x^*(\varepsilon) = x_\varepsilon$ .

Далее,

$$(c, x^*(\varepsilon)) - (c, \bar{x}) = \left( c, \alpha(\varepsilon) \frac{p_n}{\|p_n\|} \right) = \frac{(p_n, c) (p_n, \bar{x})}{\|p_n\|^2} \left( 1 + \frac{1}{\varepsilon} \frac{(p_n, c)}{(p_n, \bar{x})} \right), \quad (3.97)$$

$$\|p_n\|^2 = -(p_n, a_n) = \frac{|\Gamma(a_1, \dots, a_n)|}{|\Gamma(a_1, \dots, a_{n-1})|} = \frac{D}{D_a^n}. \quad (3.98)$$

Из (3.96) имеем  $\varepsilon_0 = -(p_n, c) / (p_n, \bar{x})$ .

Подставляя это выражение, а также (3.94), (3.95) и (3.98) в (3.97), получаем  $(c, x^*(\varepsilon)) - (c, \bar{x}) = u_n D_b^n / D_a^n (1 - \varepsilon_0 / \varepsilon)$ , что и требовалось доказать.

Отметим, что соотношения между  $\bar{x}$  и  $\bar{x}(\varepsilon)$  изучались В. А. Булавским [13] и О. Мангасарьяном [160] в связи с разработкой итеративных алгоритмов в линейном программировании. В частности, ими был получен результат о существовании (без указания способа вычисления) при определенных условиях такого  $\varepsilon_0 > 0$ , что  $\bar{x}(\varepsilon) = \bar{x}$  при  $\varepsilon \in (0, \varepsilon_0]$ .

Использование квадратичной регуляризации приводит к необходимости решать задачи квадратичного программирования для отдельных блоков. В том случае, когда блоки состоят из одного линейного ограничения общего вида и двусторонних ограничений на отдельные переменные (такая ситуация характерна для задач производственно-транспортного планирования), решение достигается с помощью простого алгоритма. В общем случае требуется хорошая стандартная программа квадратичного программирования, реализующая некоторый конечный прямой метод квадратичного программирования и могущая работать с любого допустимого начального приближения. Дело в том, что при переходе от одной итерации к другой при максимизации  $\varphi(u)$  меняются лишь коэффициенты линейной части целевой функции, поэтому удобно в качестве начального приближения на  $(k+1)$ -й итерации выбирать оптимальное решение  $x(u_k)$ , полученное на предыдущей итерации.

## § 6. Задача об оптимальном планировании производства и распределения труб

Эффективность использования схем декомпозиции в сочетании с квадратичной регуляризацией и методами негладкой оптимизации наиболее ярко проявляется при решении так называемых обобщенных распределительных задач большой размерности. Примером такой задачи служит задача об оптимальном планировании производства и распределения труб, формулировка и решение которой выполнялось совместно со специалистами Всесоюзного научно-исследовательского и конструкторско-технологического института трубной промышленности (ВНИТИ) Минчермета СССР.

В стране имеется  $J$  районов потребления (индекс района  $j = 1, 2, \dots, J$ ) изделий трубного производства, номенклатура которого составляет  $I$  типоразмеров (индекс типоразмера  $i = 1, 2, \dots, I$ ). Дается совокупность  $Q^{ij}$  «заказов», представляющих собой годовую потребность (в тыс. м)  $j$ -го района в трубах  $i$ -го типоразмера. Производителями труб являются  $K$  трубных цехов (индекс цеха  $k = 1, 2, \dots, K$ ), расположенные в различных районах страны. Для каждого типоразмера  $i$  задается информация о всех возможных способах производства труб этого типоразмера. Эта информация может быть удобно представлена в виде дерева. Корнем (нулевым уровнем) дерева является сам типоразмер  $i$ . На первом уровне указываются цехи, являющиеся возможными производителями этого типоразмера. Для каждого такого цеха  $k$  на втором уровне перечисляются технологические маршруты (индекс маршрута  $l$ ), по которым типоразмер  $i$  может производиться в этом цехе. Технологический маршрут представляет собой набор оборудования (индекс оборудования  $m$ ); такие наборы задаются на третьем уровне дерева.

Рядом с каждым видом оборудования указывается «норма участия»  $a_{klm}^i$ . Эта величина представляет собой затраты времени (в часах) оборудования  $m$  из цеха  $k$  при производстве в этом цехе 1000 м труб  $i$ -го типоразмера по включающему данное оборудование технологическому маршруту  $l$ .

Математическая модель описывается следующей задачей линейного программирования: найти

$$\min \sum_{(i,j) \in Z} \sum_{(k,l) \in W_i} c_{kl}^{ij} x_{kl}^{ij} \quad (3.99)$$

при ограничениях

$$x_{kl}^{ij} \geq 0, \quad (3.100)$$

$$\sum_{(k,l) \in W_i} x_{kl}^{ij} = Q^{ij}, \quad (i,j) \in Z, \quad (3.101)$$

$$\sum_{(l,i,j) \in V_{km}} a_{klm}^i x_{kl}^{ij} \leq T_{km}, \quad (k,m) \in Q, \quad (3.102)$$

$$\underline{R}_\alpha \leq \sum_{(i,j,k,l) \in U_\alpha} x_{kl}^{ij} \leq \bar{R}_\alpha, \quad \alpha \in D. \quad (3.103)$$

Здесь введены такие обозначения:

$x_{kl}^{ij}$  — искомое количество (в тыс. м) труб  $i$ -го типоразмера, спланированное для производства в течение года в  $k$ -м цехе по технологическому маршруту  $l$  и предназначенное для  $j$ -го района потребления;

$W_i$  — множество пар  $(k, l)$  (цех — маршрут), описывающее все возможные способы производства труб типоразмера  $i$ ;

$Z$  — множество пар  $(i, j)$  (типоразмер — район), которым соответствуют ненулевые заказы  $Q^{ij}$ ;

$V_{km}$  — множество троек  $(i, j, l)$  (типоразмер — район — маршрут) таких, что  $(i, j) \in Z$ , а маршрут  $l$  допустим при производстве типоразмера  $i$  в цехе  $k$  и включает оборудование  $m$ ;

$T_{km}$  — величина годового ресурса времени (в часах) оборудования  $m$  из цеха  $k$ ;

$Q$  — множество пар  $(k, m)$  (цех — оборудование), задающее парк оборудования, загрузка которого учитывается в данной модели;

$U_\alpha$  — выделенное множество четверок  $(i, j, k, l)$ ;

$D$  — некоторое конечное множество номеров  $\alpha$ .

Ограничения (3.101) описывают необходимость удовлетворения предъявленных заказов. Ограничения (3.102) говорят о невозможности эксплуатировать в течение года каждое оборудование дольше определенного годового ресурса времени. Ограничения (3.103) являются специальными связывающими величины  $x_{kl}^{ij}$  линейными ограничениями, ограничивающими сверху и (или) снизу производство в данном цехе определенных видов продукции, исходя из определенных технологических или технико-экономических соображений. Выбор коэффициентов  $c_{kl}^{ij}$  минимизируемой линейной формы (3.99) зависит от поставленной цели: они могут задавать, например, транспортные издержки, затраты на производство или другие более комплексные критерии.

Для решения задачи (3.99)—(3.103) непосредственно применялся подход, основанный на использовании схемы декомпозиции по ограничениям в сочетании с  $\epsilon$ -квадратичной регуляризацией, который подробно был рассмотрен для общего случая в § 5. При этом в качестве выпуклого множества  $X$  выбиралось множество, задаваемое в пространстве переменных  $x_{kl}^{ij}$  ограничениями (3.100), (3.101). Количество ограничений (3.101) равно числу заказов и в конкретных расчетах было порядка  $2 \cdot 10^4$ . Общее число переменных, входивших в задачу линейного программирования (3.99)—(3.103), имело порядок  $2 \cdot 10^5$ .

Ограничениям (3.78) соответствуют в данном случае неравенства (3.102), (3.103). Их общее число не превосходило 300. Поэтому получаемая в соответствии с применяемым подходом задача максимизации вогнутой функции  $\varphi_\epsilon(\cdot)$  содержала около 300 переменных. Последняя задача решалась с помощью  $r$ -алгоритма. Во всех расчетах не более чем за 1500 вычислений субградиента  $g_{\varphi_\epsilon}(\cdot)$  функции  $\varphi_\epsilon(\cdot)$  получали 5—6 верных значащих цифр в величине максимума этой функции.

Вычисление субградиента  $g_{\varphi_\epsilon}(\cdot)$  проходило по формулам типа (3.81). Оно сводилось к нахождению для каждого заказа (т. е. пары  $(i, j) \in Z$ ) точки оптимума простой задачи квадратичного программи-

рования, которая может быть представлена после введения очевидных переобозначений в следующем виде: найти

$$\min \sum_{r=1}^s \left( d_r y_r + \frac{\varepsilon}{2Q} y_r^2 \right) \quad (3.104)$$

при ограничениях

$$\sum_{r=1}^s y_r = Q, \quad y_r \geq 0, \quad r = 1, \dots, s. \quad (3.105)$$

Решение такой задачи  $y^* = \{y_1^*, \dots, y_s^*\}$  выписывается явно. Так, если коэффициенты  $d_r$  пронумерованы в порядке неубывания, то легко проверить, что

$$y_r^* = \begin{cases} \frac{Q}{\varepsilon p} \left( \varepsilon + \sum_{r=1}^p d_r - p d_r \right) & \text{при } r \leq p, \\ 0 & \text{при } r > p, \end{cases} \quad (3.106)$$

где  $p = \max \left\{ q : \left( \varepsilon + \sum_{r=1}^q d_r - q d_q \right) \geq 0 \right\}$ .

Заметим, что вычисление субградиента  $g_{\varphi_\varepsilon}(\cdot)$  требует однократного просмотра всей информации об исходной задаче (3.99) — (3.103), что необходимо для последовательного формирования всех задач вида (3.104), (3.105).

Полученные по формулам вида (3.106) на каждой  $k$ -й итерации значения  $\bar{x}_\varepsilon(u_k)$  подставляются в ограничения (3.102), (3.103). Полученные в этих ограничениях невязки являются компонентами субградиента  $g_{\varphi_\varepsilon}(\cdot)$  в точке  $u_k$ . Заметим, что в регуляризованной задаче при  $\varepsilon > 0$  субградиент вычисляется в каждой точке  $u > 0$  однозначно, то есть совпадает с градиентом.

Пусть  $u_\varepsilon^*$  — оптимальное значение множителей Лагранжа для регуляризованной задачи. Тогда из условий дополняющей нежесткости будет вытекать критерий оптимальности: те ограничения, для которых соответствующие оптимальные множители Лагранжа больше нуля, должны выполняться в форме равенства, остальные могут выполняться и в форме неравенства. Точность выполнения этого критерия дает нам возможность судить о точности решения и в нужный момент прекращать счет. Учитывая то, что время одной итерации  $r$ -алгоритма для регуляризованной задачи несколько выше (примерно в 1,5 раза), чем время решения нерегуляризованной задачи, целесообразно получить хорошее приближение для нерегуляризованной задачи, а затем перейти к решению регуляризованной с малым начальным шагом. Можно также решить нерегуляризованную задачу с высокой точностью, произвести редуцирование к задаче меньшей размерности и уже для редуцированной задачи произвести регуляризацию. Указанные приемы позволяют уменьшить время решения задачи на 20—30 %.

Описанная математическая модель соответствует задаче оптимального планирования производства и распределения холоднодеформированных труб. Она несколько упрощается в случае сварных, горячекатаных и других видов труб.

Предложенная методика решения задач такого типа использовалась при разработке плана производства различных видов труб на трубных заводах страны начиная с 1980 г. Это позволило более полно удовлетворить потребность в дефицитных размерах труб, снизить металлоемкость продукции, что обеспечило получение значительного экономического эффекта. Следует отметить, что большое по объему информационное обеспечение, потребовавшееся при решении задач оптимального планирования производства труб, подготовлено специалистами Всесоюзного научно-исследовательского и конструкторско-технологического института трубной промышленности Минчермета СССР, работниками соответствующих отделов Госплана и Госснаба СССР.

## § 7. Стохастические распределительные задачи

Часто при решении задач планирования производства и транспортировки продукции возникают ситуации, когда производственные мощности различных групп оборудования и потребности потребителей в момент планирования не могут быть точно определены. Если предположить, что указанные показатели являются случайными величинами, то мы получаем в зависимости от конкретной ситуации ту или иную модель стохастического программирования.

Практически наиболее интересны модели многоэтапного планирования, когда «жестко» определяются лишь поставки, которые наверняка будут обеспечены производством. При этом предполагается, что дисбаланс между реальными потребностями и «жесткой» составляющей плана покрывается с помощью корректирующих воздействий, состоящих в использовании проявляющихся в процессе работы стохастически прогнозируемых резервов производства, централизованных запасов продукции, импорта и т. п. В тех случаях, когда затраты на коррекцию слишком велики, приходится отказываться от обязательного удовлетворения потребности потребителя, но в модели нужно обязательно учитывать возникающий при этом ущерб для потребителя и народного хозяйства в целом.

Вариантов стохастических моделей такого рода много. Мы остановимся на двух характерных и сравнительно простых типах моделей.

1. Предположим, что возможности (ограничения) производства заданы точно, а потребности представляют собой случайные неотрицательные величины с известными законами распределения. Затраты на коррекцию первоначального плана представляют собой выпуклую функцию от невязки между реализуемой потребностью данного потребителя и планируемыми поставками. Требуется составить такой план поставок, чтобы математическое ожидание суммарных затрат на реализацию этого плана и затрат на коррекцию было минимально.

Математическая модель, которая исследуется ниже, является аналогом детерминированной модели, описанной в § 6 гл. 3.

Пусть  $i \in I$  — возможные технологические способы (это понятие включает в данном случае и место производства),  $j \in J$  — индекс заказа потребителя (заказ — это и конкретный потребитель, и конкретный вид продукции),  $T^\alpha$ ,  $\alpha \in A$ , — правые части технологических ограничений,  $\alpha$  — индекс ограничения,  $\xi_j$  — случайная величина  $j$ -го заказа,  $x_{ij}$  — величина поставки для выполнения  $j$ -го заказа с использованием  $i$ -й технологии,  $p_{ij}^\alpha$  — технологические коэффициенты,  $c_{ij}$  — затраты на выполнение и доставку единицы продукции  $j$ -го заказа  $i$ -м потребителем,  $f_j(\cdot)$  — функции, выражающие затраты на коррекцию в зависимости от невязки между  $\xi_j$  и суммарной поставкой  $\sum_{i \in I} x_{ij}$  для

выполнения  $j$ -го заказа,  $M$  — символ математического ожидания. В этих обозначениях получаем следующую математическую модель: минимизировать

$$F(x) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + M \left[ \sum_{j \in J} f_j \left( \xi_j - \sum_{i \in I} x_{ij} \right) \right] \quad (3.107)$$

при ограничениях

$$\sum_{i \in I} \sum_{j \in J} p_{ij}^\alpha x_{ij} \leq T^\alpha, \quad \alpha \in A, \quad (3.108)$$

$$x_{ij} \geq 0, \quad i \in I, \quad j \in J. \quad (3.109)$$

Будем считать, что вероятностные распределения случайных величин  $\xi_j$  и функции  $f_j$  таковы, что  $Mf_j(\xi_j - \tau)$  существуют при любом  $j \in J$  и  $\tau \geq 0$ . Так как функции  $f_j(\cdot)$  предполагаются выпуклыми, то  $Mf_j(\xi_j - \sum_{i \in I} x_{ij})$  является выпуклой функцией от  $x$ , откуда задача (3.107) — (3.109) является задачей выпуклого программирования.

Введем обозначения:  $\sum_{i \in I} x_{ij} = y_j$ ,  $j \in J$ ,  $Mf_j(\xi_j - y_j) = \varphi_j(y_j)$ ,  $y_j \geq 0$ . Воспользуемся схемой декомпозиции по ограничениям. Для этого введем вектор  $u = \{u_\alpha\}_{\alpha \in A}$  множителей Лагранжа и рассмотрим функцию Лагранжа

$$\begin{aligned} \mathcal{L}(x, u) &= \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} \varphi_j(y_j) + \sum_{\alpha \in A} u_\alpha \left( \sum_{i \in I} \sum_{j \in J} p_{ij}^\alpha x_{ij} - T^\alpha \right) = \\ &= \sum_{i \in I} \sum_{j \in J} \left( c_{ij} + \sum_{\alpha \in A} p_{ij}^\alpha u_\alpha \right) x_{ij} - \sum_{\alpha \in A} u_\alpha T^\alpha + \sum_{j \in J} \varphi_j(y_j). \end{aligned}$$

Пусть

$$\psi(u) = \min_{x \geq 0} \mathcal{L}(x, u). \quad (3.110)$$

Если зафиксировать вектор  $y = \{y_j\}$ , где  $y_j = \sum_{i \in I} x_{ij}$ , и обозначить

$$\tilde{c}_j(u) = \min_{i \in I} \left\{ c_{ij} + \sum_{\alpha \in A} p_{ij}^\alpha u_\alpha \right\}, \quad \text{то получим, что минимум в (3.110)}$$

достигается при следующих значениях  $x = \bar{x}(y)$ :

$$\bar{x}_{ij}(y) = \begin{cases} y_j, & \text{если } i \text{ — минимальный индекс, для которого } \tilde{c}_j(u) = \\ & = c_{ij} + \sum_{\alpha \in A} p_{ij}^\alpha u_\alpha \text{ (этот индекс обозначим } i(j, u)), \\ 0 & \text{в противном случае.} \end{cases}$$

Обозначим через  $\psi(y, u)$  следующее выражение:

$$\begin{aligned} \psi(y, u) &= \sum_{j \in J} \tilde{c}_j(u) y_j - \sum_{\alpha \in A} u_\alpha T^\alpha + \sum_{j \in J} \varphi_j(y_j) = \\ &= \sum_{j \in J} (\tilde{c}_j(u) + \varphi_j(y_j)) - \sum_{\alpha \in A} u_\alpha T^\alpha. \end{aligned}$$

Очевидно, что

$$\psi(u) = \min_{y \geq 0} \psi(y, u) = \sum_{j \in J} \min_{y_j \geq 0} (\tilde{c}_j(u) y_j + \varphi_j(y_j)) - \sum_{\alpha \in A} u_\alpha T^\alpha.$$

Таким образом, для отыскания  $\psi(u)$  нужно решить для каждого  $j \in J$  задачу одномерной минимизации: найти

$$\min_{y_j \geq 0} (\tilde{c}_j(u) y_j + \varphi_j(y_j)). \quad (3.111)$$

Пусть  $y(u) = \{y_j(u)\}_{j \in J}$  — вектор оптимальных значений аргумента в задачах вида (3.111). Рассмотрим задачу отыскания  $\max_{u \geq 0} \psi(u)$ . Из общей теории следует, что  $\psi(u)$  — вогнутая функция. Ее обобщенный градиент может быть вычислен покомпонентно следующим образом:

$$g_\psi(u) = \{g_\psi^\alpha(u)\}_{\alpha \in A} = \sum_{j \in J} p_{(j, u)}^\alpha y_j(u). \quad (3.112)$$

Теперь у нас есть все для того, чтобы применить некоторый метод субградиентного типа, например  $r$ -алгоритм, для нахождения максимума  $\psi(u)$  и соответствующих оптимальных значений множителей Лагранжа.

Если функции  $\varphi_j(y_j)$  являются строго выпуклыми при  $y_j \geq 0$  и  $\varphi_j(y_j) \rightarrow +\infty$  при  $y_j \rightarrow \infty$ , то задачи минимизации вида (3.111) имеют однозначное решение, а функция  $\psi(u)$  является непрерывно дифференцируемой при  $u \geq 0$ ; если при этом первоначальная задача имеет оптимальное решение, то оптимальным значениям множителей Лагранжа  $u^*$  будет соответствовать оптимальное значение  $y = y^*$ . При известном  $y$  стохастическая задача (3.107) — (3.109) превращается в детерминированную задачу. Для получения приближения оптимального вектора  $\{x_{ij}^*\}$  можно использовать прием  $\varepsilon$ -квадратичного сглаживания, рассмотренный в § 5 гл. 3.

Если функции  $\varphi_j(y_j)$  не являются строго выпуклыми, например, если они кусочно линейны, что характерно для дискретных вероятност-

ных распределений потребности, то для получения приближения к  $y^*$  можно использовать также прием квадратичной регуляризации, добавляя к минимизируемой в (3.111) функции квадратичный член вида  $\delta_j y_j^2$  с малым  $\delta_j > 0$ .

**Пример.** Рассмотрим частный случай стохастической распределительной задачи вида (3.107)—(3.109), когда ограничения (3.108) являются чисто транспортными, т. е. имеют вид

$$\sum_{i \in J} x_{ij} \leq a_i, \quad i \in I, \quad (3.113)$$

а  $f_j(t) = k_j t^+$ , где  $k_j > 0$ ,  $t^+ = \max\{t; 0\}$  (штраф пропорционален величине недопоставки продукции). Пусть  $F_j(t)$  — функция распределения случайной величины  $\xi_j$ . Тогда

$$\varphi_j(y_j) = k_j M(\xi_j - y_j) = k_j \int_{-\infty}^{\infty} (t - y_j)^+ dF_j(t) = k_j \int_{y_j}^{+\infty} (t - y_j) dF_j(t).$$

Производная по направлению  $y_j$

$$\varphi'_j(y_j) = -k_j \int_{y_j}^{\infty} dF_j(t) = k_j(F_j(y_j) - 1)$$

(предполагается, что функции  $F_j$  непрерывны справа). Решение задачи определения  $y_j(u)$  сводится к следующему: определяется  $\tilde{c}_j(u) = \min_{i \in I} (c_{ij} + u_i)$ , где  $u = \{u_i\}_{i=1}^n$  — вектор множителей Лагранжа, соответствующий ограничениям (3.113); решение задачи (3.111) сводится к исследованию уравнения

$$\tilde{c}_j(u) + k_j(F_j(y_j) - 1) = 0. \quad (3.114)$$

Учитывая, что  $0 \leq F_j(y_j) \leq 1$ , получаем, что если  $\tilde{c}_j(u) \geq k_j$ , то в качестве  $y_j(u)$  нужно брать 0. При  $\tilde{c}_j(u) < k_j$  либо существует решение  $y_j$  уравнения (3.114) (не обязательно единственное), тогда  $y_j(u) = \bar{y}_j$ , либо существует минимальное значение  $y_j = \bar{y}$ , для которого  $\tilde{c}_j(u) + k_j(F_j(y_j) - 1) > 0$ , тогда  $y_j(u) = \bar{y}$ . Если  $F_j(t)$  обладает плотностью  $p_j(t)$ , причем  $p_j(t)$  положительна на интервале  $[a_j, b_j]$ ,  $a_j \geq 0$ , а вне этого интервала равна 0, то при  $\tilde{c}_j(u) < k_j$  получаем единственное значение  $y_j(u) \in [a, b]$ . Случай неоднозначности  $y_j(u)$  может возникнуть, когда  $F_j(t)$  имеет интервалы постоянства, что характерно для дискретных распределений. В этом случае мы можем аппроксимировать  $F_j(t)$  на интервалах постоянства линейной функцией с малым положительным уклоном, чтобы обеспечить однозначность  $y_j(u)$ .

Оптимальному значению  $u = u^*$  будут соответствовать вектор  $\{y_j(u^*)\}_{j=1}^n$ . Для получения  $\{x_{ij}\}$  остается решить транспортную задачу:



найти

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

при ограничениях

$$\begin{aligned} \sum_{i=1}^n x_{ij} &= y_j(u^*), & \sum_{j=1}^m x_{ij} &\leq a_i, \\ x_{ij} &\geq 0, & i &= 1, \dots, m, \quad j = 1, \dots, n. \end{aligned}$$

2. Другая часто встречающаяся в практике планирования модель стохастической распределительной задачи связана с определением объемов централизованных складов при заданном вероятностном распределении потребностей отдельных потребителей. Пусть  $i = 1, \dots, m$  — индексы пунктов расположения складов,  $j = 1, \dots, n$  — индексы потребителей,  $c_{ij}$  — стоимость перевозки единицы продукции от  $i$ -го склада к  $j$ -му потребителю,  $r_j$  — штраф за недопоставку единицы продукции  $j$ -му потребителю,  $b_j(\omega_j)$  — случайная величина потребности  $j$ -го потребителя с известным вероятностным распределением  $F_j(t)$ ,  $y_i$  — планируемый объем  $i$ -го склада,  $d_i$  — затраты на создание единицы запаса в  $i$ -м складе,  $x_{ij}(y, \omega)$  — величина поставки из  $i$ -го склада  $j$ -му потребителю при заданном  $y = \{y_1, \dots, y_m\}$  и известных значениях потребностей  $b_j(\omega_j)$  ( $j = 1, \dots, n$ ). Предположим, что  $b_j(\omega_j)$  ( $j = 1, \dots, n$ ) — независимые случайные величины. Получаем модель двухэтапной транспортной задачи следующего вида: найти

$$\min_{\{y: y_i \geq 0\}} \sum_{i=1}^m d_i y_i + M_\omega f(y, \omega), \quad (3.115)$$

где  $M_\omega$  — символ математического ожидания,  $f(y, \omega)$  — оптимальное значение целевой функции транспортной задачи: найти

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n c_j z_j \right) \quad (3.116)$$

при ограничениях:

$$\sum_{j=1}^n x_{ij} \leq y_i, \quad i = 1, \dots, m, \quad (3.117)$$

$$\sum_{i=1}^m x_{ij} + z_j = b_j(\omega_j), \quad j = 1, \dots, n, \quad (3.118)$$

$$x_{ij}, z_j \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (3.119)$$

Прямой способ решения задачи (3.115) заключается в применении метода обобщенных стохастических градиентов (ОСГ), описанного в § 2 гл. 2. Как следует из [34, 35], для получения обобщенного стохастического градиента  $g_\omega(y)$  от  $M_\omega f(y, \omega)$  нужно путем моделирования псевдослучайных чисел в соответствии с распределениями  $F_j(\cdot)$  по-

лучить значения  $b_j(\omega_j)$  и при фиксированных  $y = \{y_1, \dots, y_m\}$  решить транспортную задачу (3.116) — (3.119).

Пусть  $u(\omega) = \{u_i(\omega)\}_{i=1}^m$  — вектор оптимальных значений двойственных переменных, соответствующих ограничениям (3.117). Тогда  $g_\omega(y) = \{-u_i(\omega)\}_{i=1}^m$ . Соответственно стохастический субградиент  $\hat{g}_\omega(y)$  от целевой функции (3.115) дается выражением

$$\hat{g}_\omega(y) = \{d_i - u_i(\omega)\}_{i=1}^m. \quad (3.120)$$

Таким образом, для решения задачи (3.115) — (3.119) может быть применен алгоритм ОСГ в следующей форме:

- 1) задаем начальное значение  $y = y^0 = \{y_i^0\}_{i=1}^m$ ;
- 2) вычисляем  $y^1, y^2, \dots, y^k, \dots$  по следующей формуле:

$$y^{k+1} = y^k - h_k \hat{g}_\omega(y^k), \quad k = 0, 1, 2, \dots, \quad (3.121)$$

где стохастический градиент  $\hat{g}_\omega(y^k)$  подсчитывается по формуле вида (3.120), причем в качестве  $\{u_i(\omega)\}$  берутся двойственные оценки, соответствующие ограничению (3.117) в транспортной задаче (3.116) — (3.119), правые части  $\{b_j(\omega_j)\}_{j=1}^n$  которой получаются путем моделирования псевдослучайных чисел с соответствующим распределением, а вместо  $y$  подставляется  $y^k$ . Шаговые множители  $\{h_k\}_{k=0}^\infty$  удовлетворяют следующим условиям:

$$h_k > 0 \quad \forall k, \quad \sum_{k=1}^{\infty} h_k = +\infty, \quad \sum_{k=1}^{\infty} h_k^2 < +\infty,$$

которые обеспечивают сходимость процесса (3.121) к оптимальной точке с вероятностью 1. В соответствии с приведенным выше алгоритмом А. Ф. Годонога разработал программу, вошедшую в качестве функционального модуля в ППП ПЛАНЕР.

## ЗАДАЧИ СИНТЕЗА СЕТЕЙ И МЕТОД РЕШЕНИЯ РЕЛАКСАЦИОННОЙ ЗАДАЧИ

---

### § 1. Задачи синтеза сетей. Модели и свойства

Задачи синтеза сетей являются математическими моделями многочисленных экономико-математических проблем, связанных с оптимальным проектированием систем продуктопроводов различных назначений [15, 50, 65], сетей дорог, авиатрасс [67], маршрутов движения грузовых поездов [80] и т. п. Кроме того, в таких же терминах формулируется ряд задач размещения и развития предприятий [66, 68] и многие задачи унификации и стандартизации [9].

С точки зрения теории оптимизации задачи синтеза сетей относятся к классу задач минимизации вогнутой (вверх) целевой функции на транспортных многогранниках различных типов. Вогнутость целевой функции связана прежде всего с так называемым законом убывающей стоимости, в соответствии с которым удельные затраты на создание и эксплуатацию мощности для производства или транспортировки продукции убывают с ростом величины этой мощности. Например, стоимость линейного участка трубопровода в первом приближении пропорциональна стоимости металла, затраченного на производство соответствующей трубы, т. е. пропорциональна диаметру трубы, а пропускная способность — мощность трубопровода при фиксированной скорости потока в нем — пропорциональна его сечению, т. е. квадрату диаметра. Поэтому с ростом мощности стоимость трубопровода растет как корень квадратный от нее, т. е. является вогнутой функцией от мощности. Удельные затраты при этом убывают с ростом мощности. Второй причиной, обуславливающей вогнутость целевой функции, является наличие в ней компоненты, связанной с фиксированными доплатами. Эта компонента соответствует постоянным или, более общо, постоянным плюс линейным затратам от мощности для любого положительного ее значения, и нулевым затратам при нулевой мощности. К этому виду относятся обычно затраты на проектирование и подготовку технической документации по созданию объекта, изготовление макетов и опытных образцов, затраты на подготовительные работы и т. д. Во многих практических ситуациях нелинейность целевой функции существенна. Поэтому аппроксимация ее линейной функцией приводит к решению, далекому от истинного оптимума, и является неприемлемой.

Задачи минимизации вогнутой функции на выпуклом множестве относятся, вероятно, к классу задач математического программирования, для которых время их решения растет не менее чем экспоненци-

ально с ростом числа переменных. Это следует из того, что даже простейшие из задач этого класса относятся к числу  $NP$ -трудных задач.

С другой стороны, любая линейная задача булева программирования (а значит, и целочисленного программирования с ограниченными сверху значениями переменных) сводима к задаче минимизации вогнутой функции с непрерывными переменными. Для подтверждения этого достаточно вместо ограничений  $X = 0 \vee 1$  рассмотреть ограничения  $0 \leq X \leq 1$  и добавить к целевой функции  $f(X)$  вогнутую компоненту

$$M \sum_{j=1}^n x_j (1 - x_j), \text{ где } X = (x_1, \dots, x_n), M \text{— большое положительное}$$

число. Верно и обратное утверждение: задача минимизации вогнутой функции на выпуклом многограннике сводима к задаче линейного программирования с булевыми переменными. Это утверждение базируется на том простом факте, что минимум вогнутой функции на многограннике достигается в одной из его вершин. В общем случае такое сведение весьма трудоемко, так как требует перечисления всех вершин многогранника. Однако в ряде случаев, когда структура целевой функции и многогранника достаточно проста, это сведение естественно. Примерами являются задачи на многогранниках транспортного типа с сепарабельной вогнутой целевой функцией и некоторые их обобщения, рассматриваемые в этой книге.

Основным объектом анализа в данном разделе является задача синтеза сетей, которая относится к классу многопродуктовых сетевых транспортных задач с сепарабельной вогнутой целевой функцией затрат по дугам и вершинам сети. Пусть задан орграф  $G = G(V, E)$  с множеством вершин  $V$  и дуг  $E$  и непустое множество  $A = \{(i, j)\}$  упорядоченных пар вершин из  $V \times V$ , называемое множеством маршрутов. Первая вершина в паре называется началом, вторая — концом маршрута. Каждому  $a \in A$  поставлено в соответствие число  $b_a > 0$ , называемое мощностью маршрута  $a$ . Каждой дуге  $l \in E$  и каждой вершине  $i \in V$  соответствуют неотрицательные вогнутые неубывающие функции затрат на транспортировку по дугам  $f_l(X_l)$  ( $l \in E$ ) и через вершины  $\varphi_i(Y_i)$  ( $i \in V$ ), равные нулю при  $X_l = 0$  и  $Y_i = 0$ , и числа  $P_l, Q_i$ , ограничивающие сверху значения переменных  $X_l, Y_i$ . Задача синтеза сетей формулируется как задача отыскания сети, допускающей реализацию на ней всех потоков из  $A$  с заданными мощностями  $b_a$  ( $a \in A$ ), удовлетворяющей дуговым и вершинным ограничениям на величину их пропускной способности, с минимумом суммарных затрат на транспортировку на вершинах и дугах сети.

Математическая модель этой задачи формулируется в виде: найти

$$\min \left( \sum_{l \in E} f_l(X_l) + \sum_{i \in V} \varphi_i(Y_i) \right) \quad (4.1)$$

при ограничениях

$$\sum_{a \in A} x(a, l) = X_l \leq P_l, \quad l \in E, \quad (4.2)$$

$$\sum_{a \in A(i)} \sum_{l \in I(i)} x(a, l) = Y_i \leq Q_i, \quad i \in V, \quad (4.3)$$

$$\sum_{l \in J(k)} x(a, l) - \sum_{l \in I(k)} x(a, l) = \begin{cases} b_a & \text{для } k = i, \\ 0 & \text{для } k \neq i, j, a = (i, j) \in A, \\ -b_a & \text{для } k = j, \end{cases} \quad (4.4)$$

$$x(a, l) \geq 0, \quad a \in A, \quad l \in E. \quad (4.5)$$

Здесь  $x(a, l)$  — неизвестная мощность потока для маршрута  $a$  по дуге  $l$ ,  $X_l$  — суммарный поток по дуге  $l$ ,  $Y_i$  — транзитный поток через вершину  $i$ ,  $i \in V$ ,  $l \in E$ ,  $a \in A$ ,  $I(k)$  и  $J(k)$  — множества входящих и выходящих дуг для вершины  $k$  ( $k \in V$ ),  $A(k) \subset A$  — подмножество транзитных относительно  $k$  маршрутов, т. е.  $A(k) = \{a = (i, j): i \neq k, j \neq k\}$ . Ограничения (4.4) — стандартные для потоковых задач условия баланса мощности и неразрывности потока. В дальнейшем мы предполагаем, что все  $P_i$  и  $Q_i$  выбраны так, что задача (4.1)—(4.5) разрешима, и не оговариваем этот факт особо.

Наличие в целевой функции слагаемых  $\varphi_i(Y_i) \neq 0$  соответствует затратам на размещение в вершине  $i$  некоторой мощности  $Y_i$ , перерабатывающей транзитные потоки с затратами, равными  $\varphi_i(Y_i)$ . Величины  $f_i(X_i)$  есть затраты на создание и эксплуатацию коммуникации (дуги)  $l$  с пропускной способностью  $X_l$ . В принципе можно преобразовать задачу (1.1)—(1.5) таким образом, что будут выполнены условия  $\varphi_i(Y_i) \equiv 0$  ( $i \in V$ ). Такой путь не всегда рационален, но мы будем пользоваться им для упрощения дальнейшего изложения. В ряде случаев для того, чтобы избавиться от функций затрат в вершинах графа, не меняя при этом структуры ограничений задачи, достаточно заменить каждую вершину  $i$  с  $\varphi_i(Y_i) \neq 0$  парой новых вершин, которые соединены парой противоположно направленных дуг. При этом функция затрат для  $i$  отображается в аналогичные функции для новых дуг. Подобным же образом отображаются пропускные способности  $P_i$  и элементы множества  $A$ , которые имеют вершину  $i$  в качестве одной из своих компонент.

В задаче (4.1)—(4.5) без ограничения общности можно предполагать, что все  $f_l(X_l)$  — кусочно линейные функции. Действительно, поскольку минимум вогнутой функции достигается в вершине многогранника и число вершин конечно, достаточно определить значения каждой из рассматриваемых функций в вершинах и затем заменить эту функцию кусочно линейной аппроксимацией по этим точкам. Ясно, что такой способ не является конструктивным с вычислительной точки зрения из-за экспоненциального роста числа вершин с ростом размерности задачи. Важнее то, что в практических задачах функции  $f_l(X_l)$  сразу задаются (или приближаются) кусочно линейными. Такими же они являются в ряде частных случаев задачи (4.1)—(4.5), например в задаче Штейнера (см. ниже).

Кусочно линейная одномерная функция  $f(X)$  представима в свою очередь в форме

$$f(X) = \min_{r \in R} h_r(X), \quad (4.6)$$

где

$$h_r(X) = \begin{cases} c_r X + d_r, & c_r, d_r \geq 0 \text{ при } X > 0, \\ 0 & \text{при } X = 0. \end{cases} \quad (4.7)$$

Здесь  $R$  — множество участков линейности  $f(X)$ . Неотрицательность  $c_r, d_r$  — следствие неотрицательности, вогнутости и неубывания  $f(X)$  с ростом  $X$ . Используя представление  $f(X)$  в виде (4.6), задачу с сепарабельными кусочно линейными вогнутыми функциями вида (4.6), (4.7) можно свести к эквивалентной задаче с функциями вида (4.7), т. е. с линейными функциями с фиксированными доплатами.

*Л е м м а 4.1. Существует взаимно однозначное соответствие между ду множеством базисных решений задачи A: найти*

$$\min \left\{ \sum_{j=1}^n f_j(x_j) : \sum_{j=1}^n a_j x_j = b, x_j \geq 0, j = 1, \dots, n \right\}, \text{ где } a_j, j = 1, \dots, n,$$

*b — заданные  $m$ -мерные векторы, и подмножеством базисных решений*

$$\text{задачи B: найти } \min \left\{ \sum_{j=1}^n \sum_{r \in R} h_{j,r}(x_{j,r}) : \sum_{j=1}^n a_j \sum_{r \in R} x_{j,r} = b, x_{j,r} \geq 0, j = 1, \dots, n,$$

$$r \in R \right\}, \text{ определяемое отношениями } x_j = \sum_{r \in R} x_{j,r}, j = 1, \dots, n, \text{ и сохра-}$$

*няющие значения функционала.*

*Д о к а з а т е л ь с т в о.* Поскольку для каждого  $j$  в произвольном базисном решении задачи  $B$  лишь одна из переменных  $x_{j,r}$  может быть отлична от нуля (иначе базис содержит два одинаковых столбца), в качестве образа произвольного базисного решения  $\bar{X}$  задачи  $A$  выберем решение  $\bar{Y}$  задачи  $B$ , в котором для каждого  $j$  выбрано такое  $r$  из  $R$ , что  $f_j(\bar{Y}_j) = h_{j,r}(\bar{Y}_{j,r})$  при  $r \in R$ . Ясно, что такое отображение обладает требуемыми свойствами.

Применение указанного преобразования к (4.1)—(4.5) эквивалентно замене исходного графа  $G(V, E)$  на мультиграф, в котором каждая дуга  $l \in E$  заменена  $|R(l)|$  дугами, где  $|R(l)|$  — число линейных участков в графике функции  $f_l(X_l)$ . При этом ограничения на мощность каждой дуги в графе  $G$  перейдут в ограничения на суммарные мощности всех дуг, появившихся в мультиграфе вместо дуги исходного графа.

Отметим, что если все функции  $h_r(X)$  в представлении  $f(X)$  из (4.6) существенны, т. е. для каждого  $r \in R$  найдется хотя бы одно значение  $X(r)$ , для которого существует ненулевая окрестность  $S(X(r))$  такая, что  $\min_{t \in R} h_t(X) = h_r(X), \forall X \in S(X(r))$ , то упорядочение по убыванию

величин  $c_r$  соответствует упорядочению по возрастанию всех  $d_r (r \in R)$ . Этот факт полезен при разработке методов решения задач минимизации вогнутого функционала, так как позволяет после удаления из  $R$  несущественных функций работать с функциями вида (4.6) почти так же, как с функциями вида (4.7). Поскольку задачи синтеза сетей с функциями вида (4.7) на мультиграфах не имеют принципиальных отличий от аналогичных задач на графах, мы в дальнейшем будем рассматривать эти задачи в терминах графов, оговаривая особо случаи, когда требу-

ется рассмотрение мультиграфов. Аналогично будем особо оговаривать отличие от нуля функций  $\varphi_i(Y_i)$  в (4.1).

Задача (4.1)—(4.5) с функциями  $f_l(X_l)$  вида (4.7) эквивалентна следующей задаче частично целочисленного линейного программирования: найти

$$\Phi(X^*, Z^*) = \min_{X, Z} \Phi(X, Z) = \min \left( \sum_{l \in E} c_l \sum_{a \in A} x(a, l) + \sum_{l \in E} d_l z_l \right) \quad (4.8)$$

при ограничениях

$$\sum_{l \in J(k)} x(a, l) - \sum_{l \in I(k)} x(a, l) = \begin{cases} b_a & \text{при } k = i, \\ 0 & \text{при } k \neq i, j, \quad a = (i, j) \in A, \\ -b_a & \text{при } k = j, \end{cases} \quad (4.9)$$

$$\sum_{a \in A} x(a, l) \leq P_l, \quad l \in E, \quad (4.10)$$

$$0 \leq x(a, l) \leq M(a, l) z_l, \quad l \in E, \quad a \in A, \quad (4.11)$$

$$z_l = 0 \vee 1, \quad l \in E, \quad (4.12)$$

где  $M(a, l) = \min \{P_l, b_a\}$  — максимально возможный поток маршрута  $a$  по дуге  $l$ . Эквивалентность этой пары задач понимается здесь в том смысле, что оптимальное решение задачи (4.1)—(4.5), (4.7) ( $X^*$ ,  $Y^*$ ) порождает оптимальное решение задачи (4.8)—(4.12) ( $X^*$ ,  $Z^*$ ) по следующему правилу:

$$Y_{i^*} > 0 \Leftrightarrow Z_i^* = 1,$$

$$Y_i^* = 0 \Leftrightarrow Z_i^* = 0,$$

и значения функционалов на этих решениях совпадают. Отметим, что при фиксированном значении переменных  $Z_l$  задача (4.8)—(4.12) превращается в многопродуктовую сетевую транспортную задачу линейного программирования с ограничениями на пропускные способности дуг. Свойства этого типа задач и их частных случаев являлись предметом интенсивного изучения, начиная со времени создания теории линейного программирования. Для ознакомления с ними отсылаем читателя к работам [3, 25, 79].

В заключение раздела приведем (неполную) классификацию частных случаев и некоторых обобщений задачи (4.1)—(4.5), рассматриваемых далее. Иерархические отношения между ними изображены на рис. 4.1. Стрелки указывают порядок вложения задач; в квадратиках представлены их номера. Кратко перечислим их характерные свойства.

1. Задача синтеза сети есть задача (4.1)—(4.5).

2. Задача синтеза надежной сети является обобщением задачи (4.1)—(4.5) в следующем направлении. Предполагается, что в процессе эксплуатации сети пропускная способность каждой дуги может стать равной нулю, причем в любой момент число таких дуг не превышает  $k$ . Для каждого маршрута  $a \in A$  заданы нормальная и аварийные мощности; нормальная соответствует безотказной работе сети, аварийная должна быть обеспечена при выходе из строя хотя бы одной из дуг се-

ти. В условиях неопределенности моментов выхода из строя дуг сети требуется синтезировать сеть минимальной стоимости, обеспечивающую пропуск по ней заданных мощностей в нормальном и аварийном режимах. Мы рассматриваем частный случай этой задачи:  $k = 1$ , пропускные способности дуг не ограничены сверху, все  $a \in A$  имеют вид  $(0, i)$ , где  $0$  — заданная (корневая) вершина, т. е. задачу с одним источником (корнем).

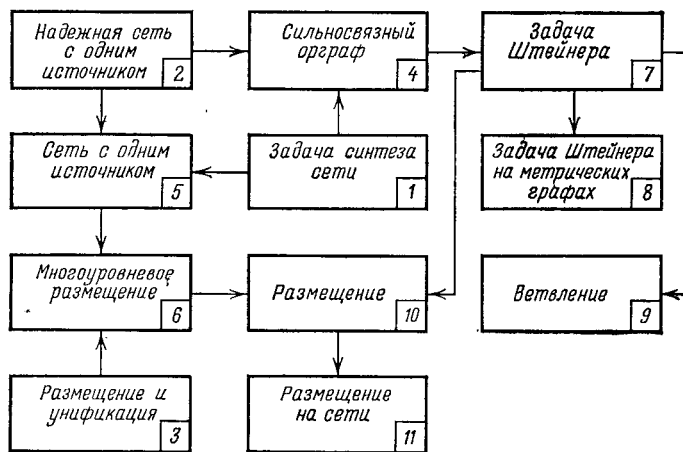


Рис. 4.1

8. Задача размещения и унификации обобщает задачи размещения, являющиеся частным случаем задачи (4.1)—(4.5), аналогично тому, как многоиндексные транспортные задачи обобщают обычную.

4. Задача нахождения сильно связанного подграфа минимального веса является частным случаем задачи (4.1)—(4.5), в которой пропускные способности дуг не ограничены сверху и функции  $f_i(X_i)$  имеют вид

$$f_i(X_i) = \begin{cases} d_i & \text{при } X_i > 0, \\ 0 & \text{при } X_i = 0. \end{cases} \quad (4.13)$$

Эта задача является также частным случаем задачи синтеза надежной сети.

5. Задача синтеза сети с одним источником — это задача (4.1)—(4.5), в которой все  $a \in A$  имеют вид  $(0, i)$ , где  $0$  — заданная корневая вершина.

6. Многоуровневая задача размещения есть частный случай задачи 5 на многодольных графах, т. е. на графах, вершины которых разбиты на  $K$  непересекающихся множеств, называемых долями. Все доли линейно упорядочены, первая доля состоит из одной вершины  $0$  (корня). Множество последней доли —  $N_k$ ,  $A = \{0, i: i \in N_k\}$ , каждая дуга  $l \in E$  имеет вид  $l = (i, j)$ , где  $i \in N_k, j \in N_{k+1}$  ( $k = 1, \dots, K - 1$ ).



7. Задача Штейнера на графе есть частный случай задачи 5, в котором все  $f_l(X_l)$  имеют вид (4.13).

8. Задача Штейнера на метрических графах есть частный случай задачи 7, в котором множество вершин  $G(V, E)$  — точки метрического пространства, а величины  $d_l$  из (4.13) пропорциональны расстояниям (в метрике пространства) между вершинами, которые соединяет дуга  $l$ .

9. Задача о кратчайшем ветвлении — это частный случай задачи 7, в котором  $A$  — полное множество, т. е.  $\forall i \in V \quad (0, i) \in A$ .

10. Задача размещения производства есть частный случай задачи 6 с  $k = 3$ .

11. Задача размещения на графах отличается от задачи 10 подобно тому, как различаются матричные и сетевые транспортные задачи.

Кроме указанных классов задач синтеза сетей рассматриваются задачи размещения технических средств с учетом сезонности выполнения работ (§ 5.3) и задача Вебера на сети с прямоугольной метрикой (§ 6.4). Эти задачи непосредственно не формулируются в виде (4.1) — (4.5), но для их решения используются методы, являющиеся обобщениями предлагаемого ниже декомпозиционного подхода.

## § 2. Задачи линейного программирования с переменными верхними границами и блокирующие множества

При решении задач целочисленного линейного программирования большинством известных методов (таких, как методы ветвей и границ, методы отсечения) возникает необходимость в решении последовательности релаксированных задач, которые получаются из исходной удалением требования целочисленности переменных. Полученное на оптимальном решении релаксированной задачи значение функционала является оценкой снизу (для задачи минимизации) функционала исходной задачи. В данном разделе рассматривается класс задач линейного программирования, включающий в себя как частный случай и релаксации задачи синтеза сетей. Для этого класса развит метод декомпозиционного типа, позволяющий свести решение исходной задачи к решению последовательности задач меньшей размерности.

Рассмотрим задачу линейного программирования вида: найти

$$\min \Phi(X, Y) = \min (cX + fY) \quad (4.14)$$

при ограничениях

$$AX = b, \quad (4.15)$$

$$X \geq 0, \quad Y \geq 0, \quad (4.16)$$

$$X \leq DY. \quad (4.17)$$

Здесь  $X$  и  $Y$  —  $n$ - и  $p$ -мерные векторы переменных;  $A$  и  $D$  — матрицы размерности  $m \times n$  и  $n \times p$  соответственно, причем  $D$  — матрица с неотрицательными элементами;  $b, c, f$  — заданные  $m$ -,  $n$ - и  $p$ -мерные векторы.

При фиксированном допустимом значении компонент вектора  $U$  задача (4.14)—(4.17) превращается в задачу с ограниченными значениями вектора  $X$ . Поэтому она называется задачей с переменными верхними границами.

Исследуем некоторые свойства задачи (4.14)—(4.17). Для упрощения изложения будем предполагать, что задача разрешима, не оговаривая далее этого специально.

Рассмотрим задачу, двойственную к (4.14)—(4.17), которая получается из нее путем сложения с целевой функцией ограничений (4.17), умноженных на соответствующие им множители Лагранжа  $U$ : найти

$$\max F(U) = \max_U \min_X (c + U)X \quad (4.18)$$

при ограничениях

$$AX = b, \quad X \geq 0, \quad (4.19)$$

$$UD \leq f, \quad U \geq 0. \quad (4.20)$$

Форма записи двойственной задачи (4.18)—(4.20) непосредственно получается из второй теоремы двойственности в линейном программировании (теоремы о дополняющей нежесткости), согласно которой для пары двойственных оптимальных базисных решений  $(X^*, Y^*)$  и  $U^*$  выполняются условия

$$Y^*(f - U^*D) = 0, \quad (4.21)$$

а для пары произвольных допустимых решений  $(X, Y)$  и  $U$  имеет место неравенство

$$Y(f - UD) \geq 0. \quad (4.22)$$

Последнее следует из (4.20) и  $Y \geq 0$ . Поэтому если  $U^*$  — оптимальное (базисное) решение задачи (4.18)—(4.20), то, определяя из теоремы двойственности вектор  $Y^*$ , построим оптимальное решение  $(X^*, Y^*)$  задачи (4.14)—(4.17). Из двойственности задач (4.14)—(4.17) и (4.18)—(4.20) следуют обычные соотношения между  $\Phi(X, Y)$  и  $F(U)$  (первая теорема двойственности в линейном программировании):

$$\Phi(X, Y) \geq F(U), \quad \Phi(X^*, Y^*) = F(U^*),$$

где  $(X^*, Y^*)$  и  $U^*$  — оптимальные решения задач (4.14)—(4.17) и (4.18)—(4.20) соответственно. Поэтому любое допустимое решение  $U$  задачи (4.18)—(4.20) порождает оценку снизу  $F(U)$  для функционала задачи (4.14)—(4.17).

Задачу минимизации  $(c + U)X$  при ограничениях (4.19) и фиксированном значении  $U = \bar{U}$  будем называть внутренней задачей. Рассмотрим также двойственную к ней задачу, вводя двойственные переменные  $V$  к условиям (4.19): найти

$$\max Vb \quad (4.23)$$

при ограничениях

$$VA \leq c + \bar{U}. \quad (4.24)$$

Пусть  $V^*(\bar{U})$  — оптимальное решение задачи (4.23), (4.24) при фиксированном  $\bar{U}$ . Обозначим через  $J(\bar{U})$  множество индексов перемен-

ных  $x_j$ , для которых соответствующие неравенства из (4.24) являются равенствами при  $V = V^*(\bar{U})$ . Полагая  $x_j = 0$  для  $j \notin J(\bar{U})$ , мы получим подмногогранник (грань) исходного многогранника, каждая точка которого является оптимальным решением внутренней задачи при фиксированном  $\bar{U}$ . Обозначим множество точек этой грани через  $X(\bar{U})$ .

Пусть  $X^0 \in X(\bar{U})$  — заданная точка. Множество  $v(X^0) = \{j: x_j^0 > 0\}$  называется носителем  $X^0$ .

Множество  $\beta(\bar{U})$  индексов  $j$  из  $J(\bar{U})$  называется блокирующим множеством для  $X(\bar{U})$ , если для любого  $X^0 \in X(\bar{U})$  имеет место  $\beta(\bar{U}) \cap v(X^0) \neq \emptyset$ .

Блокирующее множество  $\beta(\bar{U})$  называется минимальным (по включению), если не существует блокирующего множества  $\gamma(\bar{U})$  такого, что  $\gamma(\bar{U}) \subset \beta(\bar{U})$ ,  $\gamma(\bar{U}) \neq \beta(\bar{U})$ .

Блокирующие множества были введены Фалкерсоном [132] и изучались с различных точек зрения Фалкерсоном, Эдмондсом и другими [124, 133, 175]. Развита в этих работах теория использовалась для построения теории двойственности в экстремальных комбинаторных задачах, в том числе и для анализа комбинаторных задач на узкие места. Мы хотим использовать свойства блокирующих множеств для построения декомпозиционных методов точного и приближенного решения задачи (4.18)—(4.20).

Начнем с изложения основных свойств блокирующих множеств к множеству решений, лежащих на грани  $X(\bar{U})$ . При этом достаточно ограничиться лишь базисными решениями из грани  $X(\bar{U})$ .

**Теорема 4.1.** Если  $\beta(\bar{U}) \cap v(X^0) \neq \emptyset$  для любого базисного решения  $X^0 \in X(\bar{U})$ , то  $\beta(\bar{U})$  — блокирующее множество.

**Доказательство.** Пусть  $\beta(\bar{U}) \cap v(X^0) = \emptyset$  для  $X^0 \in X(\bar{U})$ . Тогда система  $x_j = 0$  ( $j \notin v(X^0)$ ) определяет непустую подгрань  $X(\bar{U})$ , которая содержит хотя бы одну вершину  $X^1$ . Для этой вершины  $v(X^1) \subset v(X^0)$  и, следовательно,  $\beta(\bar{U}) \cap v(X^1) = \emptyset$  в противоречии с условием теоремы.

Легко показать, что множество носителей базисных векторов грани  $X(\bar{U})$  попарно несравнимо по включению, т. е. для любых таких  $X^1, X^2$ ,  $v(X^1) \not\subset v(X^2)$ . Такая система множеств называется *клаттером*. С другой стороны, минимальные блокирующие множества также образуют клаттер. Обозначим первый клаттер через  $R(\bar{U})$ , а второй — через  $B(R(\bar{U}))$ .

**Теорема 4.2.**  $B(B(R(\bar{U}))) = R(\bar{U})$ .

За доказательством этой теоремы мы отсылаем читателя к работе [132].

Пара  $R(\bar{U}), B(R(\bar{U}))$  называется *блокирующей системой*. Одно из основных ее свойств состоит в следующем.

**Теорема 4.3.** Пусть  $J_1, J_2$  — произвольное разбиение  $J(\bar{U})$ , т. е.  $J_1 \cup J_2 = J(\bar{U})$ ,  $J_1 \cap J_2 = \emptyset$ . Тогда либо элемент из  $R(\bar{U})$  со-

держится в  $J_1$ , либо элемент из  $B(R(\bar{U}))$  — в  $J_2$ , или наоборот, но обе возможности не могут быть реализованы одновременно.

**Доказательство.** Элемент  $R(\bar{U})$  не может принадлежать  $J_1$ , если элемент  $B(R(\bar{U}))$  принадлежит  $J_2$ , и наоборот по определению блокирующего множества. С другой стороны, если  $J_1$  и  $J_2$  не содержат элементов из  $R(\bar{U})$ , то множество  $J_1$  (и  $J_2$ ) является блокирующим и, следовательно, содержит минимальное.

Вернемся теперь к задаче (4.18)—(4.20) и рассмотрим грань ее оптимальных решений при произвольном допустимом  $U = \bar{U}$ . Пусть  $J$  — множество всех столбцов матрицы  $A$ ,  $J_1 = J(\bar{U})$ ,  $J_2 = J \setminus J_1$ ,  $A^1$  и  $A^2$  — подматрицы  $A$ , порожденные  $J_1$  и  $J_2$ ,  $(c^1, c^2)$  — соответствующее разбиение вектора  $c$ . Иными словами,  $J_1$  — множество столбцов, для кото-

$$V^*(\bar{U})A^1 = c^1 + \bar{U}_1,$$

а  $J_2$  — множество столбцов, для которых

$$V^*(\bar{U})A^2 < c^2 + \bar{U}_2.$$

Тогда задачу (4.18)—(4.20) можно представить в виде: найти

$$\min \bar{c}^2 X_2$$

при ограничениях

$$A^1 X_1 + A^2 X_2 = b, \quad X_1 \geq 0, \quad X_2 \geq 0, \quad (4.25)$$

где  $\bar{c}^2 = c^2 + \bar{U}_2 - V^*(\bar{U})A^2$ .

Рассмотрим произвольную линейную комбинацию уравнений (4.25) с коэффициентами  $\Lambda = (\lambda_i)$  ( $i=1, \dots, m$ ), выбранными так, что  $b(\Lambda) = \sum_{i=1}^m b_i \lambda_i > 0$ . Обозначим

$$J_1^+(\Lambda) = \left\{ j \in J_1: \sum_{i=1}^m a_{ij} \lambda_i > 0 \right\}.$$

**Лемма 4.4.** Множество  $J_1^+(\Lambda)$  принадлежит  $B(R(\bar{U}))$ .

**Доказательство.** Положим  $x_j = 0$  для  $j \in J_1^+(\Lambda)$ . Из того, что в рассматриваемой линейной комбинации уравнений (4.25) все коэффициенты при переменных  $x_j$  ( $j \in J_1 \setminus J_1^+(\Lambda)$ ) неположительны, а свободный член  $b(\Lambda) > 0$ , следует, что она не имеет неотрицательных решений. Поскольку множество решений уравнения, являющегося линейной комбинацией некоторой системы, содержит множество ее решений, то из неразрешимости линейной комбинации в неотрицательном ортанте следует несовместность (4.25) при  $x_j = 0$  ( $j \in J_1^+(\Lambda)$ ). Последнее, по определению блокирующего множества, эквивалентно утверждению леммы.

Рассмотрим вектор-строку коэффициентов линейной комбинации, введенной в лемме, т. е. вектор

$$A(\Lambda) = \left( \sum_{i=1}^m a_{i1} \lambda_i, \quad \sum_{i=1}^m a_{i2} \lambda_i, \quad \dots, \quad \sum_{i=1}^m a_{in} \lambda_i \right)$$

с  $b(\Lambda) > 0$ . Такой вектор в дальнейшем будем называть увеличивающим. Если увеличить значения двойственных переменных  $V_i^*(\bar{U})$  на величины  $\delta\lambda_i$  ( $i = 1, \dots, m$ ), где  $\delta > 0$  достаточно мало, то для того, чтобы решение  $X^0$  осталось оптимальным, достаточно увеличить компоненты вектора  $\bar{U}$ , принадлежащие блокирующему множеству  $J_1^+(\Lambda)$ , на соответствующую компоненту вектора  $A(\Lambda)$ , умноженную на  $\delta$ , оставив без изменения компоненты  $\bar{U}$ , не принадлежащие  $J_1^+(\Lambda)$ . При этом значение  $F(\bar{U})$  увеличивается на  $\delta b(\Lambda) > 0$ .

Максимальное значение  $\delta$ , обладающее указанными свойствами, определяется из соотношения

$$\delta \sum_{i=1}^m a_{ij}\lambda_i \leq \bar{c}_j, \quad j \in J_2^+(\Lambda),$$

где

$$J_2^+(\Lambda) = \left\{ j \in J_2 : \sum_{i=1}^m a_{ij}\lambda_i > 0 \right\}.$$

Отсюда

$$\max \delta = \delta_0 = \min_{j \in J_2^+(\Lambda)} \bar{c}_j / \sum_{i=1}^m a_{ij}\lambda_i = \bar{c}_{j_0} / \sum_{i=1}^m a_{ij_0}\lambda_i. \quad (4.26)$$

При увеличении значения  $\delta$  до значения, большего чем  $\delta_0$ , столбец  $j_0$ , на котором достигается минимум в (4.26), перестает быть двойственно допустимым и, следовательно,  $X^0$  становится неоптимальным. При  $\delta = \delta_0$  столбец  $j_0$  становится из небазисного базисным и, следовательно, появляется множество новых базисных решений, в которых переменная  $x_{j_0}$  принимает положительное значение. Наконец, при любом положительном значении  $\delta$  все переменные  $x_j$  ( $j \in J_1 \setminus J_1^+(\Lambda)$ ), становятся небазисными и определяемые ими базисные решения перестают принадлежать  $X(\bar{U})$ .

Вышеприведенные утверждения могут быть сформулированы в следующем виде.

**Теорема 4.5. 1.** Для произвольных  $\bar{U}$ ,  $\Lambda$  и базисного вектора  $X^0 \in X(\bar{U})$  блокирующее множество  $J_1^+(\Lambda)$ , построенное по ограничениям (4.25), не меняется для любых значений  $\delta$  из  $[0, \delta_0]$ , где  $\delta_0$  определяется по (4.26);  $X^0$  остается при этом базисным решением для  $X(\tilde{U})$ , где

$$\tilde{U} = \begin{cases} \bar{U} + \delta A(\Lambda) & \text{для } j \in J_1^+(\Lambda), \\ \bar{U} & \text{для } j \notin J_1^+(\Lambda), \end{cases}$$

$$F(\tilde{U}) = F(\bar{U}) + \delta b(\Lambda).$$

2. При  $\delta = \delta_0$   $X^0$  остается базисным решением, однако блокирующее множество  $J_1^+(\Lambda)$  дополняется элементом  $j_0$ , на котором достигается минимум в (4.26).

Используя эту теорему, мы предлагаем в следующем разделе декомпозиционный алгоритм решения задачи (4.18)—(4.20), который формирует и решает следующую задачу линейного программирования: найти

$$\max \left( F(0) + \sum_{(U, \Lambda)} \omega(U, \Lambda) \right) \quad (4.27)$$

при ограничениях

$$\sum_{(U, \Lambda)} q_k(U, \Lambda) \omega(U, \Lambda) \leq f_k, \quad k = 1, \dots, p, \quad (4.28)$$

$$0 \leq \omega(U, \Lambda) \leq r(U, \Lambda) \quad \forall (U, \Lambda), \quad (4.29)$$

где  $U$  — произвольная допустимая точка из (4.20);  $\Lambda$  — вектор размерности  $m$ , определяющий увеличивающий вектор  $A(\Lambda)$  для соответствующего  $U$ ;  $\omega(U, \Lambda)$  — переменная, определяющая приращение функционала (4.18) при увеличении соответствующих данному  $U$  компонент из  $J_1^+(\Lambda)$  на величину  $\delta A(\Lambda)$ , где  $\delta \geq 0$ ;  $Q(U, \Lambda) = (q_k(U, \Lambda))$ ,  $k = 1, \dots, p$ , — сектор-столбец коэффициентов удельного уменьшения величин  $f_k$  при увеличении  $\omega(U, \Lambda)$ ;  $r(U, \Lambda)$  — величина, равная  $\delta_0 A(\Lambda)$ , где  $\delta_0$  определяется из (4.26). Оптимальное решение задачи (4.27)—(4.29) однозначно определяет оптимальное решение (4.18)—(4.20). Формально задача (4.27)—(4.29) содержит бесконечное число переменных, однако приводимый далее алгоритм ее решения генерирует и анализирует лишь конечное их число.

### § 3. Динамическая декомпозиция задачи (4.18)—(4.20) и метод ее решения

Начнем изложение с описания неконструктивного (ввиду бесконечного числа переменных) способа построения задачи (4.27)—(4.29), декомпозиционной по отношению к задаче (4.18)—(4.20).

По теореме 4.5 каждому допустимому вектору  $U$  можно поставить в соответствие множество векторов  $\Lambda$  с  $b(\Lambda) > 0$ , для каждого из которых однозначно определяется вектор  $A(\Lambda)$  и множество его положительных компонент  $J_1^+(\Lambda) \subset J_1$ . При увеличении соответствующих компонент вектора  $U$  на величины  $\delta \sum_{i=1}^m a_{ij} \lambda_i$  ( $j \in J_1^+(\Lambda)$ ) значение функционала увеличивается на  $\delta b(\Lambda) > 0$  (здесь  $\delta > 0$ ). Обозначим через  $\omega(U, \Lambda)$  величину  $\delta b(\Lambda)$ . Поскольку  $b(\Lambda) > 0$ , а  $\delta$  должно выбираться неотрицательным, переменная  $\omega(U, \Lambda)$  также должна принимать неотрицательные значения. С другой стороны, увеличение функционала происходит лишь для значений  $\delta \leq \delta_0$ . Поэтому, обозначив через  $r(U, \Lambda)$  величину  $\delta_0 b(\Lambda)$  для данного  $U$ , получим, что  $\omega(U, \Lambda)$  должна удовлетворять двусторонним ограничениям (4.29).

С другой стороны, значения вектора  $U$  должны удовлетворять ограничениям (4.20). При переходе к переменным  $\omega(U, \Lambda)$  они эквивалентны (4.28), где компоненты вектор-столбца условий  $Q(U, \Lambda)$

вычисляются по следующей формуле:

$$q_k(U, \Lambda) = \sum_{i=1}^m \sum_{j \in J_1^+(U)} a_{ij} \lambda_i d_{jk}, \quad k = 1, \dots, p. \quad (4.30)$$

Пусть теперь  $U_1 \neq 0$  — произвольное допустимое решение задачи (4.18)—(4.20). Оно порождает множество увеличивающих векторов  $A(\Lambda)$  и, соответственно, множество переменных  $w(U_1, \Lambda)$ . Однако любая переменная  $w(U_1, \Lambda_1)$  в оптимальном решении задачи (4.27)—(4.29) может быть положительной лишь в том случае, если положительна переменная  $w(U_0, \Lambda_0)$ , где  $U_0 \neq U_1$  — допустимое решение вида

$$U_1 = U_0 + \delta A(\Lambda_0). \quad (4.31)$$

Если это условие нарушено, то оптимальному решению задачи (4.27)—(4.29) нельзя поставить в соответствие оптимальное решение задачи (4.18)—(4.20). Однако легко показать, что указанное условие выполняется автоматически. Действительно, по утверждению второй части теоремы 4.5 найдется вектор  $\Lambda_0$  из множества векторов  $\Lambda$ , порожденных  $U_0$ , такой, что для любого данного  $\Lambda_1$  из множества векторов  $\Lambda$ , порожденных  $U_1$ , где  $U_0$  и  $U_1$  связаны соотношением (4.31), имеет место

$$J_1^+(\Lambda_1) = J_1^+(\Lambda_0) \cup j_0;$$

здесь  $j_0$  — индекс, на котором достигается минимум в (4.26). Поскольку  $j_0 \in J_2^+(\Lambda_0)$ , то  $\sum_{i=1}^m a_{ij_0} \lambda_i^0 > 0$ . По предположению относительно коэффициентов матрицы  $D$  все  $d_{jk}$  неотрицательны. Поэтому  $Q(U_1, \Lambda_1) \geq Q(U_0, \Lambda_0)$ . Это неравенство имеет по координатный смысл и, кроме того, если  $d_{j_0 k} > 0$  хотя бы для одного  $k$ , то  $q_k(U_1, \Lambda_1) > q_k(U_0, \Lambda_0)$  для этого же  $k$ . Так как коэффициенты при переменных  $w(U_0, \Lambda_0)$  и  $w(U_1, \Lambda_1)$  в целевой функции (4.27) одинаковы, то отсюда вытекает, что существует оптимальное решение  $W^*$  задачи (4.27)—(4.29), в котором из  $w^*(U_1, \Lambda_1) > 0$  следует  $w^*(U_0, \Lambda_0) = r(U_0, \Lambda_0)$ . Иными словами, оптимальному решению задачи (4.27)—(4.29) соответствует оптимальное решение задачи (4.18)—(4.20).

Перейдем теперь к описанию декомпозиционного алгоритма решения задачи (4.18)—(4.20) путем генерации последовательности решений задачи (4.27)—(4.29). Описываемый метод является, по существу, методом симплексного типа, на каждом шаге которого определено множество  $S(W)$  ненулевых переменных  $w(U, \Lambda)$  с их значениями, множество  $P(W)$  ограничений (4.28), обращающихся в равенство на текущем решении  $W$  и подмножества переменных  $N(W) \subset S(W)$ , значения которых лежат строго внутри соответствующих им отрезков (4.29). При обычном предположении о невырожденности задачи (4.27)—(4.29) имеет место равенство  $|N(W)| = |P(W)|$ . Каждый шаг метода, используя имеющуюся информацию о текущем решении  $W$ , проверяет его на оптимальность (в случае утвердительного ответа процесс решения заканчивается) и строит новое базисное решение, на

котором значение функционала увеличивается. Проверка на оптимальность или выбор переменной, вводимой в базис, осуществляется путем решения вспомогательной задачи, в результате чего определяется также  $Q(U, \Lambda)$  и  $r(U, \Lambda)$ . Все остальные преобразования, выполняемые на текущем шаге, в принципе не отличаются от соответствующих этапов симплекс-метода. Изложим принципиальную схему метода.

На предварительном шаге определяются исходные значения векторов и множеств:  $U^0 = 0$ ,  $W^0 = 0$ ,  $S(W^0) = P(W^0) = N(W^0) = \emptyset$ . Значение функционала (4.27) равно при этом  $F(0)$ . Напомним, что  $U = 0$  является допустимым решением системы (4.20), поскольку коэффициенты матрицы  $D$  и вектора  $f$  неотрицательны. Для вычисления  $F(0)$  необходимо решить задачу линейного программирования (4.18)—(4.20) при дополнительном условии  $U = 0$ .

Пусть уже выполнено  $t \geq 0$  шагов метода, и зафиксированы значение вектора  $U$ , равное  $U^t$ , и множество ненулевых значений переменных  $S(W^t) = \{W^t(U, \Lambda)\}$ . На  $(t+1)$ -м шаге выполняются следующие действия.

1. По множеству  $S(W^t)$  определяется вектор значений двойственных переменных  $Y^t = Y(W^t)$ , соответствующих системе (4.28).

2. Если  $Y^t$  имеет отрицательные компоненты, то осуществляется преобразование значений  $W^t$  и соответственно множества  $S(W^t)$  таким образом, что значение функционала на преобразованном решении увеличивается, а выбранная отрицательная компонента вектора  $Y^t$  становится нулевой. При этом из  $S(W^t)$ , возможно, исключается одна из переменных (она становится нулевой).

3. Если  $Y^t \geq 0$ , то осуществляется проверка оптимальности текущего решения  $U^t$ . Если оно оптимально, то процесс решения заканчивается, в противном случае определяется вектор  $\Lambda^t$ , которому соответствует переменная  $w(U^t, \Lambda^t)$ , вводимая в базис. При этом значение функционала (в невырожденном случае) увеличивается. Одновременно с выбором переменной вычисляются соответствующие ей компоненты вектора  $Q(U^t, \Lambda^t)$  и скаляра  $r(U^t, \Lambda^t)$ . По этой информации осуществляется шаг симплекс-метода, в результате которого переменная  $w(U^t, \Lambda^t)$  вводится в множество  $S(W^t)$ , переопределяются значения ненулевых переменных и вычисляются новый вектор  $U^{t+1}$  и  $F(U^{t+1})$ . На этом работа метода на шаге  $t+1$  заканчивается.

Прежде чем перейти к детальному описанию каждого этапа на шаге метода, сформулируем двойственную к (4.27)—(4.29) задачу и уточним понятие базиса и базисных переменных для задачи (4.27)—(4.29). Поставим в соответствие ограничениям (4.28) вектор двойственных переменных  $Y \geq 0$ , а ограничениям (4.29) —  $Z \geq 0$ . Из сказанного выше относительно числа переменных в задаче (4.27)—(4.29) следует, что  $Z$  — вектор бесконечной размерности. Задача, двойственная к рассматриваемой, формулируется в следующем виде: найти

$$\min \left( \sum_{k=1}^p f_k Y_k + \sum_{(U, \Lambda)} r(U, \Lambda) z(U, \Lambda) \right) \quad (4.32)$$



при ограничениях

$$\sum_{k=1}^p q_k(U, \Lambda) Y_k + z(U, \Lambda) \geq 1 \quad \forall (U, \Lambda), \quad (4.33)$$

$$Y_k \geq 0, \quad k = 1, \dots, p, \quad (4.34)$$

$$z(U, \Lambda) \geq 0 \quad \forall (U, \Lambda). \quad (4.35)$$

В качестве базиса для задачи (4.27)—(4.29) будем рассматривать подматрицу  $B = (q_k(U, \Lambda))$ , где  $k \in P(W^t)$ ,  $(U, \Lambda) \in N(W^t)$ . Решение задачи (4.32)—(4.35), соответствующее решению (4.27)—(4.29), определяемому множеством  $S(W^t)$ , вычисляется по следующим правилам, являющимся аналогами правил симплекс-метода и базирующимся на теореме о дополняющей нежесткости теории двойственности:

- а) положить  $\bar{Y}_k = 0$  для  $k \notin P(W^t)$ ;
- б) определить значения  $\bar{Y}_k$  ( $k \in P(W^t)$ ) путем решения системы  $Y'B = 1$ , где  $Y'$  — вектор с компонентами из множества  $P(W^t)$ ,  $1$  — единичный вектор требуемой размерности,  $B = (q_k(U, \Lambda))$ ,  $k \in P(W^t)$ ,  $(U, \Lambda) \in N(W^t)$ ;
- в) положить  $\bar{z}(U, \Lambda) = 0$  для  $(U, \Lambda) \notin S(W^t) \setminus N(W^t)$ ;
- г) вычислить значения  $\bar{z} = \max \{0, 1 - \bar{Y}'B(U, \Lambda)\}$ ,  $(U, \Lambda) \in S(W^t) \setminus N(W^t)$ , где  $B(U, \Lambda)$  — соответствующий столбец из  $B$ .

Легко проверить, что для так определенных векторов  $\bar{Y}$ ,  $\bar{z}$  выполнены условия дополняющей нежесткости, однако условия (4.34), (4.35) могут нарушаться. Если для  $\bar{Y}$  нарушено условие (4.34), т. е. для некоторого  $k$   $\bar{Y}_k < 0$ , на очередном шаге выполняется преобразование, эквивалентное шагу симплекс-метода, на котором в базис вводится вспомогательная переменная, соответствующая выбранному неравенству  $k$  из (4.34). При этом из базиса  $B$  исключается строка  $k$  и, возможно, одна из переменных из  $N(W^t)$ . Если множество  $N(W^t)$  не изменяется при переходе к новому базису  $B$ , то в последний вводится одна из строк  $l$  ( $l \notin N(W^t)$ ). Значение функционала на новом решении увеличивается. Аналогичный шаг делается и при  $\bar{z}(U, \Lambda) < 0$ .

Пусть теперь в результате описанных преобразований построены векторы  $\bar{Y} \geq 0$ ,  $\bar{z} \geq 0$ . Для проверки оптимальности текущего решения  $W^t$  задачи (4.27) — (4.29) достаточно найти переменную  $v(U^t, \Lambda)$  (где  $U^t$  — вектор, соответствующий текущему  $W^t$ ), для которой величина  $v = v(U^t, \Lambda)$ , равная  $\sum_{k=1}^p q_k(U^t, \Lambda) \bar{Y}_k$ , минимальна по всем  $(U^t, \Lambda)$ .

Если  $v \geq 1$ , то для пары  $(\bar{Y}, \bar{z})$  выполнены условия допустимости по ограничениям (4.33) — (4.35), и, следовательно, она оптимальна. Это влечет в свою очередь оптимальность текущего решения  $W^t$ . В противном случае переменная, соответствующая выбранному значению  $v$ , вводится в базис. С ней в качестве ведущей осуществляется шаг сим-

плекс-метода, в результате чего строится новое решение  $W^{t+1}$  и вычисляются все требуемые векторы и множества.

Рассмотрим теперь способ отыскания минимума величины  $v$ . Для этого рассмотрим для данного  $U = U^t$  всевозможные увеличивающие векторы  $A(\Lambda)$ , где  $b(\Lambda) > 0$ . Каждому из них соответствует множество положительных компонент из  $J_1$ , т. е. множество  $J_1^+(\Lambda)$ . Поскольку последнее не меняется при замене  $\Lambda$  на  $\delta\Lambda$  при  $\delta > 0$ , можно ввести нормирующее условие  $b(\Lambda) = 1$ , чтобы избавиться от строгого неравенства  $b(\Lambda) > 0$ . Ясно, что при этом, в силу сказанного, ни одно блокирующее множество  $J_1^+(\Lambda)$  не будет исключено из рассмотрения.

В рассматриваемых терминах задача отыскания наименьшего увеличивающего множества  $\Lambda$  и соответствующего ему значения  $v(U^t, \Lambda)$  при данном  $U^t$  формулируется в виде: найти

$$\min \sum_{k=1}^p \bar{Y}_k \sum_{j \in J_1} d_{kj} \max \left\{ 0, \sum_{i=1}^m a_{ij} \lambda_i \right\}$$

при ограничении

$$\sum_{i=1}^m b_i \lambda_i = 1.$$

Введем переменные  $\theta_j (j \in J_1)$  и сформулируем эту задачу в терминах линейного программирования: найти

$$\min \sum_{k=1}^p \bar{Y}_k \sum_{j \in J_1} d_{kj} \theta_j \quad (4.36)$$

при ограничениях

$$\theta_j \geq \sum_{i=1}^m a_{ij} \lambda_i, \quad \theta_j \geq 0, \quad j \in J_1, \quad (4.37)$$

$$\sum_{i=1}^m b_i \lambda_i = 1. \quad (4.38)$$

Рассмотрим задачу, двойственную к (4.36) — (4.38): найти

$$\max \psi \quad (4.39)$$

при ограничениях

$$\sum_{j \in J_1} a_{ij} x_j = \psi b_i, \quad i = 1, \dots, m, \quad (4.40)$$

$$0 \leq x_j \leq \sum_{k=1}^p d_{kj} \bar{Y}_k, \quad j \in J_1 = J(U^t). \quad (4.41)$$

При  $\psi = 1$  ограничения (4.40), (4.41) совпадают с (4.15) — (4.17) при  $Y = \bar{Y}$ . Отметим также, что минимум (4.36) не превосходит единицы потому что по построению вектора  $\bar{Y}$  имеет место следующая альтернатива:  $\bar{Y}'B = 1$  или  $Y = 0$ , где  $\bar{Y}'$  — множество ненулевых компонент

$\bar{Y}$  из  $P(W^t)$ . Вспоминая, что элементы матрицы  $B$  вычисляются по (4.30), получаем, что для любого вектора  $\bar{\Lambda}$ , формирующего один (любой) из столбцов матрицы  $B$ , имеет место следующий факт: значение функционала (4.36) при  $\Lambda = \bar{\Lambda}$  равно единице. Следовательно,  $\max \psi \leq 1$ .

Пусть задача (4.39)—(4.41) разрешима при  $\psi = 1$  и  $\bar{X} = (\bar{x}_j)$  ( $j \in J_1$ ) — ее допустимое решение.

**Теорема 4.6.** *Пара  $(X^*, Y^*)$ , где  $x_j^* = \bar{x}_j$ ,  $j \in J_1$ ,  $x_j^* = 0$ ,  $j \notin J_1$ ,  $Y^* = \bar{Y}$ , является оптимальным решением задачи (4.14)—(4.17).*

**Доказательство.** Поскольку  $\max \psi = 1$ , текущее решение  $W^* = W^t$  оптимально. Следовательно, оптимально и порожденное им решение  $U^t$ , а также вектор двойственных переменных  $Y^* = \bar{Y}$ , соответствующих ограничениям (4.28) и эквивалентным им ограничениям (4.20). Далее, вектор  $X^*$  удовлетворяет (4.40), (4.41) и, что эквивалентно, (4.15)—(4.17), а  $Y^*$  — ограничениям (4.16), (4.17). Из способа построения множества  $J_1$  следует, что для ненулевых компонент вектора  $X^*$  выполняются условия дополняющей нежесткости  $X^* (c + U^t - V^* (U^t) A) = 0$ , откуда следует оптимальность  $X^*$ .

Пусть  $\max \psi < 1$  и  $\bar{X}$  — оптимальное решение задачи (4.40), (4.41). Тогда, выражая компоненты вектора  $\Theta = (\theta_j)$  из условий дополняющей нежесткости

$$\theta_j = 0, \quad \text{если} \quad \bar{x}_j < \sum_{k=1}^p d_{kj} \bar{y}_k, \quad (4.42)$$

$$\theta_j = \sum_{i=1}^m a_{ij} \lambda_i, \quad \text{если} \quad \bar{x}_j > 0, \quad (4.43)$$

и подставляя их в (4.36), (4.37), получим для определения компонент наилучшего увеличивающего вектора  $\Theta \equiv A(\Lambda)$  задачу следующего вида: найти

$$\min \left\{ \sum_{i=1}^m \alpha_i \lambda_i : \sum_{i=1}^m a_{ij} \lambda_i = 0, \quad j \in J_1^+(\Lambda), \quad \sum_{i=1}^m b_i \lambda_i = 1 \right\}.$$

Здесь  $J_1^+(\Lambda) = \left\{ j : \bar{x}_j = \sum_{k=1}^p d_{kj} \bar{y}_k \right\}$  и совпадает с ранее введенным и аналогично обозначенным множеством;  $\alpha_i$  — коэффициенты при  $\lambda_i$ , полученные после подстановки (4.43) в (4.36) и приведения подобных членов. Решив полученную задачу и подставив ее решение в (4.43), найдем окончательные значения компонент вектора  $\Theta^* = A(\Lambda^t)$ . Зная вектор  $\Lambda^t$ , соответствующий переменной  $\omega(U^t, \Lambda^t)$ , по формулам (4.26) и (4.30) вычислим значения  $r(U^t, \Lambda^t)$  и  $Q(U^t, \Lambda^t)$ , необходимые для выполнения следующего шага симплекс-метода в задаче (4.27)—(4.29). В результате будут построены новое решение  $W^{t+1}$  и соответствующий ему вектор  $U^{t+1}$ , а также все необходимые для дальнейших преобразований данные.

Описанный выше метод динамической декомпозиции решения задачи (4.14)—(4.17) требует, таким образом, решения на каждой итерации симплекс-метода вспомогательной задачи вида (4.36)—(4.38) или двойственной к ней (4.39)—(4.41). Естественно, что его эффективность существенно зависит от эффективности методов решения вспомогательной задачи. В следующем разделе рассматриваются свойства этой задачи для случая, когда система (4.15) условий исходной задачи является системой ограничений сетевой транспортной задачи.

#### § 4. Динамическая декомпозиция и сетевая транспортная задача

Пусть система ограничений (4.15) является ограничениями сетевой транспортной задачи на графе  $G(N, E)$  с множеством вершин  $N$  и дуг  $E$ . Перепишем задачу (4.14)—(4.17) для этого случая в сетевых терминах, интерпретируя при этом переменные  $x_{ij}$  ( $(i, j) \in E$ ) как потоки по дугам  $(i, j)$ . Каждой вершине сети поставлена в соответствие ее потребность  $b_i$ ,  $\sum_{i \in N} b_i = 0$ , а каждой дуге  $(i, j) \in E$  — неотрицательный вес  $c_{ij}$ . Наконец, как и в (4.14)—(4.17), заданы матрица  $D = (d_{ijk})$  размерности  $|E| \times p$  с неотрицательными элементами и вектор  $\bar{f} = (f_k)$  ( $k = 1, \dots, p$ ). В этих терминах задача переписывается в следующем виде: найти

$$\min \left( \sum_{(i,j) \in E} c_{ij} x_{ij} + \sum_{k=1}^p f_k y_k \right) \quad (4.44)$$

при ограничениях

$$\sum_{j \in P(i)} x_{ij} - \sum_{j \in Q(i)} x_{ji} = b_i, \quad i \in N, \quad (4.45)$$

$$0 \leq x_{ij} \leq \sum_{k=1}^p d_{ijk} y_k, \quad (i, j) \in E, \quad (4.46)$$

$$y_k \geq 0, \quad k = 1, \dots, p; \quad (4.47)$$

здесь  $P(i)$  и  $Q(i)$  — множества входящих в  $i$  и исходящих из  $i$  дуг соответственно.

Вспомогательная задача (4.36)—(4.38) записывается для рассматриваемого случая в форме: найти

$$\min \sum_{(i,j) \in E(\bar{U})} \bar{d}_{ij} \theta_{ij} \quad (4.48)$$

при ограничениях

$$\theta_{ij} \geq \lambda_j - \lambda_i, \quad \theta_{ij} \geq 0, \quad (i, j) \in E(\bar{U}), \quad (4.49)$$

$$\sum_{i \in N} b_i \lambda_i = 1. \quad (4.50)$$

Здесь  $E(\bar{U}) = \{(i, j) \in E: v_j^*(\bar{U}) - v_i^*(\bar{U}) = c_{ij} + \bar{u}_{ij}\}$ ,  $V^*(\bar{U})$  — вектор оптимальных значений двойственных переменных для данного  $U = \bar{U}$ , соответствующих (4.45),  $\bar{d}_{ij} = \sum_{k=1}^p \bar{d}_{ijk} \bar{y}_k$ ,  $(i, j) \in E(\bar{U})$ .

Пусть  $\Theta^* = \{\theta_{ij}^*\}$  — оптимальное решение задачи (4.48) — (4.50).

**Теорема 4.7.** *Каждая ненулевая компонента вектора  $\Theta^*$  равна  $c$ , где  $c$  — некоторая константа.*

**Доказательство.** Рассмотрим грань многогранного множества (4.49), (4.50), содержащую точку  $\Theta^*$ . Она определяется системой уравнений, состоящей из (4.50) и

$$\theta_{ij} = 0, \quad (i, j) \in E_0, \quad (4.51)$$

$$\theta_{ij} = \lambda_j - \lambda_i, \quad (i, j) \in E_1, \quad (4.52)$$

где  $E_0 \cup E_1 = E(\bar{U})$ . Подставляя (4.51), (4.52) в (4.48), преобразуем задачу (4.48) — (4.50) к виду: найти

$$\min \sum_{i \in V} d_i \lambda_i \quad (4.53)$$

при ограничениях (4.50) и

$$\lambda_j - \lambda_i = 0, \quad (i, j) \in E_0 \cap E_1. \quad (4.54)$$

Из теории транспортных многогранников известно, что система (4.45) линейно зависима, откуда следует, что одну из двойственных переменных  $\lambda_i$ , скажем  $\lambda_1$ , можно положить равной нулю. Подставим  $\lambda_1 = 0$  в (4.50), (4.53), (4.54). Кроме того, выразим поочередно некоторые  $\lambda_i$  из уравнений (5.54), подставляя их выражения в (4.50), (4.53). Будем повторять эти преобразования до тех пор, пока это возможно. При этом множество исключенных переменных разобьется на два подмножества  $N_0$  и  $N_1$  таких, что  $\lambda_i = 0$ ,  $i \in N_0$ ,  $\lambda_i = \lambda_k$ ,  $i \in N_1$ , где  $k \in N_2 = N \setminus (N_0 \cup N_1)$ . Если  $|N_2| < 1$ , то уравнение (4.50) несовместно. Если  $|N_2| > 1$ , то преобразованная путем исключения переменных задача (4.50), (4.53) имеет неограниченное решение, что противоречит факту ограниченности (4.48) на множестве оптимальных решений (так как все  $\bar{d}_{ij} \geq 0$ ,  $b_i \geq 0$ ). Поэтому  $|N_2| = 1$ , т. е.  $N_2 = \{k\}$ , где  $k$  — некоторый индекс из  $N$ . Следовательно, уравнение (4.50) преобразовано к виду  $\bar{b}_k \lambda_k = 1$ ,  $\bar{b}_k > 0$ . Отсюда  $\lambda_i = \lambda_k = \bar{b}_k^{-1} = c$  ( $i \in V_1$ ). Подставляя полученные значения в (4.52), получаем, с учетом неотрицательности компонент  $\theta_{ij}^*$ , требуемый результат.

Вектор  $\Theta^*$  определяет направление наибольшего возрастания функционала  $F(U)$  в точке  $U = \bar{U}$ , которое остается таковым и при нормировке  $\sum_{i=1}^m b_i \lambda_i = \bar{b}_k$ , откуда следует  $c = 1$ . Комбинируя этот факт с пер-

вой частью теоремы 4.5 и теоремой 4.7, получаем следующий результат.

**Теорема 4.8.** Вектор  $\Theta^*$  совпадает с носителем одного из блокирующих множеств к  $X(\bar{U})$ .

Задача, двойственная к (4.48)—(4.50), формулируется как задача отыскания

$$\max \psi \quad (4.55)$$

при ограничениях

$$\sum_{i \in P(i)} x_{ij} - \sum_{j \in Q(i)} x_{ji} = \psi b_i, \quad i \in N, \quad (4.56)$$

$$0 \leq x_{ij} \leq \bar{d}_{ij}, \quad (i, j) \in E(\bar{U}). \quad (4.57)$$

Покажем, что решение этой задачи можно свести к решению последовательности задач о максимальном потоке [3] на графе  $G(N', E'(\bar{U}))$ , где  $N' = N \cup \{s, t\}$ ,  $s$  — дополнительная вершина-источник, связанная исходящими из нее дугами  $(s, i)$  пропускной способности  $\psi|b_i|$  с теми  $i \in N$ , для которых  $b_i < 0$ , а  $t$  — дополнительная вершина-сток, связанная входящими в нее дугами  $(i, t)$  пропускной способности  $\psi b_i$  с теми  $i \in N$ , для которых  $b_i > 0$ ,  $E'(\bar{U}) = E(\bar{U}) \cup \{(s, i)\} \cup \{(i, t)\}$ . Пропускные способности дуг  $(i, j) \in E(\bar{U})$  полагаются равными  $\bar{d}_{ij} \geq 0$ . Как показано в § 3, величина  $\psi^* = \max \psi$  не превосходит единицы. Поэтому начнем процесс решения задачи (4.55)—(4.57) с решения задачи о максимальном потоке на соответствующей сети при  $\psi = 1$ .

Пусть  $\{x_{ij}\} ((i, j) \in E'(\bar{U}))$  — величины дуговых потоков в решении; тогда, если максимальный поток равен  $b = \sum_{i: b_i > 0} b_i$ , то  $\max \psi = 1$ , и

процесс решения закончен. В противном случае организуем дихотомический поиск  $\max \psi$  на отрезке  $[0, 1]$ , в котором на очередном шаге выбирается значение  $\psi$ , равное середине текущего отрезка. Если при текущем значении  $\psi$  в решении задачи о максимальном потоке в графе  $G(N', E'(\bar{U}))$  с пропускными способностями исходящих из  $s$  и входящих в  $t$  дуг, равными  $\psi|b_i|$ , максимальный поток равен  $b\psi$ , то на следующем шаге в качестве нового рассматривается правая половина текущего отрезка, в противном случае — левая. Таким образом, на каждом шаге поиск осуществляется на некотором отрезке  $[\psi_0, \psi_1]$ , который обладает следующим свойством: при  $\psi = \psi_0$  система (4.56), (4.57) разрешима, при  $\psi = \psi_1$  — неразрешима.

Покажем, что описанный метод решает задачу (4.55)—(4.57) за конечное число шагов в предположении, что все  $b_i, \bar{d}_{ij}$  рациональны. Пусть  $q$  — наименьшее общее кратное чисел  $s_{ij} ((i, j) \in E(\bar{U}))$ ,  $r_i (i \in N)$ , где  $\bar{d}_{ij} = r_{ij}/s_{ij} ((i, j) \in E(\bar{U}))$ ,  $b_i = r_i/s_i (i \in N)$  — представление исходных данных в виде несократимых дробей. Тогда оптимальное значение  $\psi$  представимо в виде несократимой дроби  $p/q$ . Этот факт следует из того, что двойственный функционал (4.48) на оптимальном

решении по теореме 4.7 равен  $\sum \frac{d_{ij}}{b_i}$ , где суммирование проводится

только по ненулевым  $\Theta_{ij}$ .

Пусть  $[\psi_0, \psi_1]$  — текущий отрезок, построенный на некотором шаге дихотомического поиска.

**Л е м м а 4.3.** Если  $|\psi_1 - \psi_0| < q^{-1}$ , то  $\psi^* = p/q$ , где  $\psi_0 \leq p/q < \psi_1$ , — оптимальное значение  $\psi$ .

Доказательство следует из того, что отрезок  $[\psi_0, \psi_1]$  содержит единственную точку вида  $p/q$ .

Таким образом, для построения оптимального значения  $\psi^*$  необходимо решить не более  $(\log_2 q + 1)$  задач о максимальном потоке. Трудоемкость последней задачи растет как  $O(n^3)$ , где  $n$  — число вершин графа [3]. Отсюда, общая трудоемкость решения вспомогательной задачи (4.55) — (4.57) оценивается величиной  $O(n^3 \log_2 q)$ . Зная оптимальное решение этой задачи, нетрудно за время, не превосходящее  $O(n^3)$ , построить оптимальный вектор  $\Theta^*$  для задачи (4.48) — (4.50), используя для этого стандартный способ решения двойственной задачи по решению прямой. Отметим также, что для построения увеличивающего вектора не обязательно построение оптимального решения  $\Theta^*$ . Достаточно найти вектор  $\Theta$ , для которого значение (4.48) строго меньше единицы. Для этого можно воспользоваться следующей процедурой. Пусть  $\psi_k$  — значение  $\psi$  на  $k$ -м шаге описанного выше алгоритма отыскания  $\psi^*$ ,  $X^k = \{x_{ij}^k\}$  — значения потоков на дугах  $(i, j) \in E(\bar{U})$ , полученные при решении задачи о максимальном потоке на  $k$ -м шаге. Построим систему (4.51), (4.52), где  $E_0 = \{(i, j) : x_{ij}^k < \bar{a}_{ij}\}$ ,  $E_1 = \{(i, j) : x_{ij}^k > 0\}$ . Если полученная таким способом система (4.53), (4.54) совместна и имеет решение  $\Lambda^k$ , а значение (4.48), полученное путем подстановки  $\Theta^k = \{\theta_{ij}^k\}$ , где  $\theta_{ij}^k = \max\{0, \lambda_j^k - \lambda_i^k\}$  ( $(i, j) \in E(\bar{U})$ ), меньше единицы, то, полагая  $\bar{\Theta} = \Theta^k$ , закончим процесс решения вспомогательной задачи. В противном случае выполняется  $(k + 1)$ -й шаг алгоритма отыскания  $\psi^*$ .

Кроме описанного способа отыскания  $\psi^*$  можно попытаться построить алгоритм непосредственного решения задачи (4.55) — (4.57) с полиномиальной оценкой трудоемкости, которая, возможно, будет ниже, чем в предложенном методе. Однако такой путь пока не рассматривался.

При решении многих задач размещения и синтеза сетей, описанных в § 1, задача (4.44) — (4.47) имеет еще более специальную структуру, а именно система условий (4.45) представима в виде нескольких непересекающихся транспортных блоков, каждый из которых определяется условиями задачи о кратчайшем пути между заданной парой  $(r, s) \in A$  вершин графа  $G(N', E')$ . Последний в свою очередь является объединением  $|A|$  копий одного и того же графа  $G(N, E)$ . Иными словами, задача (4.44) — (4.47) имеет следующий вид: найти

$$\min \left( \sum_{(r,s) \in A} \sum_{(i,j) \in E} c_{ij} x_{ij}^{rs} + \sum_{k=1}^p f_k y_k \right) \quad (4.58)$$

при ограничениях

$$\sum_{i \in P(i)} x_{ij}^{rs} - \sum_{i \in Q(i)} x_{ji}^{rs} = \begin{cases} -1 & \text{при } i = r, \\ 0 & \text{при } i \neq r, s, \quad i \in N, \quad (r, s) \in A, \\ 1 & \text{при } i = s, \end{cases} \quad (4.59)$$

$$0 \leq x_{ij}^{rs} \leq \sum_{k=1}^p d_{ijh}^{rs} y_k, \quad (i, j) \in E, \quad (r, s) \in A, \quad (4.60)$$

$$y_k \geq 0, \quad k = 1, \dots, p. \quad (4.61)$$

Решение вспомогательной задачи (4.55)—(4.57) применительно к (4.58)—(4.61) сводится к отысканию минимума из максимальных потоков по парам  $(r, s)$  из  $A$ , поскольку каждый блок ограничений (4.59) в данном случае в точности соответствует ограничениям неразрывности потока в задаче о максимальном потоке для соответствующей пары  $(r, s) \in A$ . Трудоемкость решения задачи (4.55)—(4.57) при этом оценивается величиной  $O(|A| \cdot n^3)$ , где  $n$  — число вершин в каждой копии  $G(N, E)$ . Процесс решения вспомогательной задачи может быть закончен, как только решена задача о максимальном потоке для некоторой пары  $(r, s)$ , для которой максимальный поток оказался меньше единицы.

При решении задачи вида (4.58)—(4.61) можно обойтись без хранения симплекс-таблицы матрицы условий (4.28) и ее представления в виде (4.30). Необходимые данные для построения вектора условий  $Q(U, \Lambda)$ , соответствующих переменной  $w(U, \Lambda)$ , вводимой в базис, можно получить в соответствии с теоремой 4.8 путем определения носителя минимального разреза по решению задачи о максимальном потоке для выбранного блока  $(r, s) \in A$ . Поскольку в ограничениях (4.59) единственное  $b_i$  ( $i = r$ ) положительно и равно единице, то и все ненулевые компоненты оптимального решения задачи (4.48)—(4.50), равные по теореме 4.7 величине  $c = b_i^{-1}$ , равны в данном случае единице. Таким образом, минимальный разрез полностью определяет вектор  $Q(U, \Lambda)$ . Для определения значения  $r(U, \Lambda)$  — верхней грани для рассматриваемой переменной — достаточно решить задачу о кратчайшем пути из  $r$  в  $s$  на графе  $G(N, E \setminus R(U, \Lambda))$  с весами дуг  $c + U$ , где  $R(U, \Lambda)$  — минимальный разрез, рассмотренный выше. Обозначим длину кратчайшего пути  $L(U, \Lambda)$ . Пусть  $l(U, \Lambda)$  — длина кратчайшего пути для тех же  $r, s$  на графе  $G(N, E)$  с весами дуг  $c + U$ . Тогда

$$r(U, \Lambda) = L(U, \Lambda) - l(U, \Lambda).$$

Поскольку трудоемкость определения длины кратчайшего пути есть  $O(n^2)$ , то общая трудоемкость шага симплекс-метода, складывающаяся из трудоемкостей решения  $|A|$  задач о максимальном потоке, задачи о кратчайшем пути и трудоемкости симплекс-преобразований базиса задачи (4.58)—(4.61), оценивается величиной, не превосходящей  $O(p^2 + |A|n^3)$ .

Описанный метод динамической декомпозиции может быть использован для приближенного решения задачи (4.58)—(4.61). Величина



$F(U)$  в этом случае является нижней оценкой для функционала  $\Phi(X, Y)$  на оптимальном решении задачи (4.58)—(4.61). В следующей главе такой подход рассмотрен применительно к ряду частных случаев задачи (4.58)—(4.61). Как отмечалось выше, блокирующие множества (минимальные разрезы) в таких случаях полностью определяют выбор переменных, включаемых в множество  $S(W^t)$  ненулевых переменных на  $t$ -м шаге описанного алгоритма. Поэтому именно в терминах блокирующих множеств описаны схемы предлагаемых далее алгоритмов.

Рассмотренный в этой главе метод динамической декомпозиции применительно к задаче (4.14)—(4.17) допускает обобщения в ряде направлений. Например, ограничение (4.17) можно заменить на  $HX \leq \leq DY$ , где  $H$  — произвольная  $(0, 1)$ -матрица. Некоторые примеры таких задач также приведены в следующей главе.

## ЗАДАЧИ РАЗМЕЩЕНИЯ ПРОИЗВОДСТВА И СИНТЕЗА СЕТЕЙ И МЕТОДЫ ИХ РЕШЕНИЯ

---

В этой главе мы рассмотрим некоторые типичные примеры задач размещения производства и синтеза сетей, их свойства и методы решения. Рассматриваемые задачи формулируются в виде моделей целочисленного линейного программирования. Для соответствующих линейных релаксаций детализируется метод динамической декомпозиции, описанный в гл. 4. Затем для декомпозированных задач излагается метод точного или приближенного их решения. Полученное таким образом значение функционала двойственной задачи используется в качестве нижней оценки для функционала исходной задачи, а ее решение — для построения приближенного целочисленного решения исходной задачи, что в свою очередь дает оценку сверху для оптимального значения функционала исходной задачи. Полученные нижние и верхние оценки используются в схеме ветвей и границ для построения оптимального решения или приближенного решения с оценками его отклонения от оптимального.

### § 1. Простейшая задача размещения производства: свойства и построение оценок

Простейшая задача размещения производства однородного продукта заключается в отыскании

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n g_{ij} x_{ij} + \sum_{i=1}^m f_i(X_i) \right) \quad (5.1)$$

при ограничениях

$$\sum_{i=1}^m x_{ij} = B_j, \quad j = 1, \dots, n, \quad (5.2)$$

$$\sum_{j=1}^n x_{ij} = X_i, \quad i = 1, \dots, m, \quad (5.3)$$

$$x_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (5.4)$$

где  $f_i(X_i)$  — функции с фиксированными доплатами вида

$$f_i(X_i) = \begin{cases} a_i X_i + b_i & \text{при } X_i > 0, \\ 0 & \text{при } X_i = 0. \end{cases} \quad a_i, b_i \geq 0, \quad (5.5)$$

Задача (5.1)—(5.5) исследовалась в многочисленных работах с различных точек зрения. В частности, В. П. Черенин [80] и В. П. Черенин и В. Р. Хачатуров [81] разработали для ее решения метод последовательных расчетов. М. А. Эфроимсон и Т. Л. Рей [129] и другие применяли для ее решения метод ветвей и границ, С. С. Лебедев и М. И. Ковалевская [43] — метод множителей Лагранжа в сочетании с последовательным перебором. В работе [69] для нее предложен метод ветвей и границ в сочетании с методом декомпозиции для решения релаксированной задачи. Близкие подходы были позднее описаны в [10, 113]. Эта задача является наиболее простой среди задач размещения, но сохраняет все принципиальные трудности, возникающие при их решении.

Задача (5.1)—(5.5) эквивалентна задаче частично целочисленного программирования следующего вида: найти

$$\min F(Y, y) = \min \left( \sum_{i=1}^m \sum_{j=1}^n c_{ij} y_{ij} + \sum_{i=1}^m b_i Y_i \right) \quad (5.6)$$

при ограничениях

$$\sum_{i=1}^m y_{ij} = 1, \quad j = 1, \dots, n, \quad (5.7)$$

$$0 \leq y_{ij} \leq Y_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (5.8)$$

$$Y_i = 0 \vee 1, \quad i = 1, \dots, m; \quad (5.9)$$

здесь все  $c_{ij} = (g_{ij} + a_i) B_j$ .

В дальнейшем простейшая задача размещения рассматривается в форме (5.6)—(5.9). Мы начнем с рассмотрения метода динамической декомпозиции применительно к задаче (5.6)—(5.8).

Поставим в соответствие ограничениям (5.7), (5.8) двойственные переменные  $u_j, v_{ij}$  ( $i = 1, \dots, m, j = 1, \dots, n$ ). Задача, двойственная к (5.6)—(5.8), формулируется в виде: найти

$$\max \sum_{j=1}^n u_j \quad (5.10)$$

при ограничениях

$$\sum_{j=1}^n v_{ij} \leq b_i, \quad i = 1, \dots, m, \quad (5.11)$$

$$u_j \leq c_{ij} + v_{ij}, \quad v_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (5.12)$$

Введем упорядочение строк матрицы  $C = (c_{ij})$  для каждого ее столбца  $j$  ( $j = 1, \dots, n$ ):

$$i <_j k, \Leftrightarrow (c_{ij} < c_{kj}) \vee ((c_{ij} = c_{kj}) \wedge (i < k)), \\ i \neq k, \quad i, k = 1, \dots, m, \quad j = 1, \dots, n.$$

Иными словами, строка  $i$  предшествует строке  $k$  в упорядочении по столбцу  $j$ , если либо  $c_{ij} < c_{kj}$ , либо  $c_{ij} = c_{kj}$  и  $i < k$ .

Порядковые номера строк в упорядочении будем обозначать  $r(j)$ . Если ясно, о каком столбце идет речь, индекс  $j$  у  $r(j)$  будем опускать. Через  $(U^*, V^*)$  обозначим оптимальное решение задачи (5.10)—(5.12).

Метод динамической декомпозиции для задачи (5.6)—(5.9) можно получить из несколько иных соображений, чем в рассмотренной ранее общей задаче. Для изложения такого подхода сформулируем вначале некоторые утверждения относительно рассматриваемой двойственной задачи, которые позволят преобразовать ее к требуемому виду.

**Т е о р е м а 5.1.** *Существует оптимальное решение  $(U^*, V^*)$  задачи (5.10)—(5.12), удовлетворяющее условиям*

$$u_j^* = c_{1(j)j} + v_{1(j)j}^*, \quad j = 1, \dots, n, \quad (5.13)$$

$$c_{rj} + v_{rj}^* \leq c_{r+1j} + v_{r+1j}^*, \quad r = 1, \dots, m-1, \quad j = 1, \dots, n, \quad (5.14)$$

$$v_{rj}^* \geq v_{r+1j}^*, \quad v_{mj}^* \geq 0, \quad r=1, \dots, m-1, \quad j = 1, \dots, n. \quad (5.15)$$

**Д о к а з а т е л ь с т в о.** Для каждого  $j$  найдем индекс  $s(j)$  такой, что

$$c_{sj} + v_{sj}^* = \min_{1 \leq r \leq m} (c_{rj} + v_{rj}^*) = d(j), \quad (5.16)$$

и положим  $\tilde{v}_{rj} = \max \{0, d(j) - c_{rj}\}$  ( $r = 1, \dots, m, j = 1, \dots, n$ ). Тогда очевидно, что

1)  $0 \leq \tilde{v}_{rj} \leq v_{rj}^*$  ( $r = 1, \dots, m, j = 1, \dots, n$ ) и, следовательно,  $\tilde{V}$  удовлетворяет ограничениям (5.11);

2) для  $\tilde{V}$  выполнены ограничения (5.14), (5.15) (это следует из способа упорядочения элементов  $c_{rj}$  матрицы  $C$ );

3) для  $(U^*, \tilde{V})$  выполнены ограничения (5.13) (это также следует из способа упорядочения  $C$  и оптимальности  $(U^*, V^*)$ );

4) Решение  $(U^*, \tilde{V})$  оптимально, так как значение (5.10) для него совпадает с соответствующим значением для  $(U^*, V^*)$ .

Поскольку условия (5.13)—(5.15) сильнее, чем условия (5.12), то вместо задачи (5.10)—(5.12) можно рассматривать более сильную задачу (5.10), (5.11), (5.13)—(5.15), оптимальное решение которой является оптимальным и для (5.10)—(5.12). Введем новые переменные  $\omega_{rj}$  ( $r = 1, \dots, m, j = 1, \dots, n$ ), которые связаны со старыми  $v_{rj}$  следующим преобразованием:

$$v_{rj} = \sum_{s=r}^m \omega_{sj}, \quad r = 1, \dots, m, \quad j = 1, \dots, n. \quad (5.17)$$

Исключая переменные  $u_j$  с помощью формул (5.13), получим с учетом 5.16) следующую задачу: найти

$$\max \Phi(W) = \max \sum_{r=1}^m \sum_{j=1}^n \omega_{rj} + c_0 \quad (5.18)$$

при ограничениях

$$\sum_{r=1}^m \sum_{j=1}^n a_{irj} \omega_{rj} \leq b_i, \quad i = 1, \dots, m, \quad (5.19)$$

$$0 \leq \omega_{rj} \leq d_{rj}, \quad r = 1, \dots, m, \quad j = 1, \dots, n. \quad (5.20)$$

Здесь  $c_0 = \sum_{i=1}^n c_{1(i)}$ ,  $A = (a_{irj})$ , где

$$a_{irj} = \begin{cases} 1, & \text{если } i <_i r, \quad i, r = 1, \dots, m, \quad j = 1, \dots, n, \\ 0, & \text{иначе.} \end{cases} \quad (5.21)$$

Покажем, что задача (5.18) — (5.20) является по отношению к (5.10)—(5.12) декомпозированной, т. е. она соотносится с (5.10)—(5.12), как (4.27)—(4.29) с (4.18)—(4.20). Для этого рассмотрим задачу, двойственную к (5.6)—(5.8): найти

$$\max_{V \geq 0} \min_{Y \geq 0} \sum_{i=1}^m \sum_{j=1}^n (c_{ij} + v_{ij}) y_{ij}$$

при ограничениях

$$\begin{aligned} \sum_{i=1}^m y_{ij} &= 1, \quad j = 1, \dots, n, \\ \sum_{j=1}^n v_{ij} &\leq b_i, \quad i = 1, \dots, m. \end{aligned}$$

Пусть  $\bar{V} \geq 0$  — произвольное оптимальное решение; тогда  $\bar{v}_{ij} > 0$ , если  $c_{ij} + \bar{v}_{ij} = d(j)$  ( $i = 1, \dots, m, j = 1, \dots, n$ ), где  $d(j)$  определяется как в формуле (5.16). При этом  $c_{ij} + \bar{v}_{ij} = c_{1(j)} + \bar{v}_{1(j)}$  для  $j = 1, \dots, n$ .

Легко понять, что блокирующими множествами  $\beta(\bar{V})$  являются множества индексов  $i$  из некоторого столбца  $j$  ( $j = 1, \dots, n$ ), для которых имеет место

$$c_{ij} + \bar{v}_{ij} = d(j), \quad (5.22)$$

и только они. Действительно, увеличение значений  $\bar{v}_{ij}$  для любого  $j$  и соответствующего множества индексов  $i$  из  $\beta(\bar{V})$  увеличивает значение функционала. С другой стороны, увеличение компонент вектора  $\bar{V}$  по любому подмножеству, не содержащему множества  $\beta(\bar{V})$ , не увеличивает функционал, так как базисным решениям внутренней задачи (минимизации по  $Y$ ) соответствуют наборы  $(i(j), j)$ , для которых  $c_{i(j)} + \bar{v}_{i(j)} = d_j$  ( $j = 1, \dots, n$ ), а в этом случае найдется решение  $\bar{Y}$ , не пересекающееся по единичным компонентам с рассматриваемым подмножеством. Для любого  $\bar{V}$  увеличение его компонент по блокирующему множеству ведет к увеличению функционала до тех пор, пока множество индексов  $i$ , для которых выполнено (5.22), не расширится.

При этом равенство (5.22) будет выполняться для порядкового номера  $r + 1$  такого, что  $i(r + 1)$  не принадлежит рассматриваемому блокирующему множеству и  $r + 1$  — минимальный номер с таким свойством. Здесь  $i(r + 1)$  — индекс строки, имеющей порядковый номер  $r + 1$  в упорядочении  $\underset{j}{\succ}$ . Если обозначить приращение функционала

при такой операции через  $w_{r,j}$ , то ясно, что в совокупности все  $w_{r,j}$  должны удовлетворять ограничениям (5.19), (5.20), а значения целевых функций (5.10) и (5.18) на оптимальных решениях соответствующих им задач совпадают.

Для завершения доказательства эквивалентности задач (5.18)—(5.20) и (4.27)—(4.29) покажем, что имеют место соотношения типа (4.31).

**Теорема 5.2.** *Существует такое оптимальное решение  $W^*$  задачи (5.18)—(5.20), что для всех  $j = 1, \dots, n$*

а) из  $w_{rj}^* > 0$  следует  $w_{sj}^* = d_{sj}$  для  $1 \leq s < r$ ,

б) из  $w_{rj}^* < d_{rj}$  следует  $w_{sj}^* = 0$  для  $r < s \leq m$ .

**Доказательство.** Непосредственно следует из того, что все переменные  $w_{r,j}$  входят в целевую функцию с одинаковыми коэффициентами, и того, что  $a_{r,j} \leq a_{r+1,j}$  ( $r = 1, \dots, m - 1, j = 1, \dots, n$ ). Здесь  $a_{r,j}$  — вектор-столбец матрицы условий  $A$ , соответствующий переменной  $w_{r,j}$ . Действительно, пусть  $W^*$  — оптимальное решение, не удовлетворяющее, например, а). Это значит, что имеется пара  $(s, j)$  такая, что  $w_{sj}^* < d_{sj}$ . Положим  $\bar{w}_{sj} = w_{sj}^* + \varepsilon$ ,  $\bar{w}_{rj} = w_{rj}^* - \varepsilon$ ,  $\bar{w}_{kl} = w_{kl}^*$  при  $l = j, k \neq r, s$  или при  $l \neq j$  и выберем  $\varepsilon$  равным  $\min\{w_{rj}^*, d_{sj} - w_{sj}^*\}$ . Тогда  $\bar{W}$  — допустимое и оптимальное решение. Если для  $\bar{W}$  не выполнено а), повторим с ним аналогичное преобразование. Через конечное число шагов получим решение  $\hat{W}$ , удовлетворяющее а). Аналогично доказывается справедливость условий б).

Теорема 5.2 позволяет существенно уменьшить трудоемкость вычислений на шаге симплекс-метода при решении задачи (5.18)—(5.20).

Рассмотрим схему этого метода применительно к задаче (5.18)—(5.20). Начнем с одного из возможных алгоритмов построения начального тупикового решения (НТР), который просматривает переменные  $w_{r,j}$  в порядке циклического изменения значения индекса  $j$ , начиная с  $j = 1$  и полагая затем  $j := j + 1 \pmod{n}$ .

При текущем значении  $j$  выбирается не фиксировавшаяся на предыдущих шагах переменная  $w_{r,j}$ , соответствующая минимальному по включению столбцу  $a_{r,j}$  матрицы  $A$ ; иными словами, столбцу с минимальным среди нефиксированных переменных индексом упорядочения  $r$  и переменной  $w_{r,j}$  присваивается максимально допустимое (с учетом фиксации просмотренных ранее переменных) значение  $\bar{w}_{r,j}$ . Обозначим:  $N$  — счетчик просмотренных алгоритмом столбцов матрицы  $C$ ,  $r = r(j)$  — индекс рассматриваемого на текущем шаге элемента  $c_{ij}$  в упорядочении  $\underset{j}{\prec}$ ,  $i(r)$  — номер строки, соответствующей данному  $r$ .

На выходе алгоритм НТР определяет:  $\Phi$  — значение функционала на

построенном решении,  $DW = (DW(j))$  —  $n$ -мерный вектор,

$$DW(j) = \begin{cases} \bar{\omega}_{rj}, & \text{если } \exists r: 0 < \bar{\omega}_{rj} < d_{rj}, \quad j = 1, \dots, n, \\ 0 & \text{в противном случае,} \end{cases}$$

$$IP = (IP(j)) \text{ — } n\text{-мерный вектор,}$$

$$IP(j) = \begin{cases} r, & \text{если } 0 < \bar{\omega}_{rj} < d_{rj}, \quad j = 1, \dots, n, \\ 0 & \text{в противном случае,} \end{cases}$$

$$\bar{b} = b - A\bar{W}.$$

По  $DW$ ,  $IP$  построенное решение  $\bar{W}$  определяется следующим образом:

$$\bar{\omega}_{rj} = \begin{cases} d_{rj} & \text{для } 1 \leq r < IP(j), \\ 0 & \text{для } IP(j) < r \leq n, \quad j = 1, \dots, n, \\ DW(j) & \text{для } r = IP(j). \end{cases}$$

**Алгоритм НТР.**

**Шаг 1.** Положить:

$$\Phi = \sum_{j=1}^n c_{1(j)j}, \quad \bar{b} = b, \quad DW = IP = 0, \quad N = 0, \quad j = 1, \quad r = 1.$$

**Шаг 2.** Положить:

$$d = \begin{cases} c_{r+1j} - c_{rj}, & \text{если } r < m, \\ \infty, & \text{если } r = m, \end{cases}$$

$$\delta = \min_{1 \leq s \leq r} \{d, \bar{b}_{i(s)}\},$$

(5.23)

$$\Phi = \Phi + \delta, \quad \bar{b}_{i(s)} := \bar{b}_{i(s)} - \delta, \quad s = 1, \dots, r.$$

**Шаг 3.** Если  $\delta < d$ , положить  $DW(j) = \delta$ ,  $IP(j) = r$ ,  $N := N + 1$ , перейти к шагу 7.

**Шаг 4.** Положить  $j := j + 1$ .

**Шаг 5.** Если  $j > n$ , положить  $j = 1$ ,  $r := r + 1$ .

**Шаг 6.** Если  $IP(j) = 0$ , перейти к шагу 2, иначе — к шагу 4.

**Шаг 7.** Если  $N < n$ , перейти к шагу 4, иначе — СТОП.

Описанный алгоритм трудоемкости  $O(m^2n)$  строит допустимое базисное решение. В невырожденном случае этому решению можно поставить в соответствие единственную треугольную базисную матрицу  $B \subseteq A$ , которая строится по следующим правилам: множество ее строк соответствует  $\bar{b}_i = 0$ , а множество столбцов — переменным  $\omega_{rj}$ , для которых  $d_{rj} > \omega_{rj} > 0$  и номер  $r$  для соответствующих  $j$  записан в  $IP(j)$ . Обозначим  $I(\bar{W})$ ,  $RJ(\bar{W})$  множества базисных строк и столбцов.

Рассмотрим критерий оптимальности текущего решения  $\bar{W}$ , который следует из теории двойственности (см., например, [37]). Пусть

$Y$  определено следующим образом:  $\bar{Y}_i = 0$  ( $i \notin I(\bar{W})$ ), а остальные компоненты  $Y$  определяются из системы  $\bar{Y}_B B = 1$ , где  $\bar{Y}_B$  — вектор, компоненты которого соответствуют строкам матрицы  $B$  (из  $I(\bar{W})$ ).

**У т в е р ж д е н и е.** *Базисное решение  $\bar{W}$  оптимально, если  $\bar{Y}$  допустимо, т. е.*

- а)  $\bar{Y} \geq 0$ ;
- б)  $(a_{sj}, \bar{Y}) \geq 1 \quad \forall (s, j): \bar{\omega}_{sj} = 0$ ;
- в)  $(a_{sj}, \bar{Y}) \leq 1 \quad \forall (s, j): \bar{\omega}_{sj} = d_{sj}$ ; здесь  $a_{sj}$  — вектор-столбец  $(s, j)$  матрицы  $A$ .

Условия а)—б) утверждения обычным образом используются для построения оптимального решения задачи (5.18)—(5.20). Так, если для некоторой компоненты  $\bar{Y}_i < 0$ , то строка  $i$  выводится из базиса (для канонической формы симплекс-таблицы это соответствует введению на текущем шаге вспомогательной переменной  $z$ , коэффициент при которой в целевой функции равен нулю, а столбец коэффициентов матрицы условий содержит единицу только в строке  $i$ , остальные коэффициенты столбца равны нулю). Если  $\bar{Y} \geq 0$ , но нарушено условие б) (или в)), то соответствующий столбец вводится в базис. При этом значение соответствующей переменной  $\omega_{sj}$  возрастает (или убывает). Наряду с изменением значения вводимой в базис переменной, в соответствии с симплекс-преобразованием, изменяются и значения базисных переменных. В результате преобразования получается новое допустимое решение  $\tilde{W}$ , в котором либо вводимая в базис переменная достигает второй своей границы (если  $\bar{\omega}_{sj} = 0$ , то  $\tilde{\omega}_{sj} = d_{sj}$ , а если  $\bar{\omega}_{sj} = d_{sj}$ , то  $\tilde{\omega}_{sj} = 0$ ), либо одна из базисных переменных (например,  $\omega_{rk}$ ) достигает своей границы. Если имеет место первый случай, то для нового решения  $\tilde{W}$  базис не меняется. Во втором случае переменная  $\omega_{sj}$  вводится в базис, а  $\omega_{rk}$  выводится. Наконец, новое решение  $\tilde{W}$  может отличаться от  $\bar{W}$  тем, что некоторая строка  $i$ , для которой (5.19) выполнялось как неравенство в точке  $\bar{W}$ , обращается в равенство для  $\tilde{W}$ . В этом случае базисная матрица расширяется за счет введения этой строки и столбца  $(s, j)$ . Для проведения итерации симплекс-метода необходима информация лишь о части переменных  $\omega_{sj}$  и коэффициентах при них. Действительно, для этого нужны коэффициенты базисных столбцов (для вычисления вектора  $\bar{Y}$  и для преобразования вектора невязок  $\bar{b}_i$  в процессе итерации) и коэффициенты векторов условий, соответствующих переменным, на которых произошел «переход» значений в  $\bar{W}$  с верхней грани на нижнюю для каждого  $j$ , кроме базисных. Индексы таких переменных в алгоритме построения начального решения для каждого  $j$  определяются значением  $IP(j)$ .

Легко видеть, что если для некоторого  $(r, j)$  нарушено условие б) утверждения, то минимум по тем  $r$ , для которых  $\omega_{rj} = 0$ , величины



$\sum_{i=1}^m a_{irj} \bar{Y}_i$  при данном  $j$  достигается при  $r = s = IP(j)$ , и в базис следует ввести столбец  $a_{sj}$ . Аналогично при нарушении условия в)  $\arg \max_{r: \bar{w}_{rj} = d_{rj}} \sum_{i=1}^m a_{irj} \bar{Y}_i = t = IP(j) - 1$ , и в базис следует ввести  $a_{tj}$ .

Отсюда следует, что для проведения итерации симплекс-метода достаточно информации не более чем об  $n$   $m$ -мерных векторах из  $A$  (по одному для каждого  $j$ ), т. е. необходимая память оценивается величиной  $O(mn)$ , а объем вычислений —  $O((\min\{m, n\})^2)$  (предполагается, что используется и хранится матрица, обратная к текущей базисной, размер которой не превосходит  $p^2$ , где  $p = \min\{m, n\}$ ). За деталями описания техники симплекс-метода в общем случае мы отсылаем читателя, например, к [25], [37], [49]. Указанные выше оценки памяти и трудоемкости для задачи (5.18)—(5.20) на итерации симплекс-метода в  $mn$  раз меньше, чем на такой же итерации для задачи (5.6)—(5.8).

Рассмотрим теперь методы негладкой оптимизации при применении их к решению задачи (5.6)—(5.8). Декомпозиция последней может, вообще говоря, осуществляться двумя способами. В первом задача сводится к отысканию

$$\max_U \Psi(\bar{U}) = \max_U \min_Y \left( \sum_{i=1}^m \sum_{j=1}^n (c_{ij} - u_j) y_{ij} + \sum_{i=1}^m b_i Y_i + \sum_{j=1}^n u_j \right)$$

при ограничениях (5.8)—(5.9).

Внутренняя задача минимизации по  $Y$  при фиксированном  $\bar{U}$  легко решается. Для этого нужно вычислить значения

$$q_i = \sum_{j=1}^n \max\{0, u_j - c_{ij}\} - b_i, \quad i = 1, \dots, m,$$

а затем положить

$$\bar{Y}_i = \begin{cases} 1, & \text{если } q_i > 0, \\ 0, & \text{если } q_i \leq 0, \end{cases} \quad i = 1, \dots, m,$$

$$\bar{y}_{ij} = \begin{cases} 0, & \text{если } \bar{Y}_i = 0 \text{ или } c_{ij} - u_j \geq 0, \\ 1, & \text{если } \bar{Y}_i = 1 \text{ и } c_{ij} - u_j > 0, \end{cases} \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Подставляя значения  $\bar{Y}$  в ограничения (5.7) и вычисляя их невязки, мы определим таким образом вектор-субградиент для  $\Psi(U)$ , который необходим для использования методов негладкой оптимизации при отыскании  $\max \Psi(U)$ . Во втором случае используется динамическая декомпозиция и методы негладкой оптимизации применяются к задаче, двойственной к (5.18)—(5.20): найти

$$\min \left( \sum_{r=1}^m \sum_{j=1}^n d_{rj} z_{rj} + \sum_{i=1}^m b_i Y_i \right) \quad (5.24)$$

при ограничениях

$$\sum_{i=1}^m a_{irj} Y_i + z_{rj} \geq 1, \quad r = 1, \dots, m, \quad j = 1, \dots, n, \quad (5.25)$$

$$z_{rj} \geq 0, \quad r = 1, \dots, m, \quad j = 1, \dots, n, \quad (5.26)$$

$$Y_i \geq 0, \quad i = 1, \dots, m. \quad (5.27)$$

Здесь  $z_{rj}$  — двойственные переменные, соответствующие ограничениям (5.20). Из ограничений (5.25), (5.26) следует, что

$$z_{rj} = \max \left\{ 0, 1 - \sum_{i=1}^m a_{irj} Y_i \right\}. \quad (5.27)$$

Подставляя (5.27) в (5.24), получаем задачу относительно переменных  $Y_i$  при тривиальных ограничениях на неотрицательность: найти

$$\min_{Y_i \geq 0} \left( \sum_{r=1}^m \sum_{j=1}^n d_{rj} \max \left\{ 0, 1 - \sum_{i=1}^m a_{irj} Y_i \right\} + \sum b_i Y_i \right); \quad (5.28)$$

она является задачей минимизации выпуклой функции. Ограничения неотрицательности в задаче (5.28) учитываются обычным способом (см. гл. 2).

Рассмотрим теперь, как экономно организовать процедуру вычисления антисубградиента —  $\nabla F(\bar{Y})$  в точке  $\bar{Y}$ , используя частичную упорядоченность столбцов матрицы  $A$ . Положим сначала  $\nabla F(\bar{Y}) = \bar{b}$ . Затем для каждого  $j = 1, \dots, n$  уточним значения компонент вектора  $\nabla F(\bar{Y})$  по следующей процедуре, в которой  $i(r(j))$  обозначает индекс строки, имеющей порядковый номер  $r(j)$  в упорядочении  $j$ -го столбца матрицы  $C$ .

**Процедура уточнения  $\nabla F(\bar{Y})$  для текущего  $j$ .** 1. Найти минимальный номер  $s(j)$  такой, что

$$\sum_{r(j)=1}^{s(j)} \bar{Y}_{i(r(j))} \geq 1.$$

Если  $s(j) = 1$ , процедура закончена.

2. Для  $r = 1, \dots, s(j) - 1$  положить

$$\nabla_{i(r(j))} F(\bar{Y}) = \nabla_{i(r(j))} F(\bar{Y}) - (c_{s-1j} - c_{rj}).$$

Обоснование этой процедуры непосредственно следует из того, что при  $r(j) \geq s(j)$

$$\max \left\{ 0, 1 - \sum_{i=1}^m a_{irj} \bar{Y}_i \right\} = \max \left\{ 0, 1 - \sum_{r(j)=1}^{s(j)} \bar{Y}_{i(r(j))} \right\} = 0$$

и из формулы

$$\sum_{r=k}^s d_{rj} = c_{sj} - c_{kj},$$

поскольку именно с таким коэффициентом переменная  $Y_{i(k(j))}$  входит в  $j$ -ю сумму в (5.28).

В обоих рассмотренных случаях методы негладкой оптимизации дают решение задачи, эквивалентной (5.6)—(5.8). Отличия второго подхода от первого состоят в следующем. Размерность нелинейной задачи в первом случае равна  $n$ , а во втором —  $m$ . Кроме того, во втором случае процедура построения компоненты решения  $X^*$  по оптимальному решению  $Y^*$  более проста. Для этого достаточно положить

$$x_{i(r(j))}^* = \min \left\{ Y_{i(r(j))}^*, \max \left\{ 0, 1 - \sum_{i=1}^m a_{irj} Y_i^* \right\} \right\},$$

$$i = 1, \dots, m, \quad j = 1, \dots, n.$$

## § 2. Метод решения простейшей задачи размещения

Рассмотрим теперь более детально метод ветвей и границ для задачи (5.6)—(5.9), использующий описанные в предыдущем разделе способы решения релаксированной задачи (5.6)—(5.8). Ключевыми моментами методов такого типа являются:

- способ построения нижней оценки функционала;
- способ построения верхней оценки функционала и соответствующего ему допустимого решения и рекорда;
- способ разбиения задачи на подзадачи (ветвления) и порядок просмотра этих подзадач.

Способам построения нижней оценки функционала для рассматриваемой задачи посвящен предыдущий раздел. Рассмотрим, как по информации, полученной при построении нижней оценки, можно построить верхнюю оценку функционала (5.6) и допустимое по ограничениям (5.7)—(5.9) решение. Пусть  $W^*$  — оптимальное решение оценочной задачи (5.18)—(5.20). Обозначим через  $b$  ( $W^*$ ) вектор с компонентами

$$b_i(W^*) = b_i - \sum_{r=1}^m \sum_{j=1}^n a_{irj} w_{rj}^* \quad (i = 1, \dots, m).$$

Рассмотрим условия, которым должно удовлетворять оптимальное в задаче (5.6)—(5.8) значение вектора  $Y^*$  (они следуют из второй теоремы двойственности линейного программирования):

- а) если  $b_i(W^*) > 0$ , то

$$Y_i^* = 0 \quad (i = 1, \dots, m);$$

- б) если  $w_{rj}^* = d_{rj}$ , то

$$\sum_{i=1}^m a_{irj} Y_i^* \leq 1;$$

- если  $0 < w_{rj}^* < d_{rj}$ , то

$$\sum_{i=1}^m a_{irj} Y_i^* = 1,$$

если  $\omega_{rj}^* = 0$ , то

$$\sum_{i=1}^m a_{irj} Y_i^* \geq 1$$

$$r = 1, \dots, m, \quad j = 1, \dots, n.$$

Однако вектор  $Y^*$ , удовлетворяющий этим условиям, не обязан быть, вообще говоря, целочисленным. Для построения хорошего в эвристическом смысле допустимого решения, удовлетворяющего и условиям (5.9), можно требовать выполнения не всех условий б), а по возможности большей их части. Условия а) выполнить легко.

Обозначим  $m(j)$  множество индексов  $i$ , для которых при заданном  $j$  имеет место равенство

$$c_{ij} + \sum_{r=1}^m a_{irj} \omega_{rj}^* = \min_i \left( c_{ij} + \sum_{r=1}^m a_{irj} \omega_{rj}^* \right).$$

Тогда решение, по которому строится верхняя оценка функционала, можно выбрать с помощью следующей процедуры.

**Процедура построения приближенного решения (ППР).** 1. Положить  $Y_i^* = 0$  для таких  $i$ , которым соответствует  $b_i(W^*) > 0$ . Положить  $J = \{1, 2, \dots, n\}$ .

2. Найти  $j_0 \in J$ , для которого  $|m(j_0)| = \min_{j \in J} |m(j)|$ . Выбрать  $i_0 \in m(j_0)$ . Положить  $Y_{i_0}^* = 1$ ,  $J = J \setminus J(i_0)$ , где  $J(i_0) = \{j: i_0 \in m(j)\}$ . Если  $|m(j_0)| > 1$ , то  $i_0$  выбирается из условия

$$i_0 = \arg \max_{i \in m(j_0)} \sum_{j \in J(i)} B_j.$$

3. Если  $J = \emptyset$ , перейти к 4, иначе — к 2.

4. Для  $j = 1, \dots, n$  найти  $\min c_{ij} = c_{i(j)}$ , положить  $x_{i(j)}^* = 1$ ,  $x_{ij}^* =$

$= 0$  для  $i \neq i_0$ ,  $i = 1, \dots, m$ .

Полученному таким образом решению  $(X^*, Y^*)$  соответствует значение функционала  $F(X^*, Y^*) \geq \Phi(W^*)$ . Если  $F(X^*, Y^*) = \Phi(W^*)$ , то решение  $(X^*, Y^*)$  оптимально, в противном случае  $F(X^*, Y^*)$  является верхней оценкой для оптимального значения функционала (5.6).

Разбиение текущей задачи на подзадачи в методе ветвей и границ можно осуществлять различными способами. Простейший — это разбиение по одной из переменных  $Y_i$ . Оно осуществляется путем фиксации значения выбранной переменной равным нулю в одной подзадаче и единице — в другой. Возможны и другие варианты, например, фиксация по группе переменных  $Y_i$  ( $i \in M$ ), где  $M \subset \{1, 2, \dots, m\}$ . В последнем случае разбиение осуществляется, например, добавлением в одном случае условия  $\sum_{i \in M} Y_i = 0$ , или, что эквивалентно,  $Y_i = 0$  ( $i \in M$ ), а в другом —  $\sum_{i \in M} Y_i \geq 1$ . Отметим, что условия  $Y_i = 0$  экви-

валентны вычеркиванию соответствующих строк  $i$  из матрицы  $C$  либо замене  $b_i$  на достаточно большие числа, такие, что в оптимальном решении оценочной задачи величины  $b_i (W^*)$  положительны. Условие  $\sum_{i \in M} Y_i \geq 1$  для некоторого  $M$  эквивалентно добавлению к матрице  $C$  столбца  $n + 1$  с

$$c_{in+1} = \begin{cases} 0 & \text{при } i \in M, \\ \infty & \text{при } i \notin M. \end{cases}$$

Ясно, что оптимальное или приближенное решение расширенной таким образом задачи, построенное по вышеуказанным правилам, удовлетворяет требуемому условию.

Метод ветвей и границ заключается в построении верхних и нижних оценок для последовательности подзадач с разбиением каждой подзадачи, для которой верхняя и нижняя оценки функционала не совпадают, на новые подзадачи меньших размеров, и их отсевом. Существен способ просмотра (ветвления) порождаемых подзадач. Наиболее распространены способ параллельного ветвления, при котором для разбиения выбирается подзадача с наименьшей нижней оценкой, и способ одностороннего ветвления, при котором подзадачи просматриваются в некотором заранее определенном порядке. При этом в ходе просмотра подзадач запоминается наилучшее из просмотренных допустимых решений вместе со значением функционала на нем — рекордом. Порядок просмотра подзадач таков, что после анализа текущей задачи в случае, если нижняя оценка для нее меньше рекорда, просматривается ее потомок (т. е. одна из подзадач, полученных разбиением данной); иначе происходит возврат к анализу очередной непросмотренной подзадачи, порожденной предком текущей подзадачи, и так далее. Процесс заканчивается, когда все подзадачи просмотрены. При этом решение, соответствующее рекорду, является оптимальным. Односторонняя схема ветвления отличается от параллельной существенно меньшими требованиями к памяти для хранения информации о процессе перебора подзадач (дерева ветвлений). Требуемая память в этом случае оценивается величиной, пропорциональной числу переменных, по которым проводится ветвление, тогда как в первом случае эта величина может расти экспоненциально.

Рассмотрим теперь некоторые нестандартные аспекты метода ветвей и границ с односторонним ветвлением применительно к задаче (5.6)—(5.9).

Понятно, что чем точнее вычисляются оценки функционала, тем меньшее число подзадач будет проанализировано в процессе работы метода ветвей и границ. Однако общая трудоемкость метода в целом не всегда будет при этом минимальной. Она зависит также и от объема вычислений, необходимых для получения оценки. Если для построения нижней оценки функционала использовать алгоритм, который строит приближенное решение оценочной задачи с существенно меньшей трудоемкостью, а получаемое таким путем значение оценки не очень сильно отличается от оптимального, то общая трудоемкость решения задачи может оказаться меньшей. Например, если использовать для

построения приближенного решения задачи (5.18)—(5.20) алгоритм НТР с трудоемкостью  $O(m^2n)$ , то выигрыш по сравнению с симплекс-методом может быть очень значительным, поскольку теоретическая трудоемкость симплекс-метода в худшем случае растет экспоненциально с ростом размерности задачи. Интересно поэтому оценить погрешность решения, получаемого алгоритмом НТР, по сравнению с оптимальным решением. Обозначим  $W^0$  решение, получаемое с помощью алгоритма НТР,  $W^*$  — оптимальное решение задачи (5.18)—(5.20). Каждая задача вида (5.18)—(5.20) определяется набором параметров условий (5.19), (5.20)  $S = (m, n, A, b, d)$ . Мы хотим найти такой набор  $S$ , на котором величина  $\Phi_S(W^0)/\Phi_S(W^*)$  достигает минимума; здесь  $\Phi_S(W)$  — значение функционала (5.18) на решении  $W$  в задаче с набором  $S$ .

Обозначим через  $\tilde{S}$  требуемый экстремальный набор и рассмотрим его свойства. Представим  $W^0$  и  $W^*$  в виде  $W^0 = \bar{W} + W^-$ ,  $W^* = \bar{W} + W^+$ , где  $\bar{W} = (\bar{w}_{rj})$ ,  $\bar{w}_{rj} = \min\{\omega_{rj}^0, \omega_{rj}^*\}$  ( $r = 1, \dots, \tilde{m}$ ,  $j = 1, \dots, \tilde{n}$ ).

Лемма 5.1. Для набора  $\tilde{S}$   $\bar{W} = 0$ .

Доказательство. Если  $\bar{W} \neq 0$ , то  $\Phi_{\tilde{S}}(\bar{W}) > 0$  и

$$\frac{\Phi_{\tilde{S}}(W^0)}{\Phi_{\tilde{S}}(W^*)} = \frac{\Phi_{\tilde{S}}(W^-) + \Phi_{\tilde{S}}(\bar{W})}{\Phi_{\tilde{S}}(W^+) + \Phi_{\tilde{S}}(\bar{W})} > \frac{\Phi_{\tilde{S}}(W^-)}{\Phi_{\tilde{S}}(W^+)}.$$

Полагая  $\bar{m} = \tilde{m}$ ,  $\bar{n} = \tilde{n}$ ,  $\bar{A} = \tilde{A}$ ,  $\bar{d} = \tilde{d} - \bar{W}$ ,  $\bar{b} = \tilde{b} - \bar{A}\bar{W}$ , получим набор  $\tilde{S}$ , для которого роль  $W^0$  играет  $W^-$ ,  $W^* = W^+$ . Действительно, множество просмотренных алгоритмом НТР столбцов в матрице  $\bar{A} = \tilde{A}$  не меняется. С другой стороны, оптимальный базис, соответствующий  $W^*$ , остается оптимальным и для  $W^+$ , т. е.  $W^+$  оптимально. Но погрешность у  $W^-$  больше, чем у  $W^0$ , т. е.  $\tilde{S}$  при  $\bar{W} \neq 0$  не экстремально.

Для заданного набора  $S$  и допустимого решения  $W$  обозначим

$$I_S(W) = \{i: (AW)_i = b_i\}, \quad RJ_S(W) = \{(r, j): \omega_{rj} \geq 0\}, \\ A(W) = (a_{irj}), \quad i \in I_S(W), \quad (r, j) \in RJ_S(W).$$

Лемма 5.2. Существует экстремальный набор  $\tilde{S}$  и оптимальное решение  $W^*$  такие, что  $A(W^*) = E$ , где  $E$  — единичная матрица.

Доказательство. Подматрица  $A(W^*)$  должна содержать единицы в каждой своей строке. Если в одной из строк содержится больше одной единицы, например в столбцах  $(r, j)$  и  $(s, k)$ , то, полагая  $\bar{d}_{rj} = \bar{d}_{rj} + \bar{d}_{sk}$ ,  $\bar{d}_{sk} = 0$  и не изменяя остальных компонент  $\tilde{S}$ , получим новый набор  $\tilde{S}$ , для которого  $\Phi_{\tilde{S}}(W^*)$  равна  $\Phi_{\tilde{S}}(W^*)$ , и с мень-

шим числом единиц в  $A(W^*)$ . Корректность этого преобразования следует из теоремы 5.2. Действительно, в одном из экстремальных наборов каждый столбец  $(r, j)$  из  $A(W^*)$  содержит ровно одну единицу, поскольку в противном случае для столбца  $(r-1, j)$  выполнено условие  $W_{r-1j}^* = \tilde{a}_{r-1j}$ . Проводя для этой пары преобразование, аналогичное вышеприведенному, получим новое  $\tilde{S}$  и соответствующее  $W^*$  с меньшим числом единиц в  $A(W^*)$ . Повторим эту операцию до тех пор, пока  $A(W^*)$  не обратится в единичную матрицу. Для завершения доказательства отметим, что по лемме 5.1 эти преобразования не меняют  $W^0$ , соответствующего начальному набору  $\tilde{S}$ .

*Лемма 5.3. Существует пара  $(\tilde{S}, W^0)$  такая, что  $A(W^0) = I$ , где  $I$  — единичный вектор.*

Доказательство использует конструкцию, аналогичную описанной в доказательстве леммы 5.2.

*Лемма 5.4. Существуют  $\tilde{S}, W^*$  и  $W^0$  такие, что  $I_{\tilde{S}}(W^0) = I_{\tilde{S}}(W^*)$ .*

*Доказательство.* Если найдется  $i$  такое, что  $i \in I_{\tilde{S}}(W^0)$ ,  $i \notin I_{\tilde{S}}(W^*)$ , то положим  $\bar{n} = \tilde{n} + 1$  и добавим к матрице  $C$ , породившей  $\tilde{S}$ , новый столбец, подобранный таким образом, что новое  $W^*$  совпадает на  $RI_{\tilde{S}}(W^*)$  со старым, а ненулевая переменная, соответствующая в новом  $W^*$  добавленному столбцу, добавляет в  $I_{\tilde{S}}(W^*)$  строку  $i$ . При этом  $W^0$  не изменяется, и, следовательно,  $\tilde{S}$  не экстремально. Аналогично, если найдется  $i$  такое, что  $i \notin I_{\tilde{S}}(W^0)$ ,  $i \in I_{\tilde{S}}(W^*)$ , легко показать, что  $W^0$  и  $W^*$  порождают  $\bar{W} > 0$ , что противоречит лемме 5.1.

*Лемма 5.5. Существуют  $\tilde{S}, W^*$  и  $W^0$  такие, что  $\tilde{b}_i = \tilde{b}_j$  ( $i, j \in I_{\tilde{S}}(W^0)$ ).*

*Доказательство.* Если это не так, то для  $A(W^0) \cup \cup A(W^*)$ , удовлетворяющих леммам 5.2 и 5.3, не выполняется лемма 5.4.

Теперь нетрудно оценить максимальную погрешность  $W^0$  и построить оптимальный набор  $\tilde{S}$ . По леммам 5.1—5.5 экстремальная матрица  $\tilde{c}$  и вектор  $\tilde{b}$ , порождающие набор  $\tilde{S}$ , обладают следующими свойствами.

1. Множество  $I$  строк  $\tilde{C}$  разбивается на два подмножества  $I_{\tilde{S}}(W^*)$  и  $I \setminus I_{\tilde{S}}(W^*)$ ,  $|I_{\tilde{S}}(W^*)| = p$ . Будем считать, что  $I_{\tilde{S}}(W^*)$  содержит первые  $p$  строк  $\tilde{C}$ .

2. Вектор  $\tilde{b}$  также разбивается на две части, причем первые его  $p$  компонент по лемме 5.5 и за счет выбора масштаба можно считать равными единице.

3. Можно считать, что  $\tilde{C}$  содержит  $\tilde{n}$  столбцов,  $\tilde{n} = p + 1$ , которые, с учетом лемм 5.1 — 5.4, имеют следующие компоненты в первых  $p$  строках:

$$\begin{aligned} \tilde{c}_{i1} &= 0, & \tilde{c}_{ii+1} &= 1, & i &= 1, \dots, p, \\ \tilde{c}_{ij} &\geq 2, & i \neq j-1, & i &= 1, \dots, p, & j &= 2, \dots, p+1. \end{aligned}$$

4. Если положить не определенные еще коэффициенты матрицы  $\tilde{c}$  и  $\tilde{b}$  удовлетворяющими условиям  $\tilde{c}_{i1} \geq 1$  ( $i = p + 1, \dots, \tilde{m}$ ),  $\tilde{c}_{ij} = 0$  ( $i = p + 1, \dots, \tilde{m}$ ,  $j = 2, \dots, \tilde{n}$ ),  $\tilde{b} > 0$  ( $i = p + 1, \dots, \tilde{m}$ ), положить  $\tilde{m} = 2p - 1$  и решить полученную задачу алгоритмом НТР, то ее решение  $W^0$  будет иметь одну отличную от нуля компоненту  $w_{p(1)}^0 = 1$ . Оптимальное решение этой задачи  $W^*$  имеет  $p$  ненулевых компонент  $w_{p(j)}^* = 1$  ( $j = 2, \dots, \tilde{n}$ ). При этом  $\Phi_{\tilde{S}}(W^0)/\Phi_{\tilde{S}}(W^*) = 1/p = 1/[\tilde{m}/2]$ , где  $[a]$  — минимальное целое число, не меньшее, чем  $a$ . Легко понять, что определенное таким образом значение  $\tilde{m}$  является минимальным для данного  $p$ , поскольку в противном случае алгоритм НТР не построит решения  $W^0$  с указанными свойствами.

Таким образом мы построим экстремальный набор для заданного значения  $p$ . Значение  $\tilde{m}$  при этом — нечетное число. Для четных значений  $\tilde{m}$  искомая матрица легко модифицируется, и приведенные рассуждения позволяют показать, что экстремальная погрешность  $W^0$  при  $\tilde{m} = 2p$  такая же, как в рассмотренном случае. Итак, доказана следующая теорема.

**Т е о р е м а 5.3.** Для произвольного  $m$  и  $n \geq [m/2] + 1$

$$\min_S \frac{\Phi_S(W^0)}{\Phi_S(W^*)} = \frac{1}{[m/2]}.$$

Гарантированная относительная точность алгоритма НТР в худшем случае с ростом  $m$  может стать сколь угодно малой величиной, однако задачи, на которых достигается минимум точности, имеют с точки зрения практических приложений ярко выраженный патологический вид. В реальных задачах размещения обычно выполняются метрические ограничения на элементы матрицы  $C$  (аналоги условий треугольника для расстояний). Легко проверить, что в рассмотренной задаче эти условия существенно нарушены. Из приведенных рассуждений можно сделать вывод, что в реальных ситуациях подобные патологические задачи должны встречаться редко. Они также редки, если, например, элементы матрицы  $C$  — случайные равномерно распределенные на некотором отрезке числа. Поэтому на рассматриваемых классах матриц  $C$  величина относительной погрешности должна быть существенно ниже. Это предположение согласуется с экспериментальными результатами, приведенными ниже, которые получены путем решения



тестовых задач отмеченных выше классов с помощью программы, реализующей метод НТР и некоторой его модификации.

Рассмотрим теперь некоторые особенности схемы организации процесса генерации и решения подзадач, возникающих в методе ветвей и границ. Обычно метод ветвей и границ предполагает решение оценочной задачи в каждой из вершин дерева ветвлений. Однако можно получить существенную экономию объема вычислений, если производить решение оценочных задач лишь в части таких вершин. Предположим, что мы решили начальную оценочную задачу, т. е. задачу, соответствующую самой верхней вершине дерева ветвлений в методе ветвей и границ с односторонним обходом дерева ветвлений. Пусть  $I^0(1)$  — множество единичных компонент в приближенном решении, построенном по оптимальному или приближенному решению оценочной задачи (5.18)—(5.20). Построим фрагмент дерева ветвлений, соответствующий полученному решению. Для этого упорядочим индексы  $I^0(1)$  по убыванию величины некоторого критерия, смысл которого — вероятность вхождения соответствующей переменной с единичным значением в оптимальное решение исходной задачи. В качестве такого критерия может быть выбрана величина  $\sum_{j=1}^n B_j x_{ij}^0 / \sum_{j=1}^n B_j$  для  $i \in I^0(1)$ . Искомый

фрагмент дерева ветвлений представим теперь в виде цепи, начальная вершина которой соответствует исходной задаче, а каждая последующая вершина  $r$  соответствует подзадаче, которая получена из предыдущей путем фиксации значения  $Y_{i(r)} = 1$  ( $r = 1, \dots, I^0(1)$ ), где  $i(r)$  — индекс переменной с порядковым номером  $r$ . Таким образом, последней вершине цепи соответствует задача, в которой все переменные из  $I^0(1)$  зафиксированы как единичные.

Покажем, что расширение этой цепи путем фиксации  $Y_i = 1$  для любого  $i \notin I^0(1)$  бесперспективно, так как при этом нельзя построить решение со значением целевой функции меньшим, чем верхняя оценка функционала, соответствующая начальной вершине цепи. Обозначим через  $k$  вершину дерева ветвлений в методе ветвей и границ, в которой решена оценочная задача, через  $I^k(1)$  — множество индексов свободных (нефиксированных в схеме ветвления) переменных, равных единице в решении, соответствующем верхней границе, построенной в этой вершине, через  $Y^k$  — множество единичных компонент в этом решении. Скажем, что произвольное допустимое решение с множеством единичных компонент  $I(1)$  нерасширяемо, если добавление к  $I(1)$  любого подмножества единичных компонент не приводит к решению с меньшим значением функционала.

**Теорема 5.4.** *Решение задачи (5.6)—(5.9), соответствующее множеству  $I^k(1)$ , нерасширяемо.*

**Доказательство.** Пусть  $W^k$  — значение вектора  $W$  на типовом решении, соответствующем нижней оценке в вершине  $k$ ,  $I^k(1)$  — множество единичных компонент в приближенном решении, построенном по  $W^k$ . Предположим, что на решении с множеством единичных компонент  $I^k(1) \cup \tilde{I}(1)$  значение функционала (5.6) меньше,

чем на рассматриваемом приближенном. Тогда для каждого  $l \in \tilde{I}(1)$  множество столбцов  $j \in J(l)$ , для которых в расширенном решении  $x_{lj} = 1$  должно удовлетворять следующим условиям:

$$\sum_{j \in J(l)} (c_{i^k(j)} - c_{lj}) > b_l, \quad c_{lj} < c_{i^k(j)}, \quad j \in J(l),$$

где  $i^k(j)$  — строка, на которой достигается  $\min c_{lj}$  для данного  $j$ .

Действительно, в противном случае исключение  $l$  из  $\tilde{I}(1)$  по крайней мере не увеличит функционал (5.6). Но легко видеть, что

$$b_l \geq \sum_{s=r(i^k(j))}^{r(m)} \omega_{sj}^k \geq \sum_{s=r(i^k(j))}^{r(l)-1} \omega_{sj}^k \geq \sum_{j \in J(l)} (c_{i^k(j)} - c_{lj}).$$

Здесь первое неравенство следует из допустимости решения  $W^k$  по ограничению  $l$  в (5.19), второе — из теоремы 5.2, а третье — из способа выбора индекса  $i^k(j)$  и тупиковости решения  $W^k$ . Получено противоречие, доказывающее справедливость утверждения теоремы.

Таким образом, из теоремы 5.4 следует, что дальнейшее ветвление задачи, соответствующей вершине, полученной из  $k$  добавлением цепи с фиксированными единичными компонентами вектора  $Y^k$ , бесперспективно. Отсюда следует, что либо  $Y^k$  порождает оптимальное решение задачи (5.6)—(5.9), либо в оптимальном решении хотя бы одна из компонент  $Y^k$  равна нулю. Поэтому не имеет смысла решать оценочные задачи для вершин цепи, порожденных вектором  $Y^k$ . После анализа вершины  $k$  можно сразу перейти к анализу вершины  $k+l$ , которая соответствует подзадаче, полученной из  $k$  путем фиксации всех переменных из  $I^k(1)$  равными единице, кроме последней (по введенному выше порядку), которая фиксируется равной нулю. Пример дерева с таким способом ветвления показан на рис. 5.1. Вершины, в которых проводится решение оценочной задачи, изображены темными кружками, остальные — светлыми кружками.

Легко понять, что число вершин, в которых проводится решение оценочной задачи, отличается на единицу от числа вершин другого типа, поскольку для каждой вершины порожденные подзадачи принадлежат к разным типам. Поэтому предложенный способ уменьшает число решаемых оценочных задач примерно вдвое.

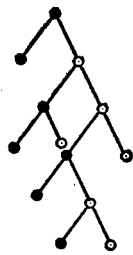


Рис. 5.1

Кратко опишем теперь схему реализованных в виде фортран-программ для ЕС ЭВМ алгоритмов решения задачи (5.6)—(5.9) и результатов численных экспериментов. Разработаны две модификации программы, которые отличаются способом построения нижней оценки. В первом варианте нижняя оценка строится с помощью процедуры НТР, во втором — полученная нижняя оценка улучшается с помощью процедуры типа замены. Суть этой процедуры состоит в следующем. Пусть  $\bar{A}$  — подматрица  $A$ , столбцы которой соответствуют тем  $k(j)$  ( $j = 1, \dots, n$ ), для которых  $\omega_{kj}^0 > 0$  и  $k(j)$  — максимальный индекс для

любого  $j$  с таким свойством, а строки  $i \in I_0$ , где  $I_0 = \{i: \bar{b}_i = 0\}$ , где  $\bar{b}_i$  получены после работы процедуры НТР. Процедура улучшения находится, если это возможно, максимальное подмножество столбцов  $\{a_1, \dots, a_l\}$  из  $\bar{A}$  мощности  $l \geq 2$  и столбец  $\bar{a}_0 \in \bar{A}$  такие, что

$$\sum_{t=1}^l a_t \leq \bar{a}_0.$$

Если такие столбцы найдены, то текущие значения  $w_i^0$ , соответствующие столбцам  $a_t$ , увеличиваются на некоторую величину  $\delta$ , а  $w_0^0$ , соответствующая столбцу  $\bar{a}_0$ , уменьшается на  $\delta$ . Величина  $\delta$  выбирается из условия

$$\delta = \min \left\{ w_0^0, \min_{1 \leq t \leq l} (d_t - w_t^0), \min_{i \in I_0} b_i / \left( \sum_{t=1}^l a_{it} - a_{i_0} \right) \right\}.$$

Ясно, что построенное таким образом новое решение  $w^0$  остается допустимым, а значение целевой функции увеличивается на величину  $\delta (l - 1)$ . Если при этом  $\delta = w_0^0$ , то в столбце  $\bar{a}_0$  находится единичная компонента с максимальным порядковым номером  $k$  ( $j$ ) и заменяется на 0. Если  $d = d_t - w_t^0$ , то в столбце  $t$  находится нулевая компонента с минимальным порядковым номером  $k$  ( $j$ ) и заменяется на единицу. Здесь  $j$  — индекс столбца матрицы  $C$ , который определяет рассматриваемый столбец подматрицы  $A$ . Наконец, если

$$\delta = b_i / \left( \sum_{t=1}^l a_{it} - a_{i_0} \right)$$

(в этом случае неравенство (5.19) для  $i_0$  обращается в равенство при  $W = \bar{W}$ ), то полагаем  $I_0 = I_0 \cup i$ . При переходе от  $W^0$  к  $\bar{W}$  пересчитываются значения  $\bar{b}_i$  для  $i \notin I_0$ , а также для тех  $i \in I_0$ , для которых  $\sum_{t=1}^l a_{it} < a_{i_0}$ . Такие  $i$  исключаются из  $I_0$ . После выполнения всех указанных преобразований процедура повторяется с новым вектором  $W^0 = \bar{W}$ .

Экспериментальное исследование методов решения задачи (5.6) — (5.9) проводилось на задачах со случайно сгенерированной матрицей  $C$ . Результаты представлены в таблице 5.1. Для порождения последовательности псевдослучайных равномерно распределенных на  $(0, 1)$  чисел  $\sigma_k$  использовалась подпрограмма, вычисляющая  $\sigma_k$  по правилам:  $\sigma_k := \{p_k \cdot p_{k+1}/4\}$ ;  $p_k := p_{k+1}$ ;  $p_{k+1} := \sigma_k$  ( $k = 1, 2, \dots$ ), где  $\{\cdot\}$  — дробная часть,  $p_1 = 3,1141593$ ,  $p_2 = 0,5421018$ .

В табл. 5.1 использованы следующие обозначения:  $m$  — число точек размещения,  $n$  — число потребителей,  $b$  — величина фиксированных доплат,  $\Phi$ ,  $F$ ,  $F^*$  — начальные нижняя и верхняя оценки, ре-

Таблица 5.1

№	$m \times n$	$b$	$\Phi$	$F$	$F^*$	$\delta$	NR	L	Q	$t$
1 а	50×50	500	5921	7850	6643	8,3		358	8	97,31
1 б	»	500	6007	7915	6643	9,6	33	207	9	354,72
2 а	»	1000	8464	11504	9358	9		294	12	274,34
2 б	»	1000	8505	11286	9358	9,1	4	117	3	290,98
3 а	»	3000	14335	19708	16163	10,7		90	4	47,24
3 б	»	3000	14503	22010	16163	10,3	17	51	4	210,78
4 а	»	5000	19050	30320	21195	9,3		102	9	100,32
4 б	»	5000	19189	30874	21195	9,5	29	55	6	272,28
5 а	»	7000	23004	340073	25195	3,2		64	5	47
5 б	»	7000	23223	32968	25195	7,8	11	33	4	190,82
6 а	»	50000	1745442	1813914	1752258	0,38	9	11	5	21,08
6 б	»	50000	1747003	1813914	1752258	0,29	9	15	5	15,02
7 а	»	70000	1881333	1988656	1894750	0,78	22	43	5	29,01
7 б	»	70000	1888843	1959119	1894750	0,31	8	27	3	32,68
8 а	»	90000	2011131	2113727	2021439	0,51	9	37	5	29,32
8 б	»	90000	2011649	2166881	2021439	0,48	22	27	4	44,78
9 а	»	100000	2063990	2243572	2081439	0,083	10	33	4	26,16
9 б	»	100000	2064446	2155679	2081439	0,082	8	33	4	43,44
10 а	»	150000	2313940	2427549	2331894	0,076	26	39	5	33,22
10 б	»	150000	2317624	2427549	2331894	0,064	22	27	4	44,78
11 а	100×100	100000	3569822	3858974	3597693	0,77	37	153	7	657,62
11 б	»	100000	3569822	3859974	3597693	0,77	35	129	6	912,40
12 а	»	200000	4207095	4605447	4247907	0,96	106	387	7	2139,34
12 б	»	200000	4209886	4405447	4247907	0,89	202	241	8	2199,88
13 а	»	300000	4716489	5402477	4736736	0,43	53	61	9	419,28
13 б	»	300000	4726489	5402477	4736736	0,4	15	29	5	351,76
14 а	»	400000	5127390	5421742	5136736	0,18	4	7	2	71,92
14 б	»	400000	5127814	5421742	5136736	0,17	4	7	2	102,96
15 а	»	500000	5497304	5990876	5536736	0,71	2	15	2	155,5
15 б	»	500000	5503251	5990876	5536736	0,6	2	15	2	257,64

кордное значение функционала,  $\delta = \frac{F^* - \Phi}{F^*} \cdot 100\%$ , NR — номер итерации, на которой построено рекордное решение, L — число итераций, Q — число изменений рекорда в методе ветвей и границ,  $t$  — время счета (на шаге GO) в секундах. Незаполненные позиции в данной и последующей таблицах означают, что соответствующие параметры не фиксировались. Примеры, помеченные индексом а, решены без использования процедуры улучшения нижней оценки, индексом б — с использованием этой процедуры. Программа, реализующая описанный метод ветвей и границ, позволяет строить приближенное решение с априорно заданным отклонением от оптимума. Во всех примерах это отклонение задавалось равным 0,01. В задачах 1—5 коэффициенты матрицы C задавались как псевдослучайные равномерно распределенные целые числа из отрезка [0,1000], в остальных задачах — как евклидовы расстояния между псевдослучайными точками (потребления и размещения), равномерно распределенными на целочисленной решетке в квадрате со стороной 1000. Фиксированные доплаты задавались равными для всех  $i$  и изменялись с целью проследить, как при этом изменяется трудоемкость задачи. Задачи 1а, 3а,

4а, 5а решались на ЕС-1060, остальные — на ЕС-1040. Из анализа приведенных данных можно сделать вывод, что задачи с матрицами  $C$ , обладающими метрическими свойствами (многие прикладные задачи относятся именно к такому классу), требуют для решения меньших временных затрат по сравнению со случайно сгенерированными матрицами. Можно также отметить, что хотя процедура улучшения оценки уменьшает, как правило, число итераций метода ветвей и границ, общее время решения задачи может как увеличиваться, так и уменьшаться. Вероятно, эта процедура будет более эффективной на задачах большей размерности.

Кроме тестовых задач решался ряд задач с реальными исходными данными выбора пунктов размещения переработки винограда [51]. Расчеты проводились с помощью алгол-программы на ЭВМ БЭСМ-6. Матрица  $C$  в этих задачах строилась по реальным затратам на перевозку продукции. Поскольку расстояния перевозки продукции от поставщиков к местам переработки в данной задаче существенно влияют на качество продукции, задача решалась при различных ограничениях на максимальное расстояние перевозки. Эти ограничения учитывались следующим образом. Если для пары  $(i, j)$  расстояние  $r_{ij}$  превышало заданную величину  $R$ , то соответствующее  $c_{ij}$  полагалось бесконечным. Для каждого пункта размещения  $i$  функция  $f_i(X_i)$  опре-

Таблица 5.2

№	$m$	$n$	$R$	$\delta$	$\Delta$	$P$	$L$	$Q$	$t$
1	95	120	40	0,5	3,2	37	790	37	7947
2	23	34	30			12	41	4	30
	—	—	40			10	83	5	49
	—	—	50			8	410	14	192
	—	—	60			7	14	4	24
	—	—	$\infty$			1	7	7	15
3	23	34	30	0	0,4	12	1	1	14
	—	—	40	0,4	2,5	8	49	6	39
	—	—	50	0,4	7,4	8	188	14	134
	—	—	60	0,5	17,6	8	290	50	223
4	28	28	30	0,9	2,2	11	9	4	18
	—	—	40	0,4	8,5	7	284	11	133
	—	—	50	0,7	6,0	6	91	6	65
	—	—	60	0,9	8,6	6	26	5	30
5	28	28	20	0	0,1	17	1	1	13
	—	—	30	0	0,1	14	1	1	13
	—	—	40	1,4	4,0	8	1639	21	615
	—	—	50	0	6,3	8	1090	106	522
6	23	27	20	0,8	0,8	13	22	2	24
	—	—	30	0	0,4	12	1	1	12
	—	—	40	0,4	1,7	8	45	6	31
	—	—	50	0	8,0	8	242	19	135
	—	—	60	0,9	17,6	8	292	49	185
7	14	17	20	0	0,2	7	1	1	8
	—	—	30	0	6,1	6	34	5	14
	—	—	40	0	0,3	4	1	1	8
	—	—	50	0	0	3	1	1	8

делялась как минимум из двух линейных функций с фиксированными доплатами. Реализованная на БЭСМ-6 программа позволяет работать непосредственно с кусочно линейными функциями без расширения множества точек размещения за счет перехода к линейным функциям с фиксированными доплатами. Отметим, что в данной программе реализована односторонняя схема ветвления с начальной фиксацией значения переменной ветвления равным нулю. Данные о решении указанных задач приведены в табл. 5.2. Здесь  $n$  — число поставщиков,  $m$  — число пунктов размещения,  $R$  — максимально допустимое расстояние (в км),  $\delta$  ( $\Delta$ ) — относительное отклонение начальной нижней (верхней) оценки от оптимального значения функционала в %,  $P$  — число пунктов размещения, выбранных в оптимальном решении,  $L$  — число итераций,  $Q$  — число улучшений рекорда в методе ветвей и границ,  $t$  — время счета (в секундах). Все задачи решены точно ( $T = 0$ ).

### § 3. Некоторые обобщения простейшей задачи размещения

В данном разделе рассмотрены некоторые классы задач размещения производства, которые в различных направлениях обобщают простейшую задачу размещения. В связи с большим разнообразием таких обобщений рассмотрение всех моделей практически невозможно. Поэтому мы ограничились выбором лишь таких характерных представителей, по анализу и решению которых у авторов имеется некоторый опыт. Большинство рассматриваемых моделей обладает многими свойствами, аналогичными свойствам простейшей задачи. В их число входят, например, вогнутость целевой функции, ее структура (сумма транспортных и производственных затрат), «транспортность» ее ограничений, применимость без принципиальных усложнений метода динамической декомпозиции и др. Поэтому при изложении последующего материала мы останавливаемся лишь на существенных отличиях данных задач от предшествующих, не повторяя общих мест.

**1. Многоэтапные задачи размещения.** Рассматриваемый класс задач отличается наличием нескольких уровней производства продукции, через которые проходит сырье, прежде чем превратится в готовую продукцию и поступит к потребителю. Так, металл до его использования должен пройти два уровня: добычу руды и его выплавку.

Рассмотрим простейшую модель размещения производства с двухэтапной обработкой продукции: найти

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n C_{ij} x_{ij} + \sum_{j=1}^n \sum_{k=1}^p D_{jk} y_{jk} + \sum_{i=1}^m f_i(X_i) + \sum_{j=1}^n g_j(Y_j) \right)$$

при ограничениях

$$\sum_{j=1}^n x_{ij} = X_i, \quad i = 1, \dots, m;$$

$$\sum_{i=1}^m x_{ij} = \sum_{k=1}^p y_{jk} = Y_j, \quad j = 1, \dots, n;$$

$$\sum_{j=1}^n y_{jh} = B_k, \quad k = 1, \dots, p;$$

$$x_{ij} \geq 0, \quad y_{jh} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad k = 1, \dots, p.$$

Здесь  $i, j, k$  — индексы мест размещения первичной и вторичной обработки потребителей;  $C = (C_{ij}), D = (D_{jh})$  — матрицы удельных затрат на транспортировку продукции между пунктами первого и второго этапа обработки и между пунктами второго этапа и потребителями;  $B = (B_k)$  — вектор объемов потребления;  $f_i(X_i), g_j(Y_j)$  — вогнутые кусочно линейные функции затрат на производство в пунктах размещения  $i, j$  первого и второго этапов соответственно.

Так же, как и простейшая задача размещения, данная сводится к следующей задаче частично целочисленного линейного программирования: найти

$$\min \left( \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^p c_{ijk} z_{ijk} + \sum_{i=1}^M a_i X_i + \sum_{j=1}^N b_j Y_j \right) \quad (5.29)$$

при ограничениях

$$\sum_{i=1}^M \sum_{j=1}^N z_{ijk} = 1, \quad k = 1, \dots, p, \quad (5.30)$$

$$\sum_{j=1}^N z_{ijk} \leq X_i, \quad \sum_{i=1}^M z_{ijk} \leq Y_j, \quad i = 1, \dots, M,$$

$$j = 1, \dots, N, \quad k = 1, \dots, p, \quad (5.31)$$

$$z_{ijk} \geq 0, \quad i = 1, \dots, M, \quad j = 1, \dots, N, \quad k = 1, \dots, p, \quad (5.32)$$

$$X_i = 0 \vee 1, \quad Y_j = 0 \vee 1, \quad i = 1, \dots, M, \quad j = 1, \dots, N. \quad (5.33)$$

Здесь  $M, N$  — мощность новых множеств значений индексов  $i, j$ , полученных путем замены каждой кусочно линейной функции  $f_i(X_i)$  и  $g_j(Y_j)$  множеством функций с фиксированными доплатами, где

$$f_i(X_i) = \min_{s \in S(i)} (a_{is} + l_{is} X_i), \quad a_{is} \geq 0, \quad l_{is} \geq 0, \quad i = 1, \dots, m,$$

$$g_j(Y_j) = \min_{t \in T(j)} (b_{jt} + h_{jt} Y_j), \quad b_{jt} \geq 0, \quad h_{jt} \geq 0, \quad j = 1, \dots, n.$$

Для упрощения записи в преобразованной задаче вместо пар индексов  $(i, s)$  и  $(j, t)$  используются  $i$  и  $j$  соответственно. Величины  $c_{ijk}$  для всех значений  $i, j, k$  в (5.29) определяются подобно тому, как определены значения  $c_{ij}$  в (5.6).

Ограничения (5.31) непосредственно следуют из того факта, что минимум вогнутой функции в исходной задаче достигается в вершине

многогранника. Поэтому в преобразованной задаче каждая из сумм принимает лишь следующие значения:

$$\sum_{j=1}^N z_{ijk} = 0 \vee 1, \quad i = 1, \dots, M, \quad k = 1, \dots, p,$$

$$\sum_{i=1}^M z_{ijk} = 0 \vee 1, \quad j = 1, \dots, N, \quad k = 1, \dots, p.$$

Задача, двойственная в (5.29) — (5.32), заключается в отыскании

$$\max_{\substack{U \geq 0 \\ V \geq 0}} \min_Z ((c_{ijk} + u_{ik} + v_{jk}) z_{ijk}) \quad (5.34)$$

при ограничениях (5.30), (5.32) и

$$\sum_{k=1}^p u_{ik} \leq a_i, \quad i = 1, \dots, M, \quad (5.35)$$

$$\sum_{k=1}^p v_{jk} \leq b_j, \quad j = 1, \dots, N, \quad (5.36)$$

где  $U, V$  — векторы двойственных переменных, соответствующих ограничениям (5.31). Отметим, что ограничения (5.31) отличаются по структуре от ограничений (4.16) в задаче (4.14)—(4.17), поэтому форма целевой функции (5.34) не соответствует форме (4.18). Но (5.34) можно привести к указанному виду путем введения дополнительных переменных и ограничений. Однако мы предпочитаем не делать этих преобразований, а анализировать задачу в форме (5.29)—(5.32) и (5.34)—(5.36) в двойственной постановке.

Использование принципа динамической декомпозиции применительно к этим задачам не имеет существенных отличий от рассмотренных ранее случаев. Для использования указанного метода необходимо определить правила построения блокирующих множеств для текущего решения  $Z(U^t, V^t)$ .

Пусть  $U^t, V^t$  — произвольные допустимые по ограничениям (5.35), (5.36) значения векторов  $U, V$ . Обозначим

$$T(k) = \{(i, j) : (i, j) = \arg \min_{r,s} (c_{rsk} + u_{rk}^t + v_{sk}^t)\}, \quad k = 1, \dots, p.$$

**Утверждение.** Минимальными блокирующими множествами, соответствующими  $(U^t, V^t)$ , являются минимальные по включению наборы индексов  $(I^t, J^t)$  такие, что для каждого  $i \in I^t$  и каждого  $j \in J^t$  найдется пара  $(i, j) \in T(k)$ , и только они.

Иными словами, минимальные блокирующие множества содержат ровно один элемент из каждой пары индексов  $(i, j)$  из множества  $T(k)$ .

В отличие от простейшей задачи размещения, в данной задаче каждому текущему вектору  $(U^t, V^t)$  и каждому потребителю  $k$  соответ-



ствует не менее двух различных минимальных блокирующих множеств. Одно из них образуют, например, все индексы  $i$  из пар  $(i, j) \in T(k)$ , а другое — все  $j$  из  $(i, j) \in T(k)$ . Кроме того, если  $|T(k)| > 1$ , блокирующие множества могут быть смешанными, т. е. содержать как индексы  $i$ , так и  $j$  из  $T(k)$ . По этой причине в данной задаче нельзя полностью упорядочить блокирующие множества для данного  $k$ , а можно ввести лишь частичный порядок, который зависит от последовательности выбираемых блокирующих множеств при построении последовательности векторов  $\{(U^t, V^t)\}$ . Для решения задачи (5.34) — (5.36) можно использовать как точные методы, обобщающие соответствующие методы решения двойственной к простейшей задаче размещения, так и приближенные. Мы рассмотрим здесь лишь метод построения приближенного тупикового решения, построенный на идеях, использованных в процедуре НТР для простейшей задачи размещения и реализованный для построения нижней оценки функционала для задачи (5.29) — (5.33) в фортран-программе, описываемой ниже.

Пусть  $\bar{a} = (\bar{a}_i)$ ,  $\bar{b} = (\bar{b}_j)$  — векторы нераспределенных остатков значений компонент векторов  $a$  и  $b$  из (5.35), (5.36) при работе описываемой процедуры. Для данной пары векторов текущих значений  $(U^t, V^t)$  и данного  $k$  назовем блокирующее множество  $\beta_k(U^t, V^t)$  очередным, если оно удовлетворяет следующим условиям:

- а) если  $i_0 \in \beta_k(U^t, V^t)$ , то  $\bar{a}_{i_0} > 0$ ;
- б) если  $j_0 \in \beta_k(U^t, V^t)$ , то  $\bar{a}_i = 0$  для всякой пары  $(i, j_0) \in T(k)$  и  $\bar{b}_{j_0} > 0$ .

Процедура построения тупикового решения для задачи (5.34) — (5.36) просматривает в циклическом порядке индекс  $k$  и для каждого его значения строит по  $T(k)$ ,  $U^t$ ,  $V^t$ ,  $\bar{a}$ ,  $\bar{b}$  очередное блокирующее множество. Если его не существует, соответствующее значение  $k$  исключается из дальнейшего рассмотрения. Если же  $\beta_k(U^t, V^t)$  построено, значения  $u_i^t$ ,  $v_j^t$  для  $i, j \in \beta_k(U^t, V^t)$  увеличиваются на  $\delta$ , где  $\delta$  выбирается так, чтобы для нового значения компонент  $U^t$ ,  $V^t$  либо множество  $T(k)$  расширилось (появился новый кратчайший путь  $(i, j, k)$  в вершину  $k$ ), либо одна из компонент  $\bar{a}_i$ ,  $\bar{b}_j$ , где  $i, j \in \beta_k(U^t, V^t)$ , при уменьшении на  $\delta$  обратилась в нуль. Затем корректируются: текущее значение целевой функции (5.34) (увеличивается на  $\delta$ ), значения  $\bar{a}$ ,  $\bar{b}$  (уменьшаются на  $\delta$  компоненты с индексами из  $\beta_k(U^t, V^t)$ ), множество  $T(k)$  (добавляется при необходимости новая пара  $(i, j)$ ). Процесс повторяется до тех пор, пока все  $k$  не будут исключены из рассмотрения.

Построенное таким образом тупиковое решение  $(\bar{U}, \bar{V})$  позволяет по правилам, аналогичным рассмотренным при анализе простейшей задачи размещения, построить допустимое решение задачи (5.29) — (5.33) и, таким образом, получить верхнюю оценку функционала (5.29). Это решение также обладает свойством нерасширяемости (см. теорему 5.4). Описанные свойства двухэтапной задачи без труда переносятся и на многоэтапные задачи размещения.

Для двухэтапной задачи размещения реализован в виде фортран-программы для ЕС ЭВМ и алгол-программы для БЭСМ-6 метод ветвей и границ, основанный на тех же принципах, что и соответствующий метод решения простейшей задачи размещения. В этих программах процедуры построения нижних и верхних границ, выбора переменной и схемы ветвления принципиально аналогичны вышеуказанным. С помощью алгол-программы была решена задача с реальными данными [51]. Размерность решенной задачи —  $M \times N \times p = 25 \times 95 \times 120$ , время решения — более шести часов. Функции затрат на размещение предприятий на обоих уровнях задавались линейными с фиксированными доплатами.

Было проведено решение серии тестовых задач на ЕС-1040. Результаты приведены в табл. 5.3. В задачах 1—8 коэффициенты матриц  $C$  и  $D$  задавались как случайные целые числа, равномерно распределенные на  $[0, 100]$ , в задачах 9—17 — как евклидовы расстояния между точками размещения производства обоих уровней и потребителей, равномерно распределенными в квадрате со стороной 500. Величины  $a_i, b_j$  из (5.29) задавались одинаковыми для всех  $i, j$  и равными  $a, b$ . Величины  $B_k$  задавались равными: в примерах 1—8 —  $(s+0,1) \times 10$ ; в примере 11 — 50; в примерах 12, 14 —  $(s+0,1) \times 100$ ; в остальных —  $(s+0,1) \times 1000$ , где  $s$  — псевдослучайное равномерно распределенное на  $(0, 1)$  число. В табл. 5.3 использованы обозначения, введенные в табл. 5.2.

**2. Простейшая задача размещения и унификации технических средств [70].** Данная задача возникает при выборе типов, количества и мест размещения технических средств, необходимых для выполнения заданного множества работ. Примером такой задачи является задача выбора типов и мест базирования вертолетов, предназначенных для выполнения ими в различных пунктах заданного района комплекса

Таблица 5.3

№	$M$	$N$	$p$	$a$	$b$	$T$	$\delta$	$\Delta$	$L$	$Q$	$t$
1	20	20	20	300	300	0	9,2	20,7	29	5	1'43"
2	20	20	20	500	500	0	9,5	42,0	25	7	1'51"
3	40	40	40	300	300	0	12,8	39,2	43	5	27'35"
4	40	40	40	500	500	0	16,7	53,2	35	6	29'23"
5	40	40	40	500	500	0	18,5	12,3	109	3	102'20"
6	40	40	40	1000	1000	0	11,7	75,7	37	5	41'06"
7	40	40	40	1000	1000	0	22,2	36,4	139	4	83'01"
8	40	40	40	1500	1500	0,05	8,6	71,2	23	5	28'31"
9	15	30	50	30000	15000	0	0	9,0	59	6	3'12"
10	15	30	50	300000	100000	0	0,3	24,4	19	6	4'33"
11	15	30	50	350000	125000	0,03	1,5	11,3	107	8	18'09"
12	15	30	50	700000	25000	0,03	2,0	14,3	89	8	15'40"
13	15	30	50	700000	50000	0	0,2	8,5	185	18	17'26"
14	15	30	50	700000	50000	0,3	0	32,8	77	9	15'37"
15	15	30	50	700000	100000	0	0,6	16,3	83	11	8'52"
16	15	30	50	3500000	100000	0	0	91,6	5	3	1'35"

работ различных видов. В этом случае разные виды работ могут быть выполнены разными типами средств с различными производительностями и затратами, которые зависят также от мест базирования вертолетов и от мест выполнения работ. Кроме суммарных затрат на выполнение работ, следует учесть затраты на производство и закупку технических средств, а также затраты на их размещение и техническое обслуживание.

Введем следующие обозначения:

$c_{ijk}$ ,  $p_{ijk}$  — удельные затраты и удельная производительность при использовании технических средств типа  $i$ , размещенных в пункте  $j$  для выполнения работ вида  $k$  ( $i = 1, \dots, m$ ,  $j = 1, \dots, n$ ,  $k = 1, \dots, p$ );

$A_k$  — объем работ вида  $k$  (в приведенных единицах) ( $k = 1, \dots, p$ );

$\lambda_i$  — коэффициент приведения к условным единицам технических средств типа  $i$  ( $i = 1, \dots, m$ );

$x_{ijk}$  — объем технических средств типа  $i$ , размещенных в пункте  $j$ , направленных для выполнения работ вида  $k$  ( $i = 1, \dots, m$ ,  $j = 1, \dots, n$ ,  $k = 1, \dots, p$ );

$X_i$ ,  $Y_j$  — объем технических средств типа  $i$  и условных единиц всех типов технических средств в пункте  $j$  ( $i = 1, \dots, m$ ,  $j = 1, \dots, n$ );

$f_i(X_i)$ ,  $g_j(Y_j)$  — вогнутые кусочно линейные функции затрат на производство и закупку  $X_i$  единиц типа  $i$  ( $i = 1, \dots, m$ ) и на размещение и обслуживание  $Y_j$  условных единиц в пункте  $j$  ( $j = 1, \dots, n$ ).

В этих обозначениях рассматриваемая задача состоит в отыскании

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p c_{ijk} x_{ijk} + \sum_{i=1}^m f_i(X_i) + \sum_{j=1}^n g_j(Y_j) \right)$$

при ограничениях

$$\sum_{j=1}^n \sum_{k=1}^p x_{ijk} = X_i, \quad i = 1, \dots, m,$$

$$\sum_{i=1}^m \sum_{k=1}^p \lambda_i x_{ijk} = Y_j, \quad j = 1, \dots, n,$$

$$\sum_{i=1}^m \sum_{j=1}^n p_{ijk} x_{ijk} = A_k, \quad k = 1, \dots, p,$$

$$x_{ijk} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad k = 1, \dots, p.$$

Из вогнутости целевой функции непосредственно следует, что в оптимальном решении каждая переменная  $x_{ijk}$  равна либо нулю, либо  $A_k/p_{ijk}$ . Используя этот факт, представим задачу в форме частично целочисленной задачи линейного программирования: найти

$$\min \left( \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^p d_{ijk} z_{ijk} + \sum_{i=1}^M a_i X_i + \sum_{j=1}^N b_j Y_j \right)$$

при ограничениях

$$\sum_{i=1}^M \sum_{j=1}^N z_{ijk} = 1, \quad k = 1, \dots, p,$$

$$\sum_{j=1}^N z_{ijk} \leq X_i, \quad i = 1, \dots, M, \quad k = 1, \dots, p,$$

$$\sum_{i=1}^M z_{ijk} \leq Y_j, \quad j = 1, \dots, N, \quad k = 1, \dots, p,$$

$$z_{ijk} \geq 0, \quad X_i = 0 \vee 1, \quad Y_j = 0 \vee 1, \quad i = 1, \dots, M,$$

$$j = 1, \dots, N, \quad k = 1, \dots, p;$$

здесь

$$d_{ijk} = \frac{c_{ijk} + l_i + \lambda_i h_j}{p_{ijk}},$$

а  $l_i, h_j, a_i, b_j, M, N, p$  соответствуют аналогичным величинам для двухэтапной задачи размещения, приведенной выше.

Легко усмотреть, что рассматриваемая задача отличается от двухэтапной задачи размещения только более общим видом коэффициентов  $d_{ijk}$ , т. е. предшествующая задача является частным случаем рассматриваемой. В табл. 5.4 приведены данные по решению тестовых примеров рассматриваемой задачи с помощью фортран-программы, реализующей метод, аналогичный описанному для двухэтапной задачи размещения.

Таблица 5.4

№	M	N	p	a	b	T	Δ	δ	L	Q	t
1	8	5	4	20	20	0,03	10,9	8,4	29	3	8
2	10	7	5	20	20	0,03	23,1	17,4	413	6	111
3	10	7	5	50	50	0,03	29,8	15,1	295	3	122
4	10	8	5	40	40	0,03	28,3	10,5	519	4	380
5	15	8	5	60	60	0,03	26,4	23,8	573	5	331
6	20	10	7	60	60	0,03	46,2	37,4	673	6	932

**3. Задача размещения технических средств с сезонным выполнением работ.** Эта задача возникает при использовании технических средств для выполнения работ в различные временные периоды. Мы рассмотрим лишь упрощенный вариант задачи в линейной постановке: найти

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T c_{ijt} x_{ijt} + \sum_{j=1}^n d_j X_j \right) \quad (5.37)$$

при ограничениях

$$\sum_{j=1}^n p_{ijt} x_{ijt} = A_{it}, \quad i = 1, \dots, m, \quad t = 1, \dots, T, \quad (5.38)$$

$$\sum_{i=1}^m x_{ijt} \leq X_j, \quad j = 1, \dots, N, \quad t = 1, \dots, T, \quad (5.39)$$

$$x_{ijt} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad t = 1, \dots, T. \quad (5.40)$$

Здесь  $c_{ijt}$ ,  $p_{ijt}$  — удельные затраты и производительность при выполнении работы вида  $i$  техническим средством типа  $j$  в период  $t$ ;

$A_{it}$  — объем работ вида  $i$  в период  $t$ ;

$x_{ijt}$  — количество средств типа  $j$ , назначенных на работу вида  $i$  в период  $t$ ;

$X_j$  — количество средств типа  $j$ ;

$d_j$  — удельные затраты на закупку и эксплуатацию средств типа  $j$ .

Двойственная к данной задаче может быть сформулирована в виде: найти

$$\max_U \min_X \sum_{i=1}^m \sum_{j=1}^n \sum_{t=1}^T (c_{ijt} + u_{jt}) x_{ijt}$$

при ограничениях (5.38), (5.40) и

$$\sum_{t=1}^T u_{jt} \leq d_j, \quad j = 1, \dots, n,$$

$$u_{jt} \geq 0, \quad j = 1, \dots, n, \quad t = 1, \dots, T.$$

При использовании метода динамической декомпозиции для решения сформулированной задачи блокирующие множества, порожденные допустимым вектором  $\bar{U}$ , определяются по следующим правилам. Для каждой пары  $(i, t)$  блокирующими являются наборы  $\{(j, t), j \in J(i, t)\}$  такие, что

$$c_{ijt} + \bar{u}_{jt} = \min_{1 \leq k \leq n} (c_{ikt} + \bar{u}_{kt}).$$

Отметим, что при точном решении задачи (5.37)–(5.40) методами динамической декомпозиции возникает необходимость решения на каждой итерации  $T$  задач распределительного типа (при фиксированном значении вектора  $\bar{X} = (\bar{X}_j)$  необходимо определить значения компонент  $\{x_{ijt}\}$ ).

Для решения задачи типа (5.37)–(5.40), в которой условия (5.39) заменены на

$$\sum_{i=1}^m x_{ijt} \leq B_j + X_j, \quad j = 1, \dots, n, \quad t = 1, \dots, T,$$

где  $B_j$  — количество средств типа  $j$ , имеющееся к началу планирования, была разработана алгол-программа для ЭВМ БЭСМ-6, реализующая метод обобщенного градиентного спуска. С ее помощью решена задача распределения самолетов сельхозавиации для выполнения сезонных работ. Размеры решенной задачи —  $m = 112$ ,  $n = 84$ ,  $T = 12$ . Время решения — свыше 20 часов. Кроме указанной, разработана PL/1-программа для ЕС ЭВМ, реализующая метод декомпозиции по переменным  $X_j$  с использованием для минимизации по внешним переменным  $r$  алгоритма (см. гл. 2). Для решения на каждой итерации вспомогательной внутренней задачи при фиксированных значениях вектора  $(X_j)$  использовалась программа, реализующая метод потенциалов для распределительной задачи, разработанная Б. Р. Френкиным (ЦЭМИ АН СССР).

**4. Выбор оптимального комплекса оборудования.** Требуется спроектировать предприятие, выпускающее заданный ассортимент продукции. Для каждого вида продукции  $i$  ( $i = 1, \dots, m$ ) задан набор из  $p_i$  возможных технологических маршрутов  $k = 1, \dots, p_i$ , каждый из которых использует оборудование различных типов  $j$  ( $j \in A_{ik}$ ),  $A_{ik} \subset \{1, 2, \dots, n\}$  — множество используемых типов оборудования. Для каждой пары  $(i, k)$  известны производственные затраты  $c_{ik}$ , а для каждого типа оборудования  $j$  — функция  $f_j(X_j)$  (кусочно линейная, вогнутая, неубывающая) затрат на размещение и закупку оборудования от объема  $X_j$  выполняемых на нем работ. Мы рассмотрим лишь простейшую задачу рассматриваемого типа, предполагая, что все

$$f_j(X_j) = \begin{cases} a_j & \text{при } X_j > 0, \\ 0 & \text{при } X_j = 0. \end{cases}$$

В случае более общих  $f_j(X_j)$  задача усложняется, но сохраняет при этом основные используемые свойства. Введем переменные  $x_{ik}$ , полагая  $x_{ik} = 1$ , если в оптимальном решении для производства продукции  $i$  выбран вариант  $k$ , и  $x_{ik} = 0$  — в противном случае ( $i = 1, \dots, m$ ,  $k = 1, \dots, p_i$ ). Далее, пусть  $z_j = 1$ , если тип оборудования  $j$  выбирается в оптимальном решении, и  $z_j = 0$  — в противном случае ( $j = 1, \dots, n$ ). Во введенных обозначениях требуется найти

$$\min \left( \sum_{i=1}^m \sum_{k=1}^{p_i} c_{ik} x_{ik} + \sum_{j=1}^n a_j z_j \right) \quad (5.41)$$

при ограничениях

$$\sum_{k=1}^{p_i} x_{ik} = 1, \quad i = 1, \dots, m, \quad (5.42)$$

$$\sum_{k=1}^{p_i} \Delta_{ijk} x_{ik} \leq z_j, \quad i = 1, \dots, m, \quad j = 1, \dots, n,$$

$$x_{ik} \geq 0, \quad i = 1, \dots, m, \quad k = 1, \dots, p_i, \quad (5.43)$$

$$z_j = 0 \vee 1, \quad j = 1, \dots, n; \quad (5.44)$$

здесь

$$\Delta_{ijk} = \begin{cases} 1, & \text{если } j \in A_{ik}, \quad i = 1, \dots, m, \quad k = 1, \dots, p_i, \quad j = 1, \dots, n, \\ 0 & \text{в противном случае.} \end{cases}$$

Вводя двойственные переменные  $u_{ij} \geq 0$ , запишем задачу, двойственную к релаксированной в форме: найти

$$\max_{U \geq 0} \min_{X \geq 0} \sum_{i=1}^m \sum_{k=1}^{p_i} \left( c_{ik} + \sum_{j=1}^n \Delta_{ijk} u_{ij} \right) x_{ik}$$

при ограничениях (5.42) и

$$\sum_{i=1}^m u_{ij} \leq a_j, \quad j = 1, \dots, n.$$

Обозначим

$$K_i(\bar{U}) = \left\{ k : c_{ik} + \sum_{j=1}^n \Delta_{ijk} \bar{u}_{ij} = \min_{1 \leq l \leq p_i} \left( c_{il} + \sum_{j=1}^n \Delta_{ijl} \bar{u}_{ij} \right) \right\}, \quad i = 1, \dots, m.$$

Блокирующие множества для данных  $i$ ,  $\bar{U}$  определяются множеством индексов  $j$  таких, что единичные коэффициенты матрицы  $\Delta = \{\Delta_{ijk}\}$  покрывают все столбцы  $k \in K_i(\bar{U})$ .

Для решения задачи (5.41)–(5.44) разработана фортран-программа для ЕС ЭВМ, построенная на идеях, близких к описанным в §§ 1, 2, и проведен ряд экспериментальных расчетов по решению тестовых задач.

#### § 4. Варианты простейшей задачи размещения

**1. Задача размещения производства однородной продукции с ограничениями на мощность.** Рассмотрим следующую задачу размещения производства в целочисленной постановке: найти

$$F(X^*) = \min \left( \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m b_j X_i \right) \quad (5.45)$$

при ограничениях

$$\sum_{i=1}^m x_{ij} = B_j, \quad j = 1, \dots, n, \quad (5.46)$$

$$\sum_{j=1}^n x_{ij} \leq A_i X_i, \quad i = 1, \dots, m, \quad (5.47)$$

$$0 \leq x_{ij} \leq M_{ij} X_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (5.48)$$

$$\sum_{i \in I_k} X_i \leq 1, \quad k = 1, \dots, p, \quad (5.49)$$

$$X_i = 0 \vee 1, \quad i = 1, \dots, m. \quad (5.50)$$

Здесь  $A_i$  — ограничение сверху на мощность предприятия, размещаемого по варианту  $i$ ,  $i = 1, \dots, m$ ,  $M_{ij} = \min \{A_i, B_j\}$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ ,  $I_k$ ,  $k = 1, \dots, p$ , — непересекающиеся подмножества индексов  $i$ . Остальные обозначения совпадают с аналогичными для простейшей задачи размещения. В отличие от простейшей данная модель позволяет учитывать ограничения сверху на мощности размещаемых предприятий, а также может рассматриваться в случае, когда мощности размещаемого производства задаются набором возможных вариантов. В последнем случае  $I_k$  — множество вариантов для заданного места размещения производства  $k$  ( $k = 1, \dots, p$ ).

Для задачи (5.45)—(5.50) рассмотрим особенности ее решения методом множителей Лагранжа в дискретном программировании (см. [130, 131]). Для этого сформулируем прежде всего задачу, двойственную к (5.45)—(5.50): найти

$$\Phi(U^*) = \max_U \min_X \left( \sum_{i=1}^m \sum_{j=1}^n (c_{ij} + u_j) x_{ij} + \sum_{i=1}^m b_i X_i + \sum_{j=1}^n B_j u_j \right) \quad (5.51)$$

при ограничениях (5.47) — (5.50). Как известно, в общем случае  $F(X^*) \geq \Phi(U^*)$  и равенство не всегда достижимо из-за разрыва двойственности в дискретных задачах. Поэтому значение  $\Phi(U^*)$  является лишь нижней оценкой для  $F(X^*)$ . Задача (5.47)—(5.51) является задачей безусловной максимизации вогнутой кусочно линейной функцией относительно переменных  $U$ . Для ее решения могут быть использованы методы негладкой оптимизации. На каждой итерации таких методов необходимо вычислять  $\Phi(U)$  и вектор-субградиент  $\nabla\Phi(\bar{U})$  в заданной точке  $\bar{U}$ . Для этого нужно решить задачу (5.47)—(5.51) при  $U = \bar{U}$ . Легко видеть, что последняя распадается на  $p$  независимых подзадач с номерами  $k = 1, \dots, p$ , каждая из которых содержит лишь переменные  $x_{ij}$  и  $X_i$ ,  $i \in I_k$ ,  $j = 1, \dots, n$ .

Каждая подзадача  $k$ ,  $k = 1, \dots, p$ , может быть переписана в виде: найти

$$\min_{i \in I_k} \min_X \left( \sum_{j=1}^n \bar{c}_{ij} x_{ij} + b_i X_i \right) \quad (5.52)$$

при ограничениях

$$\begin{aligned} \sum_{j=1}^n x_{ij} &\leq A_i, \quad i \in I_k, \\ 0 &\leq x_{ij} \leq M_{ij} X_i, \quad j = 1, \dots, n, \quad i \in I_k, \\ X_i &= 0 \vee 1, \quad i \in I_k. \end{aligned} \quad (5.53)$$

Здесь  $\bar{c}_{ij} = c_{ij} + \bar{u}_j$ ,  $i \in I_k$ ,  $j = 1, \dots, n$ ,  $\bar{u}_j$  —  $j$ -я компонента точки  $\bar{U}$ . Для решения задачи (5.52), (5.53) применим простой алгоритм, который вычисляет сначала значение  $i$ -й компоненты целевой функции (5.52), а затем из этих компонент выбирает минимальную, полагая  $X_i = 1$  только для последней. Значение  $i$ -й компоненты



вычисляется с помощью градиентного алгоритма, просматривающего переменные  $x_{ij}$  в соответствии с упорядочением величин  $\bar{c}_{ij}$  по неубыванию. При этом значение текущей переменной  $X_i$  равно единице.

Покажем, что решение задачи (5.52), (5.53) дает значение функционала, совпадающее со значением функционала соответствующей задачи с непрерывными переменными  $X_i$ . Иными словами, оптимальное решение задачи (5.52), (5.53) совпадает с одним из оптимальных решений ее линейной релаксации, которая формулируется в виде: найти

$$\min \left( \sum_{i \in I_k} \sum_{j=1}^n \bar{c}_{ij} \bar{x}_{ij} + \sum_{i \in I_k} b_i X_i \right)$$

при ограничениях

$$\begin{aligned} \sum_{j=1}^n x_{ij} &\leq A_i X_i, & i \in I_k, & \quad \sum_{i \in I_k} X_i \leq 1, \\ 0 \leq x_{ij} &\leq M_{ij} X_i, & i \in I_k, & \quad j = 1, \dots, n. \end{aligned}$$

Рассмотрим некоторые свойства этой задачи. Обозначим ее оптимальное решение  $X^*$ .

*Лемма 5.6. Если  $\bar{c}_{ij} > 0$  для некоторой пары  $(i, j)$ , то  $x_{ij}^* = 0$ .*

Доказательство тривиально. С учетом этого утверждения все  $\bar{c}_{ij} > 0$  можно заменить на  $\bar{c}_{ij} = 0$ . При этом множество оптимальных решений не изменится. Для дальнейшего анализа удобно целевую функцию переписать в форме

$$\sum_{i \in I_k} \sum_{j \in J_i} \tilde{c}_{ij} x_{ij} - \sum_{i \in I_k} b_i X_i,$$

где  $\tilde{c}_{ij} = -\bar{c}_{ij}$ ,  $i \in I_k$ ,  $j \in J_i$ ,  $J_i = \{j: \bar{c}_{ij} < 0\}$ ,  $i \in I_k$ .

*Лемма 5.7. Условия неотрицательности на переменные  $x_{ij}$ ,  $i \in I_k$ ,  $j \in J_i$  избыточны.*

Доказательство очевидно.

Обозначим  $n_i = |J_i|$ ,  $i \in I_k$ . Пусть коэффициенты  $\tilde{c}_{ij}$  упорядочены по невозрастанию, т. е.  $\tilde{c}_{i1} \geq \tilde{c}_{i2} \geq \dots$

*Лемма 5.8. Для каждого  $i \in I_k$  имеет место следующая альтернатива: либо*

$$x_{ij}^* = \begin{cases} M_{ij} X_i^* & \text{для } 1 \leq j \leq j_0, \\ A_i X_i^* - \sum_{j=1}^{j_0} x_{ij}^* & \text{для } j = j_0 + 1 \leq n_i, \\ 0 & \text{для } j > j_0 + 1, \end{cases}$$

*и при этом*

$$\sum_{j \in J_i} x_{ij}^* = A_i X_i^*,$$

либо

$$x_{ij}^* = \begin{cases} M_{ij} X_i^* & \text{для } 1 \leq j \leq n_i, \\ 0 & \text{для } j > n_i, \end{cases}$$

и при этом

$$\sum_{i \in J_i} x_{ij}^* \leq A_i X_i^*.$$

Доказательство непосредственно следует из применения к рассматриваемой задаче с фиксированными  $X_i^*$  градиентного алгоритма для каждого  $i \in I_k$ .

Следствие. Если  $\sum_{j=1}^{n_i} M_{ij} > A_i$ , то в оптимальном решении соответствующее данному  $i$  неравенство выполняется как строгое равенство. Если  $\sum_{j=1}^{n_i} M_{ij} \leq A_i$ , то соответствующее неравенство избыточно, и его можно отбросить.

Обозначим через  $\bar{I}_k$  множество индексов  $i$ , для которых вышеуказанные неравенства обращаются в равенство. С учетом леммы 5.8 и ее следствия исследуемую задачу перепишем в виде: найти

$$\min \left( \sum_{i \in I_k - \bar{I}_k} \bar{c}_i X_i + \sum_{i \in \bar{I}_k} d_i z_i \right)$$

при ограничениях

$$\sum_{i \in I_k} X_i \leq 1, \quad 0 \leq z_i \leq \bar{A}_i X_i, \quad i \in \bar{I}_k,$$

где  $\bar{c}_i = \sum_{j=1}^{i_0} \tilde{c}_{ij}$ ,  $d_i = \bar{c}_{i(j_0+1)}$ ,  $\bar{A}_i = A_i - \sum_{j=1}^{i_0+1} M_{ij}$ ,  $z_i = X_{i(j_0+1)}$ ,  $i \in \bar{I}_k$ .

Наконец, исключая переменные  $z_i$ , получим задачу относительно лишь неотрицательных переменных  $X_i$ , сумма которых не превосходит единицы. Ясно, что все ее базисные решения целочисленны

Пусть  $\bar{U}$  — текущее решение задачи (5.51),  $X(\bar{U}) = (\{x_{ij}(\bar{U})\}, \{X_i(\bar{U})\})$  — оптимальное решение внутренней задачи. Компоненты вектора  $\nabla \Phi(\bar{U})$  в точке  $\bar{U}$  вычисляются по формуле  $\nabla \Phi_j(\bar{U}) = B_j - \sum_{i=1}^m x_{ij}(\bar{U})$  ( $j = 1, \dots, n$ ). Таким образом, у нас имеются все данные

для вычисления оптимального решения задачи (5.51).

Указанный способ вычисления нижних оценок с помощью  $r$ -алгоритма реализован в фортран-программе решения задачи (5.45) — (5.50). Структура ее близка к остальным частям (построение верхней оценки, выбор переменной ветвления, способ ветвления) к программе решения простейшей задачи размещения. Поэтому мы не останавливаемся на изложении соответствующего материала. По указанной

программе проведен небольшой вычислительный эксперимент на ЕС-1022. Данные для решенных задач генерировались с помощью датчика псевдослучайных чисел, равномерно распределенных на некотором отрезке. Элементы матрицы  $C$  задавались на отрезке  $[0, 100]$ , величина  $B_j$  — на  $[20, 20 + 5m]$  ( $j = 1, \dots, n$ ),  $A_i$  — на  $[40, 40 + 10n]$  ( $i = 1, \dots, m$ ). При таком способе задания  $\sum_{j=1}^n B_j \approx 20n + 2,5 mn$ ,

$\sum_{i=1}^m A_i \approx 40m + 5mn$ . Для близких друг к другу значений  $m, n$  сум-

марная потребность примерно равна половине суммарной мощности мест производства. Значения доплат  $b_i$  принимались одинаковыми:  $b_i = v \cdot 500m$  ( $i = 1, \dots, m$ ), где  $v$  — параметр эксперимента. Результаты приведены в табл. 5.5. Здесь  $m$  — число пунктов размещения,  $n$  — число потребителей,  $q$  — число вариантов в каждом пункте размещения,  $v$  — параметр в формуле вычисления  $b_i$ ,  $t$  — время в секундах.

Таблица 5.5

№	$m$	$n$	$q$	$\lambda$	$t$	№	$m$	$n$	$q$	$\lambda$	$t$
1	10	20	2	0	152	7	20	20	1	0,5	832
2	10	20	2	0,5	542	8	20	20	1	1	142
3	10	20	2	1	1247	9	35	10	1	0	771
4	10	20	2	0,5	120	10	35	10	1	0,5	114
5	10	20	2	1	100	11	35	10	1	1	1247
6	20	20	1	0	1069						

Для решения задачи (5.45)—(5.50) можно применить и метод динамической декомпозиции. Задача, двойственная к (5.45)—(5.49), записывается в форме: найти

$$\max_{U, V, W} \min_{\lambda} \left( \sum_{i=1}^m \sum_{j=1}^n (c_{ij} + v_i + w_{ij}) x_{ij} - \sum_{k=1}^p u_k \right) \quad (5.54)$$

при ограничениях (5.46) и

$$A_i v_i + \sum_{j=1}^n M_{ij} w_{ij} \leq b_i + u_k, \quad i = 1, \dots, m, \quad k = 1, \dots, p, \quad (5.55)$$

$$u_k \geq 0, \quad v_i \geq 0, \quad w_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad k = 1, \dots, p. \quad (5.56)$$

В отличие от простейшей задачи размещения в рассматриваемой задаче присутствуют переменные  $v_i$  и  $u_k$ . Блокирующие множества здесь имеют более сложную структуру. Для заданного набора значений двойственных переменных  $\bar{V}, \bar{W}$  они определяются следующим образом.

Пусть для каждого  $j$

$$I(j) = \{ \arg \min_i (c_{ij} + \bar{v}_i + \bar{w}_{ij}) \}, \quad j = 1, \dots, n.$$

**Лемма 5.9.** *Блокирующим множеством для данного  $\bar{V}, \bar{W}$  является множество пар  $(i, j)$ ,  $j \in J(\bar{V}, \bar{W})$ ,  $i \in I(j)$ , элементы которого помечены индексами  $V$  и  $W$  в соответствии со следующим правилом: если  $\sum M_{ij} > A_i$ , то элементы  $(i, j)$ , соответствующие слагаемым в левой части неравенства, помечены индексом  $V$ , в противном случае — индексом  $W$ ; здесь суммирование для данного  $i$  ведется только по тем элементам из  $J(\bar{V}, \bar{W})$ , которые принадлежат множествам  $I(j)$ .*

**Доказательство.** Понятно, что для того, чтобы увеличить значение функционала при данном  $\bar{V}, \bar{W}$ , необходимо увеличить значения переменных ( $v_i$  или  $w_{ij}$ ), соответствующих парам  $(i, j)$  из  $I(j)$  для одного или более столбцов  $j$  матрицы  $C = \{c_{ij}\}$ . Увеличение это за счет увеличения переменных  $w_{ij}$  производится так же, как и в простейшей задаче размещения, т. е. по любому из множеств  $I(j)$ . Увеличение же за счет  $v_i$  оказывается лучшим, чем за счет соответствующих  $w_{ij}$  (с точки зрения величины оставшегося для дальнейшего распределения значения  $\bar{b}_i$ ), тогда и только тогда, когда выполняется условие утверждения. Поэтому наилучшим с указанной точки зрения способом увеличения значения функционала является увеличение значений переменных  $v_i$  в тех строках  $i$ , для которых указанное условие выполнено; остальные компоненты увеличиваются за счет  $w_{ij}$ .

Переменные  $u_k$  играют роль параметров, за счет которых можно (если это позволяет увеличить значение функционала) увеличить правые части в ограничениях (5.55). Необходимость увеличения некоторых переменных  $u_k$  определяется, как обычно, по решению прямой задачи, соответствующему заданному набору значений переменных  $(\bar{U}, \bar{V}, \bar{W})$ . Если этому набору соответствует (при выполнении условий первой и второй теорем двойственности) решение  $\bar{X}$ , в котором для всех  $k$  условия (5.49) не нарушены, то  $(\bar{U}, \bar{V}, \bar{W})$  и  $\bar{X}$  — оптимальные решения двойственной и прямой задач. Если же условие (5.49) нарушено при некотором  $k$ , следует увеличить значение  $u_k$  (и некоторые компоненты  $V, W$ ) на величину  $\delta$ , которая выбирается так, чтобы в строках матрицы  $C$ , соответствующих данному  $k$ , появился хотя бы один новый элемент в одном из множеств  $I(j)$ . После этого указанная процедура определения нового  $\bar{X}$  повторяется. Таким образом, алгоритм симплексного типа решения задачи (5.54)—(5.56) будет состоять из итераций двух типов: внутренних, на каждой из которых при фиксированном (на первой итерации — нулевом) векторе  $\bar{U}$  делается шаг симплекс-метода в декомпозированной задаче, переменные которой соответствуют блокирующим множествам, и внешних, на которых по указанной выше схеме изменяются значения переменных  $\bar{U}$  и согласованно с ними — значения  $\bar{V}, \bar{W}$ .

Какой из двух предложенных здесь способов построения нижней оценки в задаче (5.45)—(5.50) окажется эффективнее в схеме ветвей и границ, сможет показать, вероятно, только вычислительный эксперимент.

**2. Простейшая задача размещения по смешанному критерию.** На практике часто возникают задачи, в которых требуется минимизировать затраты на размещение обслуживающих пунктов плюс сумму расстояний от каждой выбранной точки размещения до самой дальней из точек, прикрепленных к данной. Примеры такого рода возникают при размещении ретрансляционных станций, пожарных депо, пунктов скорой помощи и т. д. Ниже рассматривается простейшая модель таких задач и ряд ее свойств, позволяющих использовать для их решения развиваемый здесь подход. Задача рассматриваемого типа формулируется в виде: найти

$$\min \sum_{i=1}^m (\max_{1 \leq j \leq n} c_{ij} x_{ij} + b_i X_i) \quad (5.57)$$

при ограничениях

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, n, \quad (5.58)$$

$$0 \leq x_{ij} \leq X_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (5.59)$$

$$X_i = 0 \vee 1, \quad i = 1, \dots, m. \quad (5.60)$$

Здесь  $b_i$  — стоимость размещения обслуживающего пункта в точке  $i$  ( $i = 1, \dots, m$ ),  $C = (c_{ij})$  — матрица расстояний для пар  $(i, j)$ , где  $i = 1, \dots, m, j = 1, \dots, n$  — индексы обслуживаемого пункта. Остальные обозначения и ограничения аналогичны введенным для простейшей задачи размещения.

Обычно в задачах рассматриваемого типа радиус сферы, внутри которой лежат пункты, обслуживаемые данным центром, ограничен. Это связано с техническими особенностями задачи — ограничением расстояния передаваемого качественного сигнала в задачах размещения ретрансляторов, максимально допустимым временем прибытия пожарной команды к месту пожара и т. д. Указанные ограничения учитываются путем задания соответствующих коэффициентов матрицы  $C$ . Если  $c_{ij} > r_i$ , то  $c_{ij}$  заменяется на очень большое число (здесь  $r_i$  — критическое расстояние для  $i$ -го пункта размещения). Критерий (5.57) можно рассматривать как линейную комбинацию двух критериев; первый минимизирует сумму максимальных расстояний — радиусов обслуживания выбранных пунктов, — второй — суммарные затраты на размещение обслуживающих пунктов. Поскольку эти критерии, вообще говоря, несравнимы между собой, в задачу можно ввести параметр  $\lambda$ , заменив каждое  $b_i$  на  $\lambda b_i$ , и решить последовательность таких задач при различных значениях  $\lambda$ . При этом выбор наиболее подходящего значения производится из внемоделльных (дополнительных, не отраженных в данной модели) соображений.

Перейдем к анализу свойств задачи (5.57)—(5.59). Прежде всего перепишем целевую функцию (5.57) в виде

$$\sum_{i=1}^m (t_i + b_i X_i), \quad (5.61)$$

добавив к ограничениям задачи следующие:

$$c_{ij} x_{ij} \leq t_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (5.62)$$

Легко понять, что преобразованная таким образом задача эквивалентна исходной. Сформулируем задачу, двойственную к (5.57)—(5.59): найти

$$\max_{\substack{U \geq 0 \\ V \geq 0}} \min_{X \geq 0} \sum_{i=1}^m \sum_{j=1}^n (u_{ij} + c_{ij} v_{ij}) x_{ij} \quad (5.63)$$

при ограничениях (5.58) и

$$\sum_{j=1}^n u_{ij} \leq b_j, \quad i = 1, \dots, m, \quad (5.64)$$

$$\sum_{j=1}^n v_{ij} \leq 1, \quad i = 1, \dots, m. \quad (5.65)$$

Обоснование такого преобразования является простым обобщением процедуры, примененной в аналогичном случае к простейшей задаче размещения.

**Л е м м а 5.10.** *Существует оптимальное решение  $X^*$ ,  $U^*$ ,  $V^*$  задачи (5.63)—(5.65) такое, что*

$$u_{ij}^* + c_{ij} v_{ij}^* = \lambda_j, \quad i \in I(j), \quad j = 1, \dots, n,$$

где  $I(j) = \{i : c_{ij} < \infty\}$ ,  $j = 1, \dots, n$ .

**Доказательство.** Пусть

$$u_{ij}^* + c_{ij} v_{ij}^* = \lambda_j, \quad i \in I(j), \quad j = 1, \dots, n.$$

Положим  $\lambda_j = \min_{i \in I(j)} \lambda_{ij}$  ( $j = 1, \dots, n$ ). Тогда из  $\lambda_{ij} > \lambda_j$  следует, что либо  $u_{ij}^* > 0$ , либо  $v_{ij}^* > 0$ . Если  $u_{ij}^* > 0$ , уменьшим его на  $\delta$ ,  $\delta = \min \{u_{ij}^*, \lambda_{ij} - \lambda_j\}$ . Очевидно, что новое решение, отличающееся от старого только значением  $u_{ij}^*$ , допустимо и оптимально. Если же  $u_{ij}^* = 0$ , аналогичную процедуру выполним для  $v_{ij}^*$ . Повторяя ее для всех  $(i, j)$  таких, что  $\lambda_{ij} > \lambda_j$ , получим решение, удовлетворяющее требуемому свойству.

Легко также видеть, что при любых  $\bar{U}$ ,  $\bar{V}$  таких, что  $\bar{u}_{ij} + c_{ij} \bar{v}_{ij} = \lambda_j$ , блокирующими множествами для задачи являются множества  $I(j)$ , и только они.

На основе приведенных свойств задачи можно построить алгоритм ее решения, сводящийся к решению последовательности задач вида (5.6)—(5.8) или (5.18)—(5.20). Детальное описание и анализ такого алгоритма мы предоставляем читателю.

### § 5. Задачи синтеза сетей с одним источником

В данном разделе рассматриваются некоторые свойства задач, в которых требуется построить коммуникационную сеть минимальной стоимости, соединяющую заданный источник с множеством потребителей при условии, что затраты на создание каждого участка такой сети определяются кусочно линейной неубывающей функцией от потока по этому участку. Такие задачи возникают при проектировании водопроводных, электрических, информационных сетей некоторых классов и в ряде других ситуаций. К аналогичным моделям сводятся и задачи, в которых можно разместить не один, а несколько источников неограниченной сверху мощности и требуется кроме сети коммуникаций выбрать мощности источников, затраты на размещение которых также задаются функциями вышеуказанного типа. В последнем случае достаточно ввести в задачу фиктивный источник, соединенный потенциальными коммуникациями только с местами возможного размещения. Функции затрат на этих коммуникациях полагаются совпадающими с соответствующими им функциями затрат на размещение производства, а затраты на размещение из рассмотрения исключаются.

Пусть  $G(V, E)$  — орграф, каждой дуге  $(i, j) \in E$  которого сопоставлена кусочно линейная неотрицательная неубывающая функция  $f_{ij}(y_{ij})$ , где  $y_{ij}$  — неизвестный поток по дуге  $(i, j)$ , а каждой вершине  $i \in V \setminus \{0\}$  поставлена в соответствие потребность  $B_i \geq 0$ . Вершина 0 — источник. В приведенных обозначениях требуется найти

$$\min \sum_{(i,j) \in E} f_{ij}(y_{ij}) \quad (5.66)$$

при ограничениях

$$\sum_{i \in I(j)} y_{ij}^k - \sum_{i \in I(i)} y_{ji}^k = \begin{cases} B_k & \text{при } j = k, \\ 0 & \text{при } j \neq 0, k, k \in V \setminus \{0\}, \\ -B_k & \text{при } j = 0, \end{cases} \quad (5.67)$$

$$\sum_{k \in V \setminus \{0\}} y_{ij}^k = y_{ij}, \quad (i, j) \in E, \quad (5.68)$$

$$y_{ij}^k \geq 0, \quad (i, j) \in E, \quad k \in V \setminus \{0\}; \quad (5.69)$$

здесь  $I(j)$  и  $J(j)$  — множества заходящих и исходящих дуг для вершины  $j$ .

Как и ранее рассмотренные, данная задача сводится к задаче частично целочисленного линейного программирования на орграфе с параллельными дугами (то есть с дугами, у которых начала и концы одинаковы).

Для простоты изложения и обозначений мы, однако, рассмотрим лишь модель, соответствующую орграфу без параллельных дуг. Более общий случай не привносит дополнительных принципиальных трудностей. Пусть каждой дуге  $(i, j) \in E$  орграфа  $G(V, E)$  поставлены в соответствие два числа:  $a_{ij} \geq 0$  — удельные затраты и  $b_{ij} \geq 0$  — фиксированные доплаты за создание коммуникации  $(i, j)$  с пропускной способностью  $X_{ij}$ . Рассматриваемый аналог задачи (5.66)—(5.69) формулируется в виде: найти

$$\min \sum_{k \in V \setminus \{0\}} \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in E} b_{ij} X_{ij} \quad (5.70)$$

при ограничениях

$$\sum_{i \in J(j)} x_{ij}^k - \sum_{i \in I(i)} x_{ij}^k = \begin{cases} 1 & \text{при } j = k, \\ 0 & \text{при } j \neq 0, k, \quad k \in V \setminus \{0\}, \\ -1 & \text{при } j = 0, \end{cases} \quad (5.71)$$

$$0 \leq x_{ij}^k \leq X_{ij}, \quad (i, j) \in E, \quad k \in V \setminus \{0\}, \quad (5.72)$$

$$X_{ij} = 0 \vee 1, \quad (i, j) \in E. \quad (5.73)$$

Здесь  $c_{ij}^k = a_{ij} b_k$ ,  $(i, j) \in E$ ,  $k \in V \setminus \{0\}$ . Условия (5.71) следуют из (5.67) и из того факта, что поток для вершины  $k$  ( $k \in V \setminus \{0\}$ ) по любой дуге  $(i, j)$  либо равен нулю, либо равен  $b_k$ .

Двойственная декомпозиционная задача к (5.70)—(5.72) записывается в виде: найти

$$\max_{U \geq 0} \min_{X \geq 0} \sum_{k \in V \setminus \{0\}} \sum_{(i,j) \in E} (c_{ij}^k + u_{ij}^k) x_{ij}^k \quad (5.74)$$

при ограничениях (5.71) и

$$\sum_{k \in V \setminus \{0\}} u_{ij}^k \leq b_{ij}, \quad (i, j) \in E. \quad (5.75)$$

Обозначим через  $L_k(\bar{U})$  множество кратчайших путей, соединяющих источник 0 с потребителем  $k$  в графе  $G$  с весами дуг  $(c_{ij}^k + \bar{u}_{ij}^k)$ ,  $k \in V \setminus \{0\}$ , через  $\bar{U}$  — допустимый по (5.75) набор неотрицательных значений переменных  $\bar{u}_{ij}^k$ ,  $(i, j) \in E$ . Блокирующим множеством, определяемым данными  $k$  и  $\bar{U}$ , является минимальное множество дуг таких, что каждый путь из  $L_k(\bar{U})$  содержит хотя бы одну дугу из этого множества. Иными словами, блокирующее множество является разрезом, отделяющим  $k$  от 0 в множестве кратчайших путей  $L_k(\bar{U})$ .

В реальных задачах рассматриваемого типа могут встречаться графы  $G$  с несколькими сотнями и даже тысячами вершин. Естественно, решение таких задач точными методами практически невозможно. Поэтому мы не останавливаемся здесь на методах точного решения оценочной задачи, а рассмотрим лишь приближенные алгоритмы.



Для построения приближенного значения нижней оценки путем решения задачи (5.74), (5.75) можно использовать процедуру, которая обобщает процедуру НТР для простейшей задачи размещения. В ней в циклическом порядке просматриваются вершины-потребители. На текущем шаге выделится множество кратчайших путей в графе  $G$  с весами дуг  $\bar{c}_{ij}^k = c_{ij}^k + \bar{u}_{ij}^k$ , где  $\{\bar{u}_{ij}^k\}$  — построенное на предыдущих шагах множество значений переменных, а  $k$  — индекс просматриваемой вершины. Обозначим через  $G^k$  подграф  $G$ , соответствующий вершинам и дугам  $G$ , лежащим на этих путях. Далее в  $G^k (V^k, E^k)$  выделяется некоторый разрез  $R_k$ , отделяющий источник от вершины  $k$ , обладающий тем свойством, что на его дугах

$$\bar{b}_{ij} = b_{ij} - \sum_{k \in V \setminus \{0\}} \bar{u}_{ij}^k > 0, \quad (i, j) \in E.$$

Если для данного  $k$  разреза с таким свойством нет, вершина  $k$  из дальнейшего рассмотрения исключается. Обозначим  $\delta_1 = \min_{(i, j) \in R_k} \bar{b}_{ij}$ ,

$[0, k]$  — кратчайший путь из 0 в  $k$  в графе  $G^k$ ,  $l_k$  — его длина. Положим  $\delta_2 = \bar{l}_k - l_k$ , где  $\bar{l}_k$  — длина кратчайшего пути из 0 в  $k$  в графе  $G(V, E \setminus R_k)$  с весами дуг  $\bar{c}_{ij}^k$ . Если такого пути нет, положим  $\bar{l}_k = \infty$ .

Пусть  $\delta = \min \{\delta_1, \delta_2\}$ . Вычислим новые значения

$$\bar{u}_{ij}^k = \begin{cases} \bar{u}_{ij}^k + \delta & \text{для } (i, j) \in R_k, \\ \bar{u}_{ij}^k & \text{для } (i, j) \notin R_k. \end{cases}$$

Добавим к текущему значению нижней оценки  $\Phi(\bar{U})$  величину  $\delta$  (на начальном шаге  $\Phi(\bar{U}) = 0$ ,  $\bar{U} = 0$ ) и перейдем к анализу следующего значения  $k$ . Процедура заканчивается, когда все вершины-потребители исключены из рассмотрения.

Отметим, что если на очередном шаге  $\delta = \delta_1$ , то величина  $\bar{b}_{ij}$  становится равной нулю хотя бы для одного из  $(i, j) \in R_k$ . Если же  $\delta = \delta_2$ , то для данного  $k$  расширяется (путем добавления кратчайшего пути длины  $\bar{l}_k$ ) множество  $L_k$  кратчайших путей в  $G$  с весами  $\bar{c}_{ij}^k$ . Легко понять, что в худшем случае данная вершина  $k$  может обрабатываться описанной процедурой  $O(n^2)$  раз, где  $n = |V|$ . Например, если выбирать в качестве очередного разреза  $R_k$  ближайший из допустимых к вершине  $k$ , то в худшем на каждый случай  $\delta = \delta_1$  может приходиться  $O(m)$  случаев  $\delta = \delta_2$ , где  $m$  пробегает все значения от  $n$  до 1. При этом число случаев  $\delta = \delta_1$  будет иметь порядок  $O(n)$ . Трудоемкость одной итерации процедуры при таком способе выбора  $R_k$  оценивается величиной  $O(n^2)$  и определяется трудоемкостью построения разреза  $R_k$  и кратчайшего пути длины  $\bar{l}_k$  в графе  $G(V, E \setminus R_k)$ . Таким образом, общая трудоемкость описанной процедуры оценивается величиной  $n \cdot O(n^2) \cdot O(n^2) = O(n^5)$ . Реализация процедуры в представленном выше виде требует дополнительной памяти порядка  $O(n^3)$  на хранении компонент  $\bar{U}$ .

Требования по памяти могут быть уменьшены до  $O(n^2)$  за счет использования других параметров, определяющих решение задачи построения оценки. Рассмотрим соответствующую переформулировку, введя предварительно следующие обозначения. Пусть  $T_0$  — дерево кратчайших путей на графе  $G$  с весами дуг  $a_{ij}$  из вершины 0 в  $V \setminus \{0\}$ ,  $p_i^0$  — длина пути в  $T_0$  из 0 в  $i$ ,  $i \in V$ . Для каждой вершины  $k$ ,  $k \in V \setminus \{0\}$ , с  $B_k > 0$  в процессе работы процедуры строится дерево  $T^k$  путей, ведущих в вершину  $k$ , с множеством вершин  $V(T^k)$  и дуг  $E(T^k)$ . Одновременно переопределяется дерево кратчайших путей из корня  $T_0$  и соответствующий ему вектор длин кратчайших путей (потенциалов)  $(p_i^0)$ ,  $i \in V$ . Кроме этого, для каждого  $k$  производится разбиение множества вершин  $V$  на подмножества  $P^k$  и  $Q^k$ ,  $P^k \cup Q^k = V$ ,  $P^k \cap Q^k = \emptyset$ .

В начале работы процедуры каждое  $T^k$  состоит из единственной вершины  $k$ , а каждое  $Q^k$  содержит те и только те вершины, которые достижимы из  $k$  в дереве  $T_0$  (в том числе и саму вершину  $k$ ); множество таких вершин обозначим через  $S(T_0, k)$ . На выходе процедура построения тупикового решения (ПТР) определяет значение нижней оценки  $\Phi$  и подграф  $\bar{G}(V, \bar{E})$ , где  $\bar{E} = \{(i, j) : \bar{b}_{ij} = 0\}$ .

Введем вспомогательный переменный вектор  $IP$  длины  $n$ ,  $k$ -я компонента которого равна единице, если для соответствующей вершины  $B_k > 0$  и вершина  $k$  еще не исключена из рассмотрения. В противном случае  $IP(k) = 0$  ( $k = 1, \dots, n$ ). Обозначим через  $n_1$  число ненулевых компонент вектора  $B$ .

**Процедура ПТР.** Вход: граф  $G(V, E)$  с множеством вершин  $V = (0, 1, \dots, n)$ ;  $B = (B_k)$  ( $k = 1, \dots, n$ );  $\{a_{ij}\}$ ,  $\{b_{ij}\}$ ,  $(i, j) \in E$ . Выход:

$$\Phi, \{\bar{b}_{ij}\}, (i, j) \in E, \bar{G}(V, \bar{E}).$$

Шаг 1. Построить ордереро  $T_0$  с корнем 0 и потенциалами  $p_i^0$ ,  $i \in V$ , на графе  $G$  с длинами дуг  $a_{ij}$ ,  $(i, j) \in E$ .

Шаг 2. Для всех  $k$  с  $B_k > 0$  положить:  $T^k = T_0$ ;  $V(T^k) = \{k\}$ ;  $p_i^k = p_i^0 B_k$ ,  $i \in V$ ;  $Q^k = S(T_0, k)$ ;  $P^k = V \setminus Q^k$ ,  $IP(k) = 1$ .

Шаг 3. Положить  $\Phi = 0$ ,  $\bar{b}_{ij} = b_{ij}$ ,  $(i, j) \in E$ ,  $k = 1$ ,  $r = 1$ .

Шаг 4. Если  $IP(k) = 0$ , перейти к шагу 9.

Шаг 5. Найти

$$\delta_1 = \min_{\substack{i \in P^k \\ j \in V(T^k)}} \bar{b}_{ij} = \bar{b}_{i, j_1},$$

$$\delta_2 = \min_{\substack{i \in P^k \\ j \in Q^k \setminus V(T^k)}} (p_j^k - p_i^k - a_{ij} B_k) = p_{j_2}^k - p_{i_2}^k - a_{i_2, j_2} B_k,$$

$$\delta = \min \{\delta_1, \delta_2\}.$$

Шаг 6. Положить  $\Phi := \Phi + \delta$ ,  $p_i^k := p_i^k + \delta$ ,  $i \in Q^k$ ,  $\bar{b}_{ij} := \bar{b}_{ij} - \delta$ ,  $(i, j) \in E$ ,  $i \in P^k$ ,  $j \in V(T^k)$ .

Шаг 7. Если  $\delta = \delta_1$ , положить  $V(T^k) := V(T^k) \cup i_1$ ,  $E(T^k) := E(T^k) \cup (i_1, j_1)$ ,  $Q^k := Q^k \cup S(T^k, k)$ ,  $P^k = V \setminus Q^k$  и перейти к шагу 11.

Шаг 8. Если  $\delta = \delta_2$ , положить  $T^k := T^k \cup (i_2, j_2) \setminus (l, j_2)$ , где  $l$  — вершина, непосредственно предшествующая  $j_2$  в «старом»  $T_0^k$ . Положить  $P^k := P^k \cup S(T^k, i_2)$ ,  $Q^k = V \setminus P^k$ .

Шаг 9. Положить  $k := k + 1$ . Если  $k > n$ , положить  $k = 1$ .

Шаг 10. Перейти к шагу 4.

Шаг 11. Если  $i_1 \neq 0$ , перейти к шагу 9.

Шаг 12. Положить  $IP(k) = 0$ ,  $r := r + 1$ .

Шаг 13. Если  $r = n_1$ , перейти к шагу 14, иначе — к шагу 9.

Шаг 14. Построить  $\bar{G}(V, \bar{E}) = \bigcup_{k: B_k > 0} T^k$ .

Отметим, что в  $\bar{G}$  существует путь из 0 в любую вершину  $k$ , для которой  $B_k > 0$ .

Трудоёмкость описанной процедуры —  $O(n^5)$  — достаточно высока для ее практического использования. Рассмотрим вариант этой процедуры, трудоёмкость которой есть  $O(n^3)$ . Однако при этом приходится отказаться от циклического просмотра вершин, а рассматривать каждую вершину  $k$  с  $B_k > 0$  только один раз, увеличивая максимально длину кратчайшего пути при заданных величинах  $\bar{b}_{ij}$ , полученных в результате просмотра предшествующих вершин. Полученная таким путем нижняя оценка функционала будет, вообще говоря, хуже, чем полученная с помощью вышеуказанной процедуры. Рассматриваемый вариант основан на том простом факте, что при заданных  $\bar{b}_{ij}$ ,  $(i, j) \in E$  максимальный вклад в значение функционала равен длине кратчайшего пути  $[0, k]$  в  $G(V, E)$  с весами дуг  $a_{ij}B_k + \bar{b}_{ij}$ . Остается определить новые значения  $\bar{b}_{ij} = \bar{b}_{ij} - \bar{u}_{ij}^k$ , где  $\bar{u}_{ij}^k$ ,  $0 \leq \bar{u}_{ij}^k \leq \bar{b}_{ij}$ , — та часть  $\bar{b}_{ij}$  для каждой дуги  $(i, j) \in E$ , которую следует добавить к  $a_{ij}B_k$ , чтобы кратчайший путь из 0 в  $k$ , проходящий по  $(i, j)$ , был не короче, чем путь  $[0, k]$ .

Приведем описание модифицированной процедуры.

**Процедура ПТРМ.** Вход и выход такие же, как в ПТР.

Шаг 1. Положить  $\Phi = 0$ ,  $\bar{b}_{ij} = b_{ij}$ ,  $(i, j) \in E$ ,  $k = 1$ .

Шаг 2. Если  $B_k = 0$ , перейти к шагу 7.

Шаг 3. В графе  $G$  с весами дуг  $a_{ij}B_k + \bar{b}_{ij}$  построить дерево  $T_k$  кратчайших путей в вершину  $k$  с потенциалами  $q_i$ ,  $i \in V \setminus k$ ,  $q_k = 0$ . Потенциал  $q_0$  — длина кратчайшего пути из 0 в  $k$ .

Шаг 4. Положить  $p_0 = 0$ ,

$$p_j = \max \{(q_0 - q_j), \min_{(i,j) \in E} (p_i + a_{ij}B_k)\}, \quad j \in V \setminus 0. \quad (5.76)$$

При этом вершины  $j$  просматриваются в таком же порядке, как в алгоритме Дейкстры [79], т. е. последовательность  $\{p_j\}$  — неубывающая.

Шаг 5. Для каждой дуги  $(i, j) \in E$  положить

$$\bar{b}_{ij} = \min \{\bar{b}_{ij}, (\bar{b}_{ij} - (p_j - p_i) + a_{ij}B_k)\}.$$

Шаг 6. Положить  $\Phi := \Phi + q_0$ .

Шаг 7. Положить  $k := k + 1$ . Если  $k < n$ , перейти к шагу 2.

Шаг 8. Положить  $\bar{E} = \{(i, j) : \bar{b}_{ij} = 0\}$ .

Если применять для выполнения шага 4 модификацию алгоритма Дейкстры, которая заключается в учете лишь компоненты  $q_0 - q_j$  при вычислении  $p_j$ , то трудоемкость выполнения пунктов 1—3 не превзойдет  $O(n^2)$ . Отсюда общая трудоемкость составит  $O(n^3)$ . Значения  $\bar{u}_{ij}^k$  (которые не фигурируют в процедуре) вычисляются по формуле

$$\bar{u}_{ij}^k = \max \{0, p_j - p_i - a_{ij}B_k\}, \quad (i, j) \in E. \quad (5.77)$$

Покажем, что описанная процедура строит для каждого просматриваемого  $k$  допустимое тупиковое решение  $\{\bar{u}_{ij}^k\}$ .

Теорема 5.5. 1. Значения  $\bar{u}_{ij}^k$  удовлетворяют ограничениям

$$0 \leq \bar{u}_{ij}^k \leq \bar{b}_{ij}, \quad (i, j) \in E.$$

2. Длина любого пути из 0 в  $k$  в графе  $G$  с весами дуг  $(a_{ij}B_k + \bar{u}_{ij}^k)$ ,  $(i, j) \in E$ , не менее чем  $q_0$  и уменьшение значения  $\bar{u}_{st}^k$  для любой дуги  $(s, t) \in E$  ведет к уменьшению длины кратчайшего пути  $[0, k]$ .

Доказательство. 1. Если для некоторой дуги  $(i, j) \in E$   $p_j \neq q_0 - q_j$ , то  $\bar{u}_{ij}^k = 0$ , так как в этом случае из (5.77) следует  $a_{ij}B_k \geq p_j - p_i$ . Пусть теперь  $p_j = q_0 - q_j$ . Тогда из неравенств  $p_i \geq q_0 - q_i$  и  $q_i \leq q_j + a_{ij}B_k + b_{ij}$  имеем  $p_j - p_i \leq q_i - q_j \leq a_{ij}B_k + \bar{b}_{ij}$ , откуда  $\bar{u}_{ij}^k \leq \bar{b}_{ij}$ .

2. Для произвольной дуги  $(s, t)$  с  $\bar{u}_{st}^k > 0$  путь, состоящий из  $(s, t)$  и кратчайшего пути  $[t, k]$ , является кратчайшим из  $s$  в  $k$  в графе  $G$  с весами дуг  $a_{ij}B_k + \bar{u}_{ij}^k$ . Если  $p_s = q_0 - q_s$ , то длина кратчайшего пути из 0 в  $k$ , проходящего по  $(s, t)$ , равна  $q_0 - q_s + a_{st}B_k + \bar{u}_{st}^k + q_t = q_0$ . Поэтому уменьшение значения  $\bar{u}_{st}^k$  приведет к уменьшению длины кратчайшего пути из 0 в  $k$ . Если  $p_s > q_0 - q_s$ , то длина кратчайшего пути из 0 в  $t$ , проходящего по дуге  $(s, t)$ , равна длине кратчайшего пути  $[0, t]$ , т. е.  $q_0 - q_t$ , а длина кратчайшего пути  $[t, k]$  равна  $q_t$ . Следовательно, длина кратчайшего пути из 0 в  $k$ , проходящего по  $(s, t)$ , равна  $q_0 - q_t + q_t = q_0$ . Тем самым и в этом случае уменьшение  $\bar{u}_{st}^k$  невозможно без уменьшения  $q_0$ .

Процедура ПТРМ строит допустимое множество значений  $\bar{b}_{ij}$  для каждой вершины  $k$ . Однако это множество не дает максимально возможную величину  $\sum_{(i,j) \in E} \bar{b}_{ij}$ . Вообще говоря, при использовании проце-

дур подобного типа для построения оценок функционала следует максимизировать на каждом шаге суммарный остаток  $b_{ij}$  лишь по тем дугам, для которых на последнем шаге величины  $\bar{b}_{ij}$  становятся равными нулю. Однако, поскольку это множество заранее неизвестно,

можно искать такое распределение  $\bar{u}_{ij}^k$  при каждом  $k$ , которое максимизирует суммарный остаток величин  $\bar{b}_{ij}$  или, что эквивалентно, минимизирует  $\sum_{(i,j) \in E} \bar{u}_{ij}^k$ . Соответствующая оптимизационная задача при фиксированном  $k$  записывается в виде: найти

$$\min \sum_{(i,j) \in E} u_{ij} \quad (5.78)$$

при ограничениях

$$\bar{a}_{ij} + u_{ij} \geq p_j - p_i, \quad (i, j) \in E, \quad (5.79)$$

$$0 \leq u_{ij} \leq \bar{b}_{ij}, \quad (i, j) \in E, \quad (5.80)$$

$$p_0 = 0, \quad p_k = q_0, \quad (5.81)$$

где  $\bar{a}_{ij} = a_{ij} B_k$ ,  $(i, j) \in E$ ,  $u_{ij}$ ,  $(i, j) \in E$  и  $p_i$ ,  $i \in V$  — искомые неизвестные величины. Ограничения (5.79) отражают то естественное требование, что длина кратчайшего пути из 0 в  $j$  (равная  $p_j$ ) не превосходит длины кратчайшего пути из 0 в  $i$  плюс длина дуги  $(i, j)$  в графе с весами дуг  $\bar{a}_{ij} + u_{ij}$ . Ограничение (5.81) эквивалентно тому, что длина кратчайшего пути из 0 в  $k$  равна заданной величине  $q_0$ .

Трудоёмкость точного решения задачи (5.78)—(5.81), по-видимому, существенно выше чем  $O(n^2)$  — трудоёмкости процедуры ПТРМ, строящей ее приближенное решение. Поэтому для того чтобы рекомендовать использование точного решения в процессе построения нижней оценки, необходим достаточно представительный вычислительный эксперимент.

На основе процедуры ПТРМ разработан алгоритм решения задачи (5.70)—(5.73) методом ветвей и границ, близкий по своей структуре к вышеописанным алгоритмам для рассмотренных классов задач размещения. Он реализован в виде АССЕМБЛЕР-программы для ЕС ЭВМ. Программа работает с мультиграфами, функции затрат на дугах которых задаются формулой

$$\varphi_{ij}^k(x_{ij}) = \begin{cases} \alpha_k r_{ij} x_{ij} + \beta_k r_{ij} & \text{при } x_{ij} > 0, \\ 0 & \text{при } x_{ij} = 0, \end{cases} \quad k = 1, \dots, K, \quad (5.82)$$

где  $k$  — индекс мультидуги, соединяющей данную пару вершин. Предполагается, что коэффициенты  $\alpha_k$  и  $\beta_k$  одинаковы для всех пар  $(i, j)$ , которые соединяются одинаковым количеством мультидуг,  $r_{ij}$  — длина дуги из  $i$  в  $j$ . Таким образом, как линейные затраты, так и фиксированные доплаты пропорциональны в (5.82) длине коммуникации  $r_{ij}$ . Кроме того, предусмотрен способ задания коэффициентов  $a_{ij}^k$  и  $b_{ij}^k$  в формуле

$$\varphi_{ij}^k(x_{ij}) = \begin{cases} a_{ij}^k x_{ij} + b_{ij}^k & \text{при } x_{ij} > 0, \\ 0 & \text{при } x_{ij} = 0 \end{cases}$$

с помощью подпрограммы пользователя. Число параллельных мультидуг ограничено в программе десятью. Тестовые просчеты проводились

$n$	$\Phi$	$F$	$P$	$Q$	$\theta$	$n$	$\Phi$	$F$	$P$	$Q$	$\theta$
20	199	199	0	0	+	75	2883	2929	4	1	+
	350	414	263	13	+		3454	3455	2	1	+
	506	506	0	0	+		4059	4093	0	0	—
	558	558	0	0	+		4233	4233	0	0	+
	698	699	10	1	+		4480	4592	0	0	—
30	341	341	0	0	+	90	1259	1264	0	0	—
	508	537	21	3	+		1832	1922	0	0	—
	852	860	—	0	—		2768	2785	8	2	+
	1000	1000	0	0	+		3671	3690	0	0	—
	1196	1196	0	0	+		3739	3741	5	1	+
	1222	1222	0	0	+		4640	4640	0	0	+
40	38797	38797	0	0	+	122	4681	4767	0	0	—
	53372	53372	0	0	+		5068	5279	0	0	—
	75406	75414	38	1	+		2188	2194			
	98343	98608	2	1	+		3154	3426			
	109751	109751	0	0	+		5136	5153			
	122538	122578	204	1	+		5274	5288			
	131962	1319C2	0	0	+		6728	6739			
50	526	526	0	0	+	150	8256	8273			
	673	692	—	—	—		8344	8495			
	793	793	0	0	+		8809	9242			
	928	1026	23	5	+		3089	3108			
	1665	1709	—	—	—		4212	4684			
	2037	2070	—	—	—		7338	7381			
	2475	2494	27	1	+		9815	9838			
	2701	2859	37	2	+		10027	10228			
75	980	988	0	0	—	12063	12374				
	2000	2176	82	7	—	13186	13261				
	2009	2162	0	0	—	15375	15980				

на ЕС-1030 на случайно сгенерированных графах и на ряде реальных примеров. Числовые данные представлены в табл. 5.6. Здесь  $n$  — число вершин графа,  $\Phi$  — начальная нижняя оценка,  $F$  — начальная верхняя оценка,  $P$  — число ветвлений,  $Q$  — число улучшений рекорда. Знак  $+$  в столбце  $\theta$  означает, что задача решена точно. Для задач с  $n = 122, 150$  просчитывался только один шаг метода ветвей и границ. Для одного и того же  $n$  задачи различаются лишь значениями коэффициентов  $\alpha_k, \beta_k$  ( $k = 1, \dots, K$ ). Во всех задачах  $K = 3$ . Задачи для  $n = 40$  соответствуют реальным исходным данным.

В табл. 5.7 приведены аналогичные данные для тестовых задач, в которых все коэффициенты  $\alpha_k$  равны 0 (задачи Штейнера на графе). Среднее время вычисления на одном шаге для решенных задач из табл. 5.6, 5.7 приведено в табл. 5.8 в строках 2, 3 соответственно. В ней также приведено среднее общее число вершин с  $B_i = 0$  и число таких вершин, включенных в оптимальное или рекордное решение (строки 4, 5, соответственно). Строка 1 содержит значения  $n$ .

Таблица 5.7

$n$	$\Phi$	$F$	$P$	$Q$	$\theta$	$n$	$\Phi$	$F$	$P$	$Q$	$\theta$
20	57	57	0	0	+	75	507	528	—	0	—
	68	68	0	0	+		768	784	18	3	+
	116	139	47	4	+		834	836	2	1	+
	186	186	0	0	+		1014	1021	—	0	—
	231	238	3	1	+		1056	1056	0	0	+
30	167	192	68	7	+	90	678	694	—	0	—
	284	285	4	1	+		1238	1247	—	0	—
	290	299	0	0	+		1406	1460	—	0	—
	306	306	0	0	+		1682	1696	4	1	+
	460	460	27	3	+		4368	4416	1	0	+
40	387	387	0	0	+	122	479	482	—	0	—
	746	746	0	0	+		485	492	—	0	—
	984	985	38	1	+		647	657	—	0	—
	1192	1207	27	2	+		749	758	—	0	—
	1975	1986	12	1	+		818	823	—	0	—
50	143	150	20	2	+	150	586	600	—	0	—
	246	260	60	4	—		645	652	—	0	—
	367	367	0	0	+		678	681	—	0	—
	409	428	8	1	+		849	862	—	0	—
	671	673	13	1	+		941	953	—	0	—

Таблица 5.8

1	20	30	40	50	75	90	122	150
2	1,3	2,4	5,1	9	30	40	80	125
3	0,5	0,9	1,8	4	12	16	32	50
4	8	11	7	22	26	33	—	—
5	2	3	5	5	10	12	—	—

Программная реализация описанного алгоритма осуществлена также в одной из организаций Госстроя УССР М.А. Быковченко. Она использовалась для решения ряда практических задач по оптимизации региональных схем водоснабжения в районной планировке [14]. Проведенные расчеты для ряда областей УССР показали достаточно высокую ее эффективность по сравнению с вариантами, разработанными традиционными методами без использования ЭВМ. Суммарные народнохозяйственные затраты в полученных решениях сокращаются на 3—15 %.

Как уже отмечалось, частным случаем задачи синтеза древовидных сетей является задача Штейнера на графах. Одним из практических приложений последней является задача проектирования монтажных соединений на печатных платах. В этой задаче задано множество точек на плате, которые должны быть соединены между собой так, чтобы суммарная длина соединений была минимальна. Особенностью этой задачи является возможность проведения соединений лишь с помощью отрез-

Таблица 5.9

<i>N</i>	<i>n</i>	$\Delta$	<i>P</i>	<i>Q</i>	<i>R</i>	$\delta$	<i>t</i>
1	20	3	6,78	11	2	2,8	1,4
2	20	3	1,3	1	1	1,3	0,12
3	20	3	2,4	1	1	2,4	0,12
4	20	3	4,14	14	2	1,4	1,7
5	20	3	1,05	1	1	1,05	0,12
6	20	3	1,64	1	1	1,64	0,12
7	20	3	2,82	1	1	2,82	0,12
8	20	3	7,14	193	9	3,0	25
9	20	3	9	26	2	3,0	3
10	20	3	5	11	3	2,6	1,3
11	20	3	7	20	3	2,0	2,5
12	20	3	3,3	40	3	3,0	5
13	20	3	9	96	6	3,0	12,5
14	30	3	4	12	2	3,0	8
15	30	3	2,5	1	1	2,5	0,65
16	30	3	7	35	3	1,7	22,7
17	30	3	4,5	13	1	2,9	8,5
18	40	3	3,7	13	1	2,9	17,3
19	40	3	3,98	5	1	2,75	6,18
20	50	3	2,7	1	1	2,7	3,3
21	50	3	7,4	30	1	3,6	100
22	100	3	8,5	1	1	8,5	120

ков прямых, параллельных границам прямоугольной платы. Математически такая задача сводится к построению дерева Штейнера на плоскости с прямоугольной метрикой, в которой расстояние между двумя заданными точками  $X$  и  $Y$  с координатами  $(x_1, x_2)$  и  $(y_1, y_2)$  определяется по формуле  $r(X, Y) = |x_1 - y_1| + |x_2 - y_2|$ .

В ряде работ (см., например, [41]) отмечено, что существующие методы решения задачи Штейнера на плоскости с прямоугольной метрикой неэффективны и позволяют решать ее лишь для случаев, когда число заданных точек не превышает десяти. Был проведен вычислительный эксперимент по использованию метода динамической декомпозиции для построения оценок и решения указанного класса задач. Для этого разработана фортран-программа, реализующая способы построения оценок и схему ветвления с учетом особенностей задачи Штейнера на плоскости с прямоугольной метрикой. Эксперимент проводился на случайных задачах, в которых множество координат основных точек генерировалось с помощью датчика псевдослучайных чисел.

Результаты эксперимента приведены в табл. 5.9. Тип ЭВМ — ЕС-1040. Здесь  $n$  — число заданных точек,  $\Delta, \delta$  — задаваемая и полученная точность в процентах,  $P$  — относительная погрешность решения, полученного на первой итерации

$$P = \frac{F - \Phi}{F} \cdot 100 \%,$$



где  $F$  — верхняя оценка,  $\Phi$  — нижняя оценка,  $Q$  — количество шагов в методе ветвей и границ,  $R$  — количество улучшений рекорда,  $t$  — время в мин.

## § 6. Задача синтеза надежных сетей

Одним из важных для практики и одновременно малоизученных в настоящему времени классов задач являются задачи оптимального синтеза надежных сетей. Задачи такого типа возникают при проектировании систем связи, дорог, тепло- и продуктопроводов различных видов и т. д. В отличие от обычных задач синтеза сетей в них предполагается, что элементы системы могут время от времени выходить из строя, нарушая тем самым нормальное функционирование всей системы или ее части; в таком случае система должна работать определенное время в аварийном режиме, пока не будут устранены последствия поломки или аварии вышедшего из строя элемента. При работе в аварийном режиме система должна обеспечивать пониженный, как правило, уровень требований потребителей (например, поступление тепла, пропускную способность каналов связи для пар абонентов и т. д.). Имеется ряд работ [1, 64], в которых задача синтеза надежных сетей рассматривается в вероятностном аспекте. В них предполагаются заданными вероятности отказа элементов системы. Однако при решении практических задач получение и обоснование соответствующих статистических данных кажется, на наш взгляд, затруднительным. Поэтому в данной работе рассматривается другой подход, связанный с понятием неопределенности. Предполагается, что элементы системы выходят из строя в неопределенные моменты времени, однако в любой момент может выйти из строя не более одного элемента.

Рассматривается задача синтеза надежной сети с одним источником, хотя используемые здесь идеи могут быть непосредственно перенесены на более общий случай. Мы предполагаем, что стоимость сети складывается из капитальных затрат на создание сети, которые являются вогнутыми кусочно линейными функциями от пропускной способности дуг сети, и затрат на эксплуатацию сети, линейно зависящих от потока по дугам в режиме нормальной эксплуатации. Вообще говоря, эксплуатационные затраты являются выпуклыми функциями от потока, но учет выпуклости приводит к усложнению соответствующей модели. С другой стороны, задача с выпуклыми функциями может быть заменена задачей с линейными функциями (в этом случае можно считать погрешность по функционалу) либо сведена к решению последовательности задач с линейными от потока эксплуатационными затратами. Мы не учитываем также изменение величины эксплуатационных затрат при работе системы в аварийном режиме, предполагая, что доля общего времени такой работы мала, и считая, что в условиях неопределенности эта погрешность в принципе невычислима.

Пусть  $0$  — индекс вершины-источника сети,  $N = \{1, 2, \dots, n\}$  — множество индексов вершин-потребителей,  $E$  — множество допустимых ориентированных соединений источника с потребителями и пар потребителей друг с другом. Тогда задачу синтеза надежной сети

можно сформулировать как задачу отыскания

$$\min \sum_{(i,j) \in E} (f_{ij}(\max\{x_{ij}, y_{ji}\}) + c_{ij}|x_{ij} - y_{ji}|) \quad (5.83)$$

при ограничениях

$$\sum_{i \in P(i)} x_{ij} - \sum_{j \in Q(i)} x_{ji} = X_i, \quad i \in N, \quad (5.84)$$

$$\sum_{i \in P(i)} y_{ij} - \sum_{j \in Q(i)} y_{ji} = Y_i, \quad i \in N, \quad (5.85)$$

$$X_i + Y_i \geq A_i, \quad X_i \geq B_i, \quad Y_i \geq B_i, \quad i \in N. \quad (5.86)$$

Здесь  $x_{ij}$ ,  $y_{ij}$  — неизвестные потоки в аварийном режиме по дугам  $(i, j) \in E$ . Если в оптимальном решении задачи  $x_{ij} > 0$ ,  $y_{ij} > 0$  для некоторой дуги  $(i, j)$ , то это означает необходимость включения в оптимальную сеть двух параллельных коммуникаций с пропускной способностью  $x_{ij}$  и  $y_{ij}$  соответственно (при этом  $x_{ji} = y_{ji} = 0$ ). В нормальном режиме функционирования системы потоки по этим коммуникациям также равны  $x_{ij}$ ,  $y_{ij}$ . Если  $x_{ij} > 0$ ,  $y_{ij} = 0$ , то в сеть включается одна коммуникация с пропускной способностью, равной  $\max\{x_{ij}, y_{ji}\}$ , в этом случае  $x_{ji} = 0$ . Поток в нормальном режиме равен  $|x_{ij} - y_{ji}|$  и направлен от  $i$  к  $j$ , если  $x_{ij} > y_{ji}$ , и от  $j$  к  $i$  при  $y_{ji} > x_{ij}$ . Аналогичная картина с точностью до перестановки  $x_{ij}$  и  $y_{ji}$  имеет место и при  $y_{ij} > 0$ ,  $x_{ij} = 0$ . Величины  $X_i$ ,  $Y_i$  в (5.84) — (5.86) — неизвестные объемы продукта, поступающие в  $i \in N$  по  $x$ - и  $y$ -коммуникациям. Ограничения (5.86) требуют, чтобы суммарный поступающий объем в нормальном режиме был не меньше, чем заданная нормальная потребность  $A_i$ , а в аварийном режиме (при выходе из строя любой коммуникации) — не меньше аварийной потребности  $B_i$ ,  $i \in N$ .

Целевая функция (5.83) состоит из двух компонент: первая — сумма кусочно линейных вогнутых неубывающих положительных функций  $f_{ij}(\max\{x_{ij}, y_{ji}\})$  капитальных затрат от пропускной способности по дуге  $(i, j)$ , вторая — сумма линейных функций с положительными коэффициентами эксплуатационных затрат от потока по дуге  $(i, j)$ . Предполагается, что  $f_{ij}(t) = f_{ji}(t)$  для любой дуги  $(i, j)$ . Это соответствует тому факту, что пропускная способность дуги  $(i, j)$  равна пропускной способности дуги  $(j, i)$ . Поскольку эксплуатационные затраты для потока по дуге  $(i, j)$  не должны быть, вообще говоря, равными затратам для  $(j, i)$ , мы не предполагаем, что  $c_{ij} = c_{ji}$ .

Разобьем  $N$  на  $N_1 = \{i : A_i > 2B_i\}$  и  $N_2 = \{i : A_i \leq 2B_i\}$ .

Очевидно, ограничения (5.86) эквивалентны

$$X_i + Y_i = A_i, \quad X_i \geq B_i, \quad Y_i \geq B_i, \quad i \in N_1, \quad (5.87)$$

$$X_i = Y_i = B_i, \quad i \in N_2. \quad (5.88)$$

Для упрощения дальнейшего изложения рассмотрим лишь случай, когда  $N = N_2$  ( $N_1 = \emptyset$ ), т. е. когда аварийная потребность каждой

вершины  $i \in N$  не меньше, чем половина нормальной ее потребности. Все изложенное ниже без принципиальных затруднений переносится на общий случай. Для решения задачи (5.83)—(5.86) применима схема ветвей и границ. Детализация этой схемы на этапе построения нижней оценки функционала (5.83) методом динамической декомпозиции приводит к анализу свойств задачи (5.83)—(5.86) с линеаризованной целевой функцией. Эта задача (с линейным функционалом) возникает, как и в ранее рассмотренных случаях, на начальной итерации метода динамической декомпозиции при вычислении начального значения функционала, а затем на каждой последующей итерации для определения величины, на которую следует увеличить значения двойственных переменных, соответствующих выбранному на этой итерации блокирующему множеству. Поэтому необходимо иметь эффективный алгоритм ее решения.

Рассмотрим эквивалентную постановку задачи (5.83)—(5.85), (5.88) с линейными затратами на синтез сети, т. е. при

$$f_{ij}(\max\{x_{ij}, y_{ji}\}) = b_{ij} \max\{x_{ij}, y_{ji}\}, \quad (i, j) \in E.$$

Она формулируется следующим образом: найти

$$\min \sum_{(i,j) \in E} [b_{ij}(x_{ij} + u_{ij} + z_{ij}) + c_{ij}(x_{ij} + u_{ij})] \quad (5.89)$$

при ограничениях

$$\sum_{i \in P(i)} (x_{ij} + z_{ij}) - \sum_{i \in Q(i)} (x_{ji} + z_{ji}) = B_i, \quad i \in N, \quad (5.90)$$

$$\sum_{i \in P(i)} (u_{ij} + z_{ij}) - \sum_{i \in Q(i)} (u_{ji} + z_{ji}) = B_i, \quad i \in N, \quad (5.91)$$

$$x_{ij} \geq 0, \quad u_{ij} \geq 0, \quad z_{ij} \geq 0, \quad (i, j) \in E. \quad (5.92)$$

Здесь каждая переменная  $u_{ij}$  соответствует переменной  $y_{ji}$ , а каждая  $z_{ij}$  в оптимальном решении равна  $\min\{x_{ij}^*, y_{ji}^*\}$ , где  $x_{ij}^*, y_{ji}^*$  — значения соответствующих переменных в оптимальном решении задачи (5.83) — (5.85), (5.88),  $(i, j) \in E$ . Эквивалентность рассматриваемой пары задач следует из того, что в оптимальном решении  $(X^*, U^*, Z^*)$  для любой дуги  $(i, j) \in E$  значения переменных  $x_{ij}^*$  и  $u_{ij}^*$  не могут быть отличны от нуля одновременно при  $c_{ij} > 0$ . Действительно, если это не так, то, увеличивая значение  $z_{ij}^*$  на  $\delta = \min\{x_{ij}^*, y_{ji}^*\}$  и уменьшая  $x_{ij}^*, u_{ij}^*$  на  $\delta$ , получим новое допустимое решение с меньшим значением функционала. Отсюда следует, что на оптимальном решении задачи (5.89) — (5.92) значение  $x_{ij}^* + u_{ij}^* + z_{ij}^*$  равно  $\max\{x_{ij}^*, y_{ji}^*\}$ , а  $x_{ij}^* + u_{ij}^* = \max\{x_{ij}^* - y_{ji}^*, 0\}$ .

Покажем теперь, как получить решение задачи (5.89)—(5.92) по решению некоторой матричной транспортной задачи. Отметим, что потоки по дугам, соответствующие переменным каждой группы  $X, U, Z$ ,

представимы в виде сумм потоков, соответствующих кратчайшим путям из вершины 0, в вершину 0 и между некоторыми парами  $(i, j)$  вершин соответственно. При этом кратчайшие пути для  $X$ - и  $U$ -потоков вычисляются на графе  $G$  с весами дуг  $(b_{ij} + c_{ij})$ , а кратчайшие пути для  $Z$ -потоков — на графе  $G$  с весами дуг  $b_{ij}$ . Построим векторы  $(d_i^+)$ ,  $(d_i^-)$  ( $i = 1, \dots, n$ ) длин кратчайших путей из 0 в  $i$  и из  $i$  в 0 и матрицу  $H = (h_{ij})$  длин кратчайших путей из  $i$  в  $j$ ,  $i \neq j$ ,  $i, j = 1, \dots, n$ . Если потребность вершины  $i$  в оптимальном решении задачи (5.89) — (5.92) удовлетворяется полностью или частично  $X$ -потоком (аналогично  $U$ -потоком), то стоимость такого потока равна  $d_i^+ x_i$  (аналогично  $d_i^- u_i$ ), где  $x_i$ ,  $u_i$  — величины этих потоков. Если же вершина  $i$  получает всю или часть потребности с помощью  $Z$ -потока, идущего из вершины  $j$ , то стоимость его равна  $h_{ji} z_{ji}$ , где  $z_{ji}$  — его величина. В последнем случае к этой стоимости следует добавить затраты, связанные с доставкой  $z_{ji}$  единиц из 0 в  $j$  и из  $i$  в 0. Поскольку эта доставка осуществляется по кратчайшим путям, общие затраты на  $Z$ -поток из  $j$  в  $i$  равны  $(h_{ji} + d_i^- + d_j^+) z_{ji}$ . С учетом сказанного задача (5.83) — (5.85), (5.88) эквивалентна задаче отыскания

$$\min \sum_{i=1}^n ((d_i^+ x_i + d_i^- u_i) + \sum_{\substack{j=1 \\ i \neq j}}^n (h_{ji} + d_i^- + d_j^+) z_{ji})$$

при ограничениях

$$x_i + \sum_{\substack{j=1 \\ i \neq j}}^n z_{ji} = B_i, \quad i = 1, \dots, n,$$

$$u_j + \sum_{\substack{i=1 \\ i \neq j}}^n z_{ji} = B_j, \quad j = 1, \dots, n,$$

$$x_i \geq 0, \quad u_i \geq 0, \quad z_{ji} \geq 0, \quad i \neq j, \quad i, j = 1, \dots, n.$$

Исключая отсюда переменные  $x_i$ ,  $u_j$ , получим вариант открытой матричной транспортной задачи вида: найти

$$\max \sum_{i,j=1}^n k_{ij} z_{ij}$$

при ограничениях

$$\sum_{j=1}^n z_{ij} \leq B_i, \quad i = 1, \dots, n,$$

$$\sum_{i=1}^n z_{ij} \leq B_j, \quad j = 1, \dots, n,$$

$$z_{ij} \geq 0, \quad i, j = 1, \dots, n,$$

где  $k_{ij} = -(h_{ij} + d_i^+ + d_j^- - d_i^+ - d_i^-)$  ( $i, j = 1, \dots, n$ ). Очевидно, если все  $k_{ij} < 0$ , то последняя задача имеет тривиальное нулевое решение. Этим свойством обладает, например, задача с симметричной матрицей  $C = (c_{ij})$ . В этом случае из симметричности матрицы  $B = (b_{ij})$  следует, что  $d_i^+ = d_i^-$  ( $i = 1, \dots, n$ ), а  $k_{ij} = -h_{ij} \leq 0$ . Оптимальное решение исходной задачи с таким свойством определяется деревом кратчайших путей из корня и получается путем добавления к нему аналогичного дерева с противоположной ориентацией.

Легко понять, что задача (5.89)—(5.92) обладает многими свойствами транспортных задач. В частности, имеют место следующие свойства.

1. Если все  $B_i$  целочисленны, то базисное оптимальное решение задачи (5.89)—(5.92) также целочисленно.

2. Число ненулевых компонент в базисном оптимальном решении  $n(X^*, U^*, Z^*)$  при условии  $B_i > 0$  ( $i = 1, \dots, n$ ) ограничено следующими величинами:  $n \leq n(X^*, U^*, Z^*) \leq 2n - 1$ .

**Доказательство.** Левое неравенство следует из того, что в каждую вершину  $i$  ( $i = 1, \dots, n$ ) должна входить хотя бы одна дуга, соответствующая ненулевой переменной, а из каждой вершины  $i$  ( $i = 1, \dots, n$ ) — выходить хотя бы одна такая дуга. Минимальное множество дуг образует цикл с числом дуг, равным  $n$ . Правое неравенство является следствием базисности решения.

# ПОЛИНОМИАЛЬНО РАЗРЕШИМЫЕ ЗАДАЧИ, СВЯЗАННЫЕ С РАЗМЕЩЕНИЕМ ПРОИЗВОДСТВА И СИНТЕЗОМ СЕТЕЙ

## § 1. Градиентные алгоритмы и кратчайшие ветвления в орграфах

Большинство алгоритмов, решающих комбинаторные экстремальные задачи за полиномиальное от длины входной информации время, можно разбить на три основных класса. В первый из них входят так называемые градиентные алгоритмы, или алгоритмы покоординатной минимизации. Эти алгоритмы наиболее просты по структуре и оптимизирующей части, которая состоит из просмотра переменных задачи (прямой или двойственной) в некотором определенном данными задачи порядке и присвоения очередной переменной ее максимального или минимального (в зависимости от рассматриваемого критерия) допустимого значения. Присвоенное таким образом значение переменной в дальнейшем уже не изменяется. Впервые алгоритмы такого типа для задачи построения минимального связывающего дерева на графе были предложены Краскалом [149] и Примом [58]. Впоследствии Дейкстра [119] предложил подобный алгоритм для задачи построения дерева кратчайших путей на графе, а Эдмондс [122] — для задачи построения кратчайшего ориентированного дерева с корнем (кратчайшего ветвления) на орграфе. Эдмондс [125] исследовал градиентные алгоритмы для задач нахождения независимого множества максимального веса в матроидах и полиматроидах и доказал, что они строят точное решение.

Градиентные алгоритмы используются не только для точного решения комбинаторных задач, но и для приближенного решения с гарантированной относительной погрешностью. Например, в [163] рассмотрена задача, близкая к простейшей задаче размещения: найти

$$\max \left( \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} - \sum_{i=1}^m b_i X_i \right)$$

при ограничениях

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

$$0 \leq x_{ij} \leq X_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n,$$

$$\sum_{i=1}^m X_i \leq k, \quad X_i = 0 \vee 1, \quad i = 1, \dots, m,$$

где  $k$  ( $k \leq m$ ) — заданная константа. Кроме ограничения на число

выбираемых в оптимальном решении мест размещения, она отличается от ранее рассмотренной знаком критерия: в ней требуется максимизировать, а не минимизировать целевую функцию. В указанной работе для рассматриваемой задачи предложен и обоснован градиентный алгоритм, строящий приближенное решение, относительная погрешность которого не превосходит  $e^{-1} \approx 0,37$ .

Вторая группа алгоритмов строит последовательность решений или псевдорешений комбинаторной задачи, каждое из которых получается из предшествующего путем перестройки последнего с помощью некоторой процедуры обмена. Примером такой процедуры является альтернирующая последовательность или цепь. Четные элементы такой цепи соответствуют некоторым ненулевым элементам текущего решения, а нечетные — последующего решения. Примерами алгоритмов такого типа являются алгоритмы построения паросочетаний в графах [121, 123], независимого множества максимального веса в матрицах [125, 128] и их обобщений [127, 157].

Наконец, третья группа алгоритмов устроена совершенно иным образом и основана на методе эллипсоидов применительно к комбинаторным задачам (см. [141] и § 2.6). Алгоритмы такого типа наиболее универсальны. Они, как правило, решают за полиномиальное время все задачи, которые могут быть решены алгоритмами предшествующих классов. Однако их теоретическая трудоемкость в последнем случае оказывается большей.

Указанное деление на классы в некоторой степени условно, так как существуют алгоритмы, сочетающие в себе элементы из разных классов, а также алгоритмы, которые обладают рядом свойств, не соответствующих ни одному из отмеченных выше.

В данной главе анализируется ряд моделей, связанных в основном с частными задачами синтеза сетей, для точного и приближенного решения которых можно использовать полиномиально ограниченные алгоритмы. Исследование таких частных случаев позволяет выделить те характерные особенности, наличие которых переводит задачу из класса трудноразрешимых в класс полиномиально разрешимых задач.

В этом разделе мы рассмотрим модификацию алгоритма, предложенного Эдмондсом [122] для решения задач об оптимальном ветвлении на ориентированных графах. Задача заключается в отыскании на ориентированном графе с неотрицательными весами дуг ориентированного дерева минимального суммарного веса с корнем в заданной (или произвольной) вершине. Искомое дерево должно обладать такой ориентацией дуг, чтобы в нем существовали ориентированные пути из корня в любую вершину графа. Указанная задача является частным случаем задачи синтеза сетей с одним источником (корнем), в которой функции затрат на дугах постоянны при любом положительном потоке и равны нулю при нулевом потоке по ним, а каждой вершине, отличной от корня, соответствует положительная потребность.

Опишем модифицированный алгоритм Эдмондса для задачи о ветвлении минимального веса, трудоемкость которого есть  $O(n^2)$ , где  $n$  — число вершин графа. В процессе работы алгоритма каждая вершина графа получает ряд пометок различного типа. Вначале все вер-

шины графа, кроме корня, считаются непомеченными. Корень помечен корневой пометкой, текущее значение функционала равно нулю. Информация о графе представлена в форме матрицы  $C_s = (c_{ij})$  ( $i, j = 1, \dots, n$ ), где  $c_{ij} \geq 0$  — вес дуги  $(i, j)$ . Если  $(i, j) \notin E$ , то  $c_{ij}$  полагают равным достаточно большому числу  $M$ . В процессе работы к матрице  $C$  будут добавляться новые столбцы (числом не более  $n$ ) и будет формироваться новая матрица  $N = (n_{ij})$  (вначале пустая) с числом столбцов, не большим  $n$ , и числом строк, равным  $n$ . Столбцы  $N$  и столбцы, добавляемые к  $C$ , находятся во взаимно однозначном соответствии. Они соответствуют некоторым подмножествам вершин исходного графа  $G$ . Их элементы  $c_{ij}$  и  $n_{ij}$  являются соответственно весами некоторых дуг  $(i, k)$ ,  $i \notin V(j)$ ,  $k \in V(j)$ , и номерами вершин исходного графа, соответствующих концам этих дуг. Кроме того, будет сформирован список  $S$  дуг  $(i, j)$ , из которых будет построено оптимальное ветвление  $E(T)$  (вначале  $S = \emptyset$ ).

Алгоритм начинает работу с произвольной, не помеченной корневой пометкой вершины  $j$ . Для текущей вершины  $j$  (в матрице  $C$  ей соответствует столбец  $j$ ) определяется дуга минимального веса  $(i_0, j)$ , входящая в нее, т. е.  $c_{i_0j} = \min_{1 \leq i \leq n} c_{ij}$ . Вершина  $j$  получает пометку «просмотрена» и пометку-потенциал  $v_j$ , равную  $c_{i_0j}$ , значение функционала  $\Phi$  увеличивается на  $c_{i_0j}$ , а дуга  $(i_0, n(j))$  включается в список  $S$ . Здесь  $n(j) = j$ , если  $j$  — индекс исходной вершины  $G$ , и  $n(j) = n_{i_0j}$ , если  $j$  — новая вершина  $G$ , построенная в процессе работы алгоритма. Кроме того, запоминается и само значение  $j$ . Таким образом, список  $S$  состоит из троек элементов  $(i_0, n(j), j)$ . Затем производится анализ вершины  $i_0$ . Если она имеет корневую пометку, то алгоритм переходит к выполнению процедуры ветвления, описанной ниже, после чего возвращается к поиску вершины, которая не имеет пометки «просмотрена».

Если такая вершина найдется, то алгоритм обрабатывает ее так же, как и данную вершину  $j$ . В противном случае алгоритм заканчивает работу с текущим значением  $\Phi$  и ордеревом, построенным процедурой ветвления. Если вершина  $i_0$  не имеет корневой пометки, но имеет пометку «просмотрена», это означает, что дуги из  $S$  образовали новый орцикл, проходящий по дуге  $(i_0, j)$ . В этом случае выполняется процедура стягивания цикла, в результате которой к матрицам  $C$  и  $N$  добавляется новый столбец, что эквивалентно замене всех вершин, входящих в этот цикл, новой псевдовершиной, добавляемой к графу  $G$ . После работы процедуры стягивания цикла, в результате которой вершины, входящие в орцикл, получают циклическую пометку, алгоритм продолжает работу, рассматривая в качестве текущей вершины  $j$  полученную псевдовершину. Наконец, если вершина  $i_0$  не имеет пометки «просмотрена», алгоритм выбирает ее в качестве текущей вершины и проводит с ней все описанные манипуляции.

Опишем процедуру стягивания цикла. Процедура ведет список  $I$  вершин циклов, полученных на обращениях, предшествующих данному. В начале алгоритма список  $I$  пуст. При очередном обращении к



процедуре на ее вход поступает вершина  $i_0$  с меткой «просмотрена». В матрицах  $C$  и  $N$  вводятся новые столбцы, соответствующие циклу  $l$ , который будет построен на данном этапе. Элементы  $c_{il}$  полагаются равными  $M$ . Вершина  $i_0$  объявляется текущей вершиной  $k$ . Затем осуществляется поочередный просмотр весов дуг  $(i, k)$  из столбца  $k$  матрицы  $C$ . Для  $i = 1, \dots, n$  полагаем  $c_{il} = \min\{c_{ik}, c_{ik} - v_k\}$ . Если  $c_{il} = c_{ik} - v_k$  для очередного  $i$ , полагаем  $n_{il} = i$ , в противном случае  $n_{il}$  не изменяется. Если  $c_{il} = 0$  для некоторого  $i$  и вершина  $i$  имеет пометку «просмотрена», то данное значение  $i$  запоминается как следующая за  $k$  вершина цикла. После окончания обработки текущей вершины  $k$  она включается в список формируемого цикла из  $l$ . Затем проверяется, совпадает ли очередная вершина  $i$  с вершиной  $i_0$ . Если  $i \neq i_0$ , то  $i$  полагается новой текущей вершиной и работа процедуры повторяется. Если  $i = i_0$ , то все пометки для новой вершины  $l$  полагаются равными нулю и работа процедуры заканчивается.

Процедура ветвления состоит из следующих операций. На ее входе имеется вершина  $i_0$  с корневой пометкой, а в списке  $S$  последней записана дуга, исходящая из  $i_0$ . Соответствующий элемент списка имеет вид  $(i_0, j_0, k_0)$ . Список  $S$  обрабатывается в порядке, обратном записанному, — с конца. Очередная дуга  $(i_0, j_0)$  из элемента списка  $(i_0, j_0, k_0)$  заносится в  $E(T)$ . Если  $j_0 = k_0$ , то последний элемент из  $S$  удаляется, вершина  $j_0$  получает корневую пометку и, если список не пуст, процедура повторяется с новым последним элементом. Если  $j_0 \neq k_0$ , то  $k_0$  соответствует последнему стянутому циклу, записанному в списке  $l$  в виде  $(i_0, i_1, \dots, i_p)$ .

Если  $j_0 = i_k$  для некоторого  $1 \leq k \leq p$ , то в списке  $S$  имеется единственная тройка  $(i, j_0, j_0)$ , где  $i \neq i_0$  соответствующая дуге цикла  $k_0$ , входящей в вершину  $j_0$  (в этом случае вершина  $j_0$  не принадлежит никакому другому циклу из  $l$ ). Тройка  $(i, j_0, j_0)$  удаляется из  $S$ , вершина  $j_0$  получает корневую пометку, и процедура повторяется, выбирая в качестве текущего последний элемент списка  $S$ , а в качестве  $i_0$  — первый индекс из выбранной тройки. Если  $j_0 \neq i_k$  для всех  $k = 1, \dots, p$ , то просматриваются все циклы, содержащие вершину  $j_0$ . Это легко сделать, найдя сначала первый цикл  $j_1$ , содержащий собственно вершину  $j_0$ , а затем последовательно находя циклы  $j_{s+1}$ , в которые стянута предшествующая вершина  $j_s$  ( $s = 1, \dots, r$ ). Для полученной таким образом последовательности  $j_1, \dots, j_r$  из  $S$  поочередно удаляется единственная тройка  $(i, j_s, j_{s-1})$ , где  $s = 1, \dots, r$ . После этого дуга  $(i, j_0)$  заносится в  $E(T)$ , вершина  $j_0$  получает корневую пометку и процедура повторяется, если список  $S$  еще не пуст.

На рис. 6.1 схематически показан пример работы алгоритма. На нем изображены лишь дуги, соответствующие элементам из  $S$ . Работа алгоритма определяется следующей последовательностью просматриваемых и порожденных вершин и циклов (номера порожденных циклов приведены в скобках): 1, 2, 3, 4, (7), 5, 6, (8), (9), (10). Список  $S$ , построенный в процессе работы, имеет вид (2, 1, 1), (3, 2, 2), (4, 3, 3), (2, 4, 4), (5, 3, 7), (6, 5, 5), (5, 6, 6), (3, 6, 8), (1, 4, 9), (0, 5, 10), а список циклов  $l$  будет содержать следующие элементы: (2, 3, 4), (5, 6), (7, 8), (1, 9).

Процедура ветвления работает на списках  $S$  и  $I$  следующим образом. Вначале в список  $E(T)$  заносится дуга  $(0, 5)$ . Поскольку  $k_0 = 10$ ,  $j_0 = 5$ ,  $k_0 \neq j_0$ , то анализируется цикл  $(1, 9)$ . В нем нет вершины 5, поэтому процедура определяет последовательность циклов  $(5, 6)$ ,  $(7, 8)$ ,  $(1, 9)$  (номера этих циклов соответственно 8, 9, 10) и удаляет из  $S$  дуги  $(6, 5, 5)$ ,  $(3, 6, 8)$ ,  $(1, 4, 9)$ , после чего удаляются из  $I$  и выписанные циклы. Текущим последним элементом в  $S$  становится  $(5, 6, 6)$ . Для него  $k_0 = 6 = j_0$ , поэтому дуга  $(5, 6)$  заносится в  $E(T)$ , а  $(5, 6, 6)$  удаляется из  $S$ . После анализа следующего элемента  $(5, 3, 7)$  из списка  $S$  удаляется  $(4, 3, 3)$ , из списка  $I$  —  $(2, 3, 4)$ , а в  $E(T)$  заносится дуга  $(5, 3)$ . Поскольку список  $I$  уже пуст, на следующих шагах поочередно удаляются из  $S$  оставшиеся элементы  $(2, 4, 4)$ ,  $(3, 2, 2)$  и  $(2, 1, 1)$ , а в  $E(T)$  соответственно заносятся дуги  $(2, 4)$ ,  $(3, 2)$ ,  $(2, 1)$ . В процессе работы вершины 1—6 получают корневую пометку.

Приведем обоснование изложенного алгоритма. Рассматриваемая задача может быть сформулирована в виде задачи целочисленного линейного программирования: найти

$$\min \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (6.1)$$

при ограничениях

$$\sum_{(i,j) \in R_k} x_{ij} \geq 1, \quad k \in K, \quad (6.2)$$

$$x_{ij} \geq 0, \quad (i, j) \in E, \quad (6.3)$$

$$x_{ij} = 0 \vee 1, \quad (i, j) \in E, \quad (6.4)$$

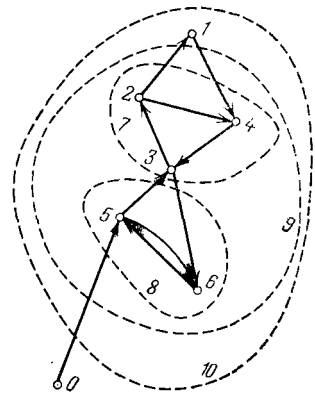


Рис. 6.1

где  $R_k$  — ориентированный разрез, определяемый произвольным подмножеством вершин  $V_k$ ,  $V_k \subset V$ ,  $0 \notin V_k$ , а  $K$  — множество всех таких разрезов. Изложенный алгоритм, в сущности, решает задачу линейного программирования, двойственную к (6.1)—(6.4). Он просматривает некоторую последовательность ориентированных разрезов из  $K$  и поочередно присваивает двойственной переменной, соответствующей текущему разрезу, максимально допустимое значение, равное минимуму из преобразованных на предшествующих шагах весов дуг, образующих этот разрез. Процедура ветвления строит по решению двойственной задачи (вообще говоря, частичному) решение прямой задачи (6.1)—(6.4). Когда процесс решения заканчивается, список дуг  $E(T)$  находится во взаимно однозначном соответствии с множеством единичных переменных из решения задачи (6.1)—(6.4). Этот факт эквивалентен следующему утверждению.

**Л е м м а 6. 1.** *Множество дуг из  $E(T)$  образует ветвление.*

**Д о к а з а т е л ь с т в о.** Множество дуг, по которым строится список  $S$ , содержит непосредственно перед обращением к процедуре ветвления орпути из вершины с корневой пометкой во все вершины из  $V$ , которые обрабатывались алгоритмом на текущем этапе. Проце-

дура ветвления удаляет ровно по одной дуге из каждого построенного орцикла, причем удаляется дуга, ведущая в такую вершину, для которой на последующих за образованием орцикла шагах строится орпуть из вершины с корневой пометкой. Следовательно, удаление таких дуг оставляет для каждой вершины цикла хотя бы один искомый орпуть. С другой стороны, в список  $E(T)$  попадает  $|S| - |I|$  дуг, входящих по одной в каждую вершину из  $V$ , обрабатываемую алгоритмом. Следовательно, множество включаемых на текущем этапе дуг  $E(T)$  образуют вместе с ранее включенными в  $E(T)$  дугами ордереву с корнем в вершине 0, т. е. ветвление.

Остается показать, что вес ветвления, получаемого после каждого обращения к процедуре ветвления, равен текущему значению  $\Phi$ . Отсюда и из первой теоремы двойственности будет следовать оптимальность текущего ветвления.

*Лемма 6.2. Вес ветвления, определяемого  $E(T)$ , равен  $\Phi$ .*

*Доказательство.* По построению  $\Phi$  равно сумме меткопотенциалов, присвоенных всем вершинам (исходным и новым, построенным стягиванием циклов). При построении новых вершин величина метки-потенциала вычитается из весов всех дуг, входящих в вершину цикла. В множестве  $E(T)$  в любой заданный цикл и в любую исходную вершину входит ровно одна дуга, начало которой не принадлежит этому же циклу. Рассмотрим любую дугу  $(i_0, j_0)$  из  $E(T)$ . Пусть вершина  $j_0$  входит в циклы с номерами  $j_1, j_2, \dots, j_r, r \geq 0$ . В этот момент, когда алгоритм просмотрит вершину  $j_r$ , он присвоит ей потенциал

$$v_{j_r} = c_{ij} - \sum_{s=0}^{r-1} v_{j_s},$$

поскольку при построении вершины с номером  $j_s, s = 1, \dots, r$ , вес дуги уменьшается на  $v_{j_s}$ . Отсюда

$$c_{i_0 j_0} = \sum_{s=0}^r v_{j_s}, \quad (i, j_0) \in E(T).$$

Суммируя эти равенства по  $(i, j_0) \in E(T)$ , получим требуемое:

$$\sum_{(i, j) \in E(T)} c_{ij} = \sum_{i=1}^{|S|} v_{j_i} = \Phi.$$

Трудоёмкость алгоритма оценивается следующим образом. Число циклов, построенных алгоритмом, не превышает  $n - 1$ . Поэтому работа по просмотру всех вершин имеет трудоёмкость не выше  $(2n - 1) O(n)$ . Трудоёмкость работы процедуры стягивания цикла с  $k$  вершинами оценивается как  $O(kn)$ . Однако общая ее трудоёмкость оценивается как  $O(n^2)$ , потому что каждая вершина  $i, 1 \leq i \leq 2n - 1$ , обрабатывается процедурой не больше одного раза. С учетом последнего замечания ясно, что общая трудоёмкость процедуры ветвления также оценивается как  $O(n^2)$ . Отсюда трудоёмкость алгоритма —  $O(n^2)$ .

## § 2. Простейшая задача размещения на древовидных сетях

Простейшая задача размещения, рассмотренная в § 1 гл. 5 в матричной форме, допускает и естественную теоретико-графовую интерпретацию, в которой заданный граф  $G(V, R)$  с множеством вершин  $V$  и ребер  $R$  с заданными весами  $d_r > 0, r \in R$  представляет множество коммуникаций, по которым транспортируется продукт от мест производства к потребителям. Предполагается, что как потребители, так и места возможного размещения производства расположены в вершинах этого графа. В каждой вершине  $i \in V$  может размещаться и производство, и потребитель. Коэффициенты  $c_{ij}$  матрицы  $C$  в этом случае вычисляются путем нахождения длины кратчайшего пути от поставщика  $i$  к потребителю  $j$  в графе с весами ребер  $d_r$ . Каждой вершине  $i \in V$  ставится в соответствие объем потребления  $B_i$  и кусочно-линейная вогнутая функция  $f_i(X_i)$  затрат на размещение производства с объемом  $X_i$ . При этом если в данной вершине нет потребителя, то  $B_i = 0$ , а если в ней нельзя размещать производство, то  $f_i(X_i) = MX_i$ , где  $M$  — достаточно большое положительное число. Преобразование задачи, осуществляющее переход от произвольных функций  $f_i(X_i)$  к постоянным функциям с фиксированными доплатами  $g_{is}(X_{is})$ ,

$$g_{is}(X_{is}) = \begin{cases} b_{is} & \text{при } X_{is} > 0, \\ 0 & \text{при } X_{is} = 0, \end{cases} \quad i \in V, \quad s = 1, \dots, S,$$

достигается путем добавления к графу  $G$  для каждой вершины  $i \in V$  новых вершин, каждая из которых соединена ребром, исходящим из нее, только с вершиной-прообразом  $i$ . Длина этого ребра равна коэффициенту  $a_{is}$  при переменной  $X_i$  на  $s$ -м участке линейности функции  $f_i(X_i)$ . Старая вершина  $i$  считается после этого точкой, где размещение производства невозможно. Новым вершинам соответствует объем потребления, равный нулю.

В сетевой постановке формулируются многие прикладные задачи. Мы рассматриваем эту постановку для выявления подклассов задач, для которых можно построить эффективные алгоритмы. Характерным с этой точки зрения параметром задачи (точнее, соответствующего графа  $G$ ) является цикломатическое число  $\nu(G)$  графа  $G$ . Напомним, что для связного графа (мы будем рассматривать только такие)  $\nu(G) = |R| - |V| + 1$ . Если  $\nu(G) = 0$ , то  $G$  — дерево, обозначаемое в дальнейшем  $T(V, R)$ . В данном разделе изучается подкласс простейших задач размещения на древовидных графах. Доказывается свойство локальной унимодулярности, из чего непосредственно следует полиномиальность их разрешимости. Однако для таких задач имеет место более сильное свойство: они могут быть решены градиентным алгоритмом, имеющим меньшую вычислительную трудоемкость, чем общие методы линейного программирования. Приводится обоснование этого алгоритма и его вычислительная схема. Кроме указанного, анализируется еще один подкласс задач размещения на древовидных графах, отличающийся

лишь способом вычисления коэффициентов  $c_{ij}$  в матрице  $C$ , для которого получены аналогичные результаты.

Рассмотрим подкласс задач с  $\nu(G) \leq \nu$ , где  $\nu$  — константа. Число различных деревьев в графе  $G$  в этом случае не превосходит  $|V|^\nu$ , и исходная задача может быть решена за время  $O(|V|^\nu P)$ , где  $P$  — трудоемкость решения задачи размещения на древовидном графе. Эту оценку можно получить простым перебором всех деревьев в графе  $G$  с последующим применением описанного ниже алгоритма.

Перейдем теперь к рассмотрению свойств простейшей задачи размещения на древовидном графе  $T(V, R)$ . Мы будем предполагать, что в каждой вершине  $i$  заданы величины  $B_i$  — объем потребления и  $b_i$  — фиксированные доплаты ( $i = 1, \dots, n$ ,  $n = |V|$ ). Из описанного выше правила перехода от произвольных функций к постоянным с фиксированными доплатами  $b_{i_s}$  легко понять, что преобразованный граф также остается деревом. Поэтому будем считать, что рассматриваемое дерево  $T(V, R)$  построено в результате указанного преобразования. Как показано в § 5.1, преобразованная двойственная к релаксированной задаче размещения имеет вид: найти

$$\max \left\{ \sum_{j=1}^n \sum_{k=1}^m \omega_{jk} : \sum_{j=1}^n \sum_{k=1}^m a_{ijk} \omega_{jk} \leq b_i, \quad i = 1, \dots, n, \right. \\ \left. 0 \leq \omega_{jk} \leq f_{jk}, \quad j = 1, \dots, n, \quad k = 1, \dots, m \right\}. \quad (6.5)$$

Здесь  $k$  — порядковый номер элемента  $c_{ij}$  в упорядоченном по неубыванию списке элементов  $j$ -го столбца матрицы удельных транспортных затрат  $C = (c_{ij})$ ,

$$f_{jk} = c_{jk+1} - c_{jk}, \quad f_{jn} = \infty, \quad j = 1, \dots, n, \quad k = 1, \dots, m-1,$$

$$a_{ijk} = \begin{cases} 1, & \text{если порядковый номер } i \text{ не больше } k, \\ 0 & \text{в противном случае.} \end{cases}$$

Выписывая задачу, двойственную к (6.5), получаем эквивалентное представление для исходной задачи: найти

$$\min \left\{ \sum_{i=1}^n \sum_{j=1}^m f_{ji} x_{ij} + \sum_{i=1}^n b_i X_i : \sum_{i=1}^n a_{ijk} X_i + x_{kj} \geq 1, \right. \\ \left. k = 1, \dots, m, \quad j = 1, \dots, n, \quad X_i = 0 \vee 1, \quad i = 1, \dots, n, \right. \\ \left. x_{kj} \geq 0, \quad k = 1, \dots, m, \quad j = 1, \dots, n \right\}. \quad (6.6)$$

Пусть  $T(V, R)$  — дерево с множеством вершин  $V$  и ребер  $R$ ,  $|V| = n$ , длина ребра  $r \in R$  равна  $d_r > 0$ . Задача (6.6)—(6.9) на множестве вершин  $V$  дерева  $T$ , в которой элементы

матрицы  $C = (c_{ij})$  вычисляются по формуле

$$c_{ij} = B_j \left( \sum_{r \in P_{ij}} d_r \right), \quad i, j = 1, \dots, n.$$

Здесь  $P_{ij}$  — множество ребер цепи, соединяющей вершины  $i, j$  в  $T$ .

Рассмотрим модель (6.5), соответствующую задаче 1. Тогда каждому вектор-столбцу  $a_{jh} \in A$  соответствует поддерево  $T_{jh}$  из  $T$ , определяемое по следующему правилу:

$$a_{jh} = 1 \Leftrightarrow i \in V(T_{jh}), \quad (6.7)$$

где  $V(T_{jh})$  — множество вершин  $T_{jh}$ . Таким образом, каждой вершине  $i \in V$  ставится в соответствие  $n$  поддеревьев, причем  $k$ -е поддерево включает в себя  $k$  вершин, ближайших по расстоянию к вершине  $i$  (включая и саму  $i$ ). Основное свойство, позволяющее строить эффективный алгоритм решения задачи 1, состоит в следующем.

**Теорема 6.1** Пусть  $\bar{A} \subset A$  — произвольная подматрица с множеством столбцов  $J(\bar{A})$ , множеством строк  $I(\bar{A})$ . Тогда в  $\bar{A}$  найдется либо строка  $i$  с  $\sum_{(jh) \in J(\bar{A})} a_{ijh} \leq 1$ , либо столбцы  $l_1, l_2$  такие,

что  $a_{il_1} \leq a_{il_2}$ ,  $i \in I(\bar{A})$ .

**Доказательство.** Для упрощения доказательства предположим, что длины ребер в  $T$  таковы, что любые две произвольные цепи имеют различные длины. Этого легко добиться с помощью достаточно малых добавок к  $d_r$ ,  $r \in R$ , не меняющих оптимального решения задачи. Цепь, соединяющую пару вершин  $i, j$  в  $T$ , будем обозначать  $[i, j]$ , а ее длину —  $d(i, j)$ . Через  $T(i, j)$  обозначим поддерево, получаемое из  $T$  удалением всех ребер, инцидентных  $i$ , кроме ребра из цепи  $[i, j]$ , и последующим удалением всех компонент полученного леса, кроме содержащей цепь  $[i, j]$ . Доказательство ведется в терминах поддеревьев, порожденных отображением (6.7).

Пусть для  $\bar{A}$  не выполняется первый вариант альтернативы. Покажем, что тогда верен второй вариант. Пусть  $[v, w]$  — цепь максимальной длины в  $T$  (диаметр  $T$ ) и вершина  $w$  принадлежит по крайней мере двум поддеревам  $T_1, T_2 \in J(\bar{A})$ , где  $T_1$  порождено вершиной  $u_1$ , а  $T_2$  — вершиной  $u_2$ . Если для  $T_1, T_2$  не верен второй вариант альтернативы, найдутся вершины  $t_1 \in T_1, t_2 \in T_2$  такие, что  $t_1 \notin T_2, t_2 \notin T_1$ . Пусть  $s$  — ближайшая к  $v$  вершина, принадлежащая цепям  $[v, w], [t_1, w], [t_2, w]$ . Для определенности пусть  $[t_2, s]$  не имеет с  $[v, w]$  общих точек, кроме  $s$  (см. рис. 6.2). Тогда по выбору  $[v, w]$

$$d(s, w) > d(s, t_2), \quad (6.8)$$

так как в противном случае  $d(v, t_2) > d(v, w)$ . Тогда  $u_1 \in T(s, w)$ , поскольку  $w \in T_1, t_2 \notin T_1$ . Отсюда следует, что

$$d(s, t_2) > d(s, t_1), \quad (6.9)$$

поскольку  $t_1 \in T_1$ . Следствием (6.8), (6.9) является

$$d(s, \omega) > d(s, t_1). \quad (6.10)$$

Но тогда либо из  $u_2 \notin T(s, \omega)$  и (6.10), либо из  $u_2 \in T(s, \omega)$  и (6.9) следует, что  $t_1 \in T_2$ . Значит, для  $T_1$  и  $T_2$  должен выполняться второй вариант альтернативы.

Квадратная  $(0,1)$ -матрица размеров  $n \times n$  ( $n \geq 3$ ) называется циклом, если она не блочная и содержит в каждой строке и в каждом столбце ровно два единичных элемента.  $(0,1)$ -матрица  $A$  называется сбалансированной [108], если любая вершина многогранника  $(AX \leq e, X \geq 0)$  целочисленна, где  $e$  — произвольный  $(0,1)$ -вектор. М. Падберг показал, что сбалансированными являются те и только те матрицы, которые не содержат циклов нечетной длины, т. е. циклов с нечетным ( $n \geq 3$ ) числом строк и столбцов. В многограннике, задаваемом системой неравенств  $AX \leq e, X \geq 0$ , со сбалансированной матрицей  $A$ , каждой вершине соответствует базисная вполне унимодулярная подматрица из  $A$ . Если вершина  $X^0$  этого многогранника оптимальна для заданного линейного функционала  $bX$ , то соответствующий ей базис порождает и оптимальное решение двойственной задачи линейного программирования, которое целочисленно при целочисленном  $b$ . С учетом этих замечаний из приведенной теоремы и ее доказательства вытекают следующие свойства матрицы  $A$  и оптимального решения задачи 1.

1. Матрица  $A$  не содержит циклов.
2. Матрица  $A$  сбалансирована.
3. Среди оптимальных решений задачи 1 имеются целочисленные.

Рассмотрим теперь другое правило порождения поддеревьев  $T_{jk} \subset T$  ( $j, k = 1, \dots, n$ ). Каждое  $T_{jk}$  однозначно определяется вершиной  $j$  и множеством его ребер следующим образом:

- а)  $R_{j1} = \emptyset$ ;
- б)  $R_{jk} = R_{j(k-1)} \cup r_{i(k)}$ ,  $k = 2, \dots, n$ ,

где  $r_{i(k)}$  — ребро минимальной длины, смежное с  $T_{j(k-1)}$ ;  $i(k)$  — индекс вершины, инцидентной  $r_{i(k)}$ ,  $i(k) \notin T_{j(k-1)}$ . Способ построения  $T_{jk}$  упорядочивает для каждого  $j = 1, \dots, n$  пары  $(k, j)$  по включению поддеревьев  $T_{jk}$ , порожденных  $j$ .

Задача 2 отличается от предыдущей лишь способом вычисления коэффициентов матрицы  $C = (c_{ij})$ :

$$c_{i(1)j} = 0, \quad j = 1, \dots, n,$$

$$c_{i(k)j} = B_j \sum_{r \in R_{jk}} d_r, \quad k = 2, \dots, n,$$

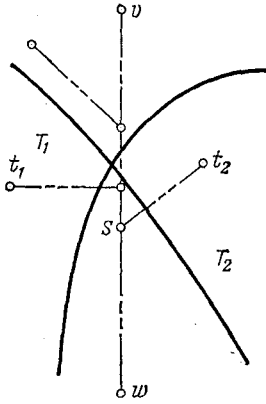


Рис. 6.2

Как и для предыдущей, для задачи 2 имеет место теорема 6.1 и ее следствия. Столбцы матрицы  $A$  и здесь порождаются правилами порождения поддеревьев  $T_{jk}$  и отображением (6.7).

**Доказательство теоремы 6.1 для задачи 2.** Обозначим через  $T(\bar{A})$  минимальное поддерево с множеством вершин, содержащим  $I(\bar{A})$ . Если в  $T(\bar{A})$  имеется ребро, не покрываемое поддеревьями из  $J(\bar{A})$ , это значит, что матрица  $\bar{A}$  блочная. В этом случае доказательство проведем для произвольного блока из  $\bar{A}$ , который будем по-прежнему обозначать  $\bar{A}$ . Рассмотрим в  $T(\bar{A})$  ребро  $r$  максимальной длины. Удаление  $r = (i, j)$  разбивает  $T(\bar{A})$  на два непересекающихся поддерева  $T_1(\bar{A}), T_2(\bar{A}), i \in T_1(\bar{A}), j \in T_2(\bar{A})$ . Пусть  $[v, \omega]$  — цепь минимальной длины, содержащая  $i, j$  и удовлетворяющая условиям  $v \in T_1(\bar{A}), \omega \in T_2(\bar{A}), v, \omega \in I(\bar{A})$ . Пусть вершина  $v$  принадлежит двум деревьям  $T_3, T_4 \in J(\bar{A})$  (в противном случае выполняется первый вариант альтернативы). Тогда, с точностью до изменения индексов, вершина  $t$ , порождающая  $T_3$ , принадлежит  $T_1(\bar{A})$ , а вершина  $u$ , порождающая  $T_4, T_2(\bar{A})$ , так как в противном случае либо  $\bar{A}$  блочная, либо справедлив второй вариант альтернативы (в зависимости от принадлежности или непринадлежности цепи  $[v, j]$  деревьям  $T_3$  и  $T_4$ ). Но и в этом случае, если существует дерево  $T_5$ , порожденное вершиной  $s, s \neq t$ , из  $T_2(\bar{A})$ , то для  $T_4, T_5$  выполнены условия теоремы (так как  $V(T_4) \supset V(T_2(\bar{A}))$ ). Иначе  $V(T_2(\bar{A})) = \{u\} = V(t_4)$ . В последнем случае найдется поддерево  $T_6$ , порожденное вершиной из  $T_1(\bar{A})$ , содержащее  $u$  и, следовательно,  $r$ . Поэтому  $V(T_1(\bar{A})) \subset V(T_6)$ , т. е.  $T_4$  и  $T_6$  сравнимы по входящим в них вершинам.

**З а м е ч а н и е.** Теорема 6.1 остается справедливой для обеих задач, если каждое ребро заменить парой противоположных дуг с различными неотрицательными длинами. Однако здесь мы не приводим доказательства этого утверждения для задачи 2. Для задачи 1 приведенное доказательство требует в этом случае лишь незначительных изменений.

Для обоснования алгоритма решения задач 1, 2 нам потребуются следующие результаты. Пусть  $W^*$  — тупиковое решение задачи (6.5), т. е. решение, полученное поочередным присвоением переменным их максимально допустимых значений. Этот процесс продолжается до тех пор, пока можно увеличивать значение какой-либо из переменных. Обозначим

$$\begin{aligned} N(W^*) &= \left\{ i : \sum_{j,k=1}^n a_{ijk} \omega_{kj}^* = b_i \right\}, \\ P(W^*) &= \{(k, j) : \omega_{kj}^* < f_{kj}\}, \\ Q(W^*) &= \{(k, j) : \omega_{kj}^* > 0\}. \end{aligned} \quad (6.11)$$

Назовем решение  $W^*$  правильным, если не найдется строк  $i_1, i_2 \in N(W^*)$  и столбцов  $r_1 \in Q(W^*), r_2, r_3 \in P(W^*), r_s = (k_s, j_s), s =$



$= 1, 2, 3$  таких, что

$$\begin{aligned} a_{i_1 r_1} &= a_{i_2 r_1} = 1, \\ a_{i_s r_2} + a_{i_s r_3} &= 1, \quad s = 1, 2, \\ a_{i_1 r_s} + a_{i_2 r_s} &= 1, \quad s = 2, 3. \end{aligned} \quad (6.12)$$

**Теорема 6.2.** *Правильное тупиковое решение оптимально. Для доказательства нам потребуются следующие вспомогательные утверждения.*

**Лемма 6.3.** *Пусть  $A = (a_{ij})$  ( $i, j = 1, \dots, n$ ) — квадратная, левая треугольная и невырожденная матрица, не содержащая подматриц  $\bar{A}$  размером  $2 \times 3$ , удовлетворяющих условиям (6.12). Тогда система*

$$XA = I, \quad (6.13)$$

где  $I$  — мерный вектор из единиц, имеет неотрицательное решение.

**Доказательство.** Пусть  $X = (x_1, \dots, x_n)$  — решение системы (6.13). Очевидно, все  $x_i$  в нем целочисленны. Предположим, что  $x_k < 0$  и  $k$  — наибольшее значение индекса, удовлетворяющее этому свойству. Обозначим  $E(k) = \{i > k : a_{ik} = 1 \wedge x_i = 1\}$ . Тогда  $k$ -е уравнение из (6.13) можно записать в виде  $\sum_{i \in E(k)} x_i + x_k = 1$ . Для него из

$x_k < 0$  следует  $|E(k)| \geq 2$ . Пусть  $j, l \in E(k)$ ,  $j < l$ ,  $l$  — максимальное значение индекса из  $E(k)$ ,  $j$  — следующее за ним. Из выбора  $j, l$  следует, что  $a_{jj} + a_{lj} = 1$ , т. е.  $a_{lj} = 0$ , а из треугольности  $A$  — что  $a_{jl} = 0$ . Поэтому подматрица  $\bar{A} \subseteq A$  со строками  $j, l$  и столбцами  $k, j, l$  удовлетворяет (6.12). Противоречие.

Пусть  $X$  — решение системы (6.13), в которой  $A$  удовлетворяет условиям леммы 6.3,  $b$  — произвольный ненулевой  $n$ -мерный  $(0,1)$ -вектор,  $B$  — матрица, полученная из  $A$  путем добавления к последней вектор-столбца  $b$ , т. е.  $B = (A|b)$ .

**Лемма 6.4.** *Если  $B$  не содержит подматриц  $\bar{A}$  размерности  $2 \times 3$ , удовлетворяющих условиям (6.12), то  $Xb \geq 1$ .*

**Доказательство.** Предположим, что  $Xb < 1$ . По лемме 6.3  $X$  —  $(0,1)$ -вектор, следовательно,  $Xb = 0$ . Пусть  $k = \max_{1 \leq j \leq n} \{j : b_j = 1\}$ ,

тогда  $x_k = 0$ . Поскольку  $Xa^k = 1$ , где  $a^k$  —  $k$ -й вектор-столбец матрицы  $A$ , то найдется строка  $l$ ,  $l > k$ , такая, что  $a_l^k = 1$ ,  $x_l = 1$ . По выбору  $k$   $b_l = 0$ . По выбору  $A$   $a_h^k = a_l^k = 1$ ,  $a_h^k = 0$ . Следовательно, подматрица  $\bar{A}$  со столбцами  $a^k, a^l, b$  и строками с номерами  $k, l$  удовлетворяет условиям (6.12), что противоречит предположениям леммы.

**Доказательство теоремы 6.2.** Пусть  $W^*$  — правильное тупиковое решение. Для простоты рассуждений будем предполагать, что  $|P(W^*) \cap Q(W^*)| = |N(W^*)|$ . Выполнение этого предположения легко достигается с помощью стандартного в линейном программировании  $\varepsilon$ -преобразования всех величин  $b_i$  и  $f_{kj}$ , превращающего исходную задачу в невырожденную. Для доказательства

оптимальности  $W^*$  построим допустимое решение  $X^*$  задачи, двойственной к (6.5), удовлетворяющее условиям дополняющей нежесткости. Из теории двойственности в линейном программировании будет следовать оптимальность  $W^*$ . Условия дополняющей нежесткости для пары решений прямой задачи и двойственной можно сформулировать в виде:

- а)  $X_i^* = 0$  для  $i \notin N(W^*)$ ;
- б)  $x_{kj}^* = 0$  для  $(k, j) \in P(W^*)$ ;
- в)  $\sum_{i \in N(W^*)} a_{ikj} X_i^* = 1$  для  $(k, j) \in P(W^*) \cap Q(W^*)$ .

Система уравнений в) с точностью до перестановки строк и столбцов удовлетворяет условиям леммы 6.3. Поэтому вектор  $X^*$ , определенный как решение системы а), в), является неотрицательным и целочисленным. По лемме 6.4 ограничения (6.6) для  $(k, j) \in P(W^*)$  выполняются для построенного выше вектора  $X^*$  с учетом б). Остается определить значения  $x_{kj}^*$  для  $(k, j) \notin P(W^*)$ . Положим

$$\text{г) } x_{kj}^* = \max \left\{ 0, 1 - \sum_{i=1}^n a_{ikj} X_i^* \right\}, \quad (k, j) \notin P(W^*).$$

Таким образом, решение  $X^*$  является допустимым, что влечет оптимальность  $W^*$ .

Для задач 1, 2 можно так упорядочить переменные в (6.5), что тупиковое решение, полученное путем поочередного присвоения максимально допустимых значений переменных в указанном порядке, будет правильным и, следовательно, оптимальным. Изложение общего для задач 1, 2 алгоритма проведем в терминах соответствующих поддеревьев. Введем необходимые определения. Выберем в  $T$  произвольную вершину, назовем ее корнем и пометим индексом 0 (ее ранг равен нулю). Ранги остальных вершин  $i$  определим как число ребер в пути из корня в эти вершины; они принимают значения  $\rho(i) = 1, 2, \dots, R$ . Вершины, в которые существует путь из данной вершины  $i$ , обладающий свойством, что вдоль него ранг возрастает, будем называть потомками  $i$ . Для корня все вершины — потомки. Поддеревья, порожденные каждой вершиной  $k$ , определяются вершиной  $i$ , присоединяемой последней. Вершину  $i$  для такого дерева будем называть порождающей, а вершину  $k$  — определяющей. В процессе упорядочения минимальное неупорядоченное дерево, порожденное вершиной  $i$ , будем называть текущим для  $i$ . Перед началом упорядочения текущим для любой вершины  $i$  является поддерево с порождающей и определяющей вершиной  $i$ , т. е. состоящее лишь из  $i$ .

*Алгоритм упорядочения* просматривает вершины дерева  $T$  в порядке невозрастания их рангов. Порядок просмотра вершин с одинаковым рангом производится с учетом их «родства», т. е. вершины-потомки одного родителя любого ранга должны просматриваться подряд. На первом шаге упорядочиваются поддеревья, вершины которых имеют ранг  $R$ , на втором — неупорядоченные ранее поддеревья с вершинами ранга  $R - 1$  и  $R$ , на последнем — неупорядоченные ранее поддеревья с вершинами любого ранга. Таким образом, на  $r$ -м шаге

процесс состоит из упорядочения ранее неупорядоченных поддеревьев, порожденных одной из вершин ранга  $R - r + 1$  и ее потомками. Опишем процедуру упорядочения для вершины  $i$  с  $\rho(i) = R - r + 1$  на шаге  $r$ .

Вначале упорядочивается дерево  $T_{ik}$  — текущее для  $i$  с определяющей вершиной  $k$  — и для  $i$  строится новое текущее дерево  $T_{il}$ . Затем поочередно упорядочиваются все текущие поддеревья с той же определяющей вершиной  $k$ , порожденные вершинами  $j$  в порядке возрастания их рангов и с учетом их «родства» (см. выше). При этом после упорядочения текущего дерева  $T_{js}$  строится новое текущее дерево  $T_{ju}$  для вершины  $j$ . Если  $u \neq t$ , где  $t$  — определяющая вершина  $T_{pt}$ ,  $p$  — «родитель»  $j$  (т. е.  $\rho(p) = \rho(j) - 1$  и  $j, p$  смежны в  $T$ ), то после просмотра всех потомков  $i$  процедура повторяется для всех вершин  $j$  — потомков  $i$  максимального ранга, обладающих указанным свойством ( $u \neq t$ ). Процедура повторяется до тех пор, пока для очередного  $T_{il}$  вершина  $l$  не станет вершиной ранга  $r - 1$ , либо не будут упорядочены все поддеревья.

Для примера ниже приведен список поддеревьев, упорядоченных с помощью описанного алгоритма. Порядок вершин в элементах списка соответствует порядку их присоединения (первая вершина — порождающая, последняя — определяющая). Дерево  $T$  изображено на рис. 6.3.

Задача 1.  $a, b, c, ac, bc, ca, bca, d, e, cae, ace, bcae, caeb, aceb, de, ec, dec, ecd, caebd, acebd, bcaed, deca, ecda, ecdab, decab$ .

Задача 2.  $a, b, c, ac, bc, ca, bca, d, e, cae, ace, bcae, de, ec, dec, eca, deca, ecad, caed, aced, bcaed, caedb, acedb, ecadb, decab$ .

Рассмотрим матрицу  $\tilde{A}$ , которая получается путем перестановки столбцов матрицы  $A$  в соответствии с описанным алгоритмом упорядочения. Пусть  $r_1 < r_2 < r_3$  — произвольные индексы ее столбцов, а  $i_1, i_2$  — строк. Легко понять, что коэффициенты подматрицы  $\bar{A} \subset \tilde{A}$ , определенной выбранными строками и столбцами, не удовлетворяют условиям (6.11). Это свойство является простым следствием предложенного алгоритма. Поэтому, если одновременно с процессом упорядочения поддеревьев (столбцов  $A$ ) определять для соответствующей текущему поддереву  $T_{js}$  переменной  $w_{js}$  ее максимально допустимое значение  $\bar{w}_{js} = \min_{i \in V(T_{js})} \bar{b}_i$  и полагать новые значения  $\bar{b}_i$  равными  $\bar{b}_i - \bar{w}_{js}$  для  $i \in V(T_{js})$ , не изменяя остальных  $\bar{b}_i$  (в начале работы  $\bar{b}_i = b_i, i \in V(T)$ ), то построенное тупиковое решение  $\bar{W}$  будет правильным и, следовательно, оптимальным. Процедура генерации текущего поддерева  $T_{js}$  и определения значения  $\bar{w}_{js}$  требует  $O(n)$  действий. Требуемая при этом память также растет, как  $O(n)$ . Поскольку число просматриваемых деревьев равно  $n^2$ , то общая трудоемкость алгоритма —  $O(n^3)$  при памяти  $O(n)$ .

### § 3. Подкласс задач Штейнера на плоскости с прямоугольной метрикой

Задача Штейнера на графе (ЗШГ) может быть сформулирована следующим образом. Дан оргграф  $G(V, E)$  с множествами вершин  $V$  и дуг  $E$ . Множество  $V$  разбито на три непересекающихся класса:  $V = \{0\} \cup V_1 \cup V_2$ , где  $0$  — корневая вершина в  $G$ ,  $V_1$  — множество основных вершин,  $V_2$  — множество дополнительных вершин. Каждой дуге  $e \in E$  поставлена в соответствие ее длина  $c_i \geq 0$ . Требуется найти в  $G$  подграф минимальной длины (т. е. подграф с минимальной суммой длин дуг), в котором существует орпуть из корня  $0$  в каждую вершину  $v \in V_1$ . Известно [26], что эта задача  $NP$ -трудна даже для таких графов, в которых вершины  $v \in V$  есть точки плоскости с прямоугольной метрикой ( $L_1$ -плоскость) и длины дуг графа  $G$  определяются в соответствии с этой метрикой, т. е.

$$c(X_1, X_2) = c(X_2, X_1) = \rho(X_1, X_2) = |x_1 - x_2| + |y_1 - y_2|, \quad (6.14)$$

где  $X_1 = (x_1, y_1)$ ,  $X_2 = (x_2, y_2)$ . С другой стороны ЗШГ разрешима за полиномиальное от длины входа время для графов с  $|V_2| = \text{const} = m$ . В этом случае задача сводится к решению не более, чем  $2^m$  задач о кратчайшем ветвлении, для которых существует полиномиальный алгоритм.

Здесь рассматривается подкласс ЗШГ на  $L_1$ -плоскости, который определяется следующим образом. Из точки  $0$  проведены четыре луча, совпадающие с лучами координатных прямых и ориентированные в направлении возрастания модулей соответствующих переменных. Через каждую точку из  $V_1$  проведены два луча, параллельные осям координат, начинающиеся в точках этих осей и ориентированные так же, как лучи, исходящие из  $0$  (т. е. в направлении возрастания модулей соответствующих переменных). Для каждого луча найдем наиболее удаленную от  $0$  его точку, принадлежащую также и другому лучу, и заменим рассматриваемый луч отрезком, ограниченным началом луча и выбранной точкой. Полученная таким образом решетка индуцирует планарный оргграф с множеством вершин  $V$  (содержащим  $V_1 \cup \{0\}$ ), являющихся общими точками двух отрезков. Участки отрезков между двумя соседними вершинами есть множество дуг графа. Ориентация дуг определяется направлением возрастания модуля одной из координат вдоль отрезка (вторая координата при этом постоянна), а их длина — метрикой пространства. Пример такого графа приведен на рис. 6.4. Класс графов указанного типа, включающий и все их подграфы, назовем классом  $GL1$ .

**З а д а ч а 1.** Построить дерево Штейнера на графе из класса  $GL1$ .

Рассматриваемую задачу можно сформулировать также в следующем виде. Для заданного множества  $V_1 \cup \{0\}$  построить ордереву  $T$  минимальной длины, соединяющее орпутями (т. е. путями с ориентацией всех дуг от источника к стоку) вершину  $0$  с каждой вершиной  $v \in V_1$  при условии, что каждый орпуть  $[0, v]$  принадлежит множеству кратчайших путей на  $L_1$ -плоскости. К задаче указанного типа сводят-

сы задачи синтеза сетей с одним источником, в которых линейные затраты  $a_e$  и фиксированные доплаты  $b_e$  на каждой дуге  $e$  пропорциональны ее длине  $\rho_e$  в прямоугольной метрике:  $a_e = k_1 \rho_e$ ,  $b_e = k_2 \rho_e$ . При этом константа  $k_1$  столь велика, что для каждой вершины  $v \in V_1$  путь из корня в  $v$  в дереве оптимального решения является кратчайшим среди всех возможных путей на плоскости. Эквивалентность этих двух формулировок непосредственно вытекает из следующих простых утверждений:

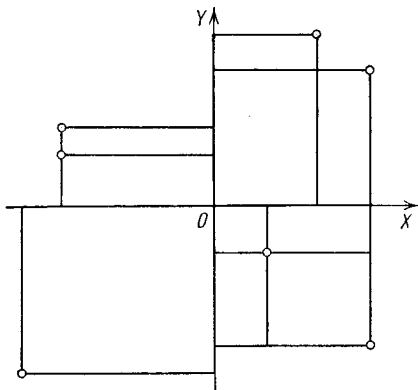


Рис. 6.4

1) решение ЗШГ есть дерево (называемое в дальнейшем деревом Штейнера);

2) существует дерево Штейнера, в котором множество вершин и дуг совпадает с некоторыми подмножествами вершин и дуг графа указанного вида.

Сформулируем вспомогательную задачу следующего вида. Скажем, что плоскость имеет разделяющую полуось (в дальнейшем  $+OY$  — положительный луч оси  $OY$ ), если часть оси, лежащая выше точки  $v_1$ , не должна принадлежать дереву

Штейнера  $T^*$ ; здесь  $v_1$  — точка с максимальной ординатой из множества  $V_1 \cap \{+OY\}$ . Отрезок  $[0, v_1]$  обязательно принадлежит  $T^*$ , поскольку он определяет единственный путь из  $0$  в  $v_1$ .

**Задача 2.** Построить дерево Штейнера на плоскости с разделяющей полуосью.

Ясно, что решение задачи 1 сводится к решению не более, чем  $n + 1$  задач 2 ( $n = |V_1|$ ). Для этого достаточно поочередно полагать точки из  $V_2$ , лежащие на разделяющей полуоси выше, чем  $v_1$ , принадлежащими  $V_1 \cup \{0\}$ , решать полученные задачи 2, а затем выбрать из них наилучшую по функционалу.

Для задачи 2 предлагается полиномиальный алгоритм, который решает задачу, двойственную к линейной релаксации задачи 2. Последняя формулируется как задача о максимальной упаковке разрезов, отделяющих вершины  $V_1$  от корня.

Вершины из  $V_1$  обрабатываются алгоритмом в некотором заданном порядке. При обработке текущей вершины  $v$  для нее строится некоторая система разрезов и перестраивается, вообще говоря, часть разрезов, порожденных предшествующими вершинами из  $V_1$  в заданном упорядочении. Обработка  $v$  (вместе с возможной перестройкой разрезов для предыдущих вершин) осуществляется за полиномиальное время.

Рассмотрим модель ЗШГ в форме задачи булева программирования и задачу, двойственную к ее непрерывной релаксации (т. е. к задаче без условий целочисленности). Пусть  $v \in V_1$ ,  $S(v)$  — произвольное множество вершин из  $V$  такое, что  $0 \notin S(v)$ ,  $v \in S(v)$ . Множество дуг из  $E$ , начало которых не принадлежит  $S(v)$ , а конец принадлежит

$S(v)$ , назовем разрезом  $R(v, S)$ , отделяющим  $v$  от 0. Поставим в соответствие дуге  $e \in E$  переменную  $z_e = 0 \vee 1$ . Тогда ЗШГ формулируется в виде: найти

$$F(Z^*) = \min \left\{ \sum_{e \in E} c_e z_e : \sum_{e \in R(v, S)} z_e \geq 1 \quad \forall S(v), v \in V_1, z_e \geq 0, \right. \\ \left. z_e = 0 \vee 1, e \in E \right\}. \quad (6.15)$$

Задача, двойственная к непрерывной релаксации (6.15), имеет вид: найти

$$\Phi(W^*) = \max \left\{ \sum_{v \in V_1} \sum_{R=R(v, S)} w(v, R) : \sum_{R(v, S) \ni e} w(v, R) \leq c_e, \right. \\ \left. e \in E, w(v, R) \geq 0 \quad \forall R = R(v, S) \right\}. \quad (6.16)$$

В общей ЗШГ  $F(Z^*) \geq \Phi(W^*)$ . Мы покажем, что для случаев, когда граф удовлетворяет условиям задачи 2, имеет место равенство  $F(Z^*) = \Phi(W^*)$ , т. е. среди оптимальных решений непрерывной релаксации задачи (6.15) найдется и оптимальное целочисленное решение.

Пусть  $\bar{W}$  — допустимое решение задачи (6.15). Обозначим

$$N(\bar{W}) = \left\{ e : \sum_{R(v, S) \ni e} \bar{w}(v, R) = c_e \right\}, \\ P(\bar{W}) = \{ R(v, S) : \bar{w}(v, R) > 0 \}.$$

Назовем решение  $\bar{W}$  *правильным*, если:

1) значение  $\bar{w}(v, R)$  нельзя увеличить ни для какого разреза  $R = R(v, S)$ , не изменяя значений других  $\bar{w}(v', R')$ , т. е.  $\bar{W}$  — тупиковое решение;

2) не существует пары  $R(v', S'), R(v'', S'') \in P(\bar{W})$  таких, что  $v' \neq v'', R(v', S') \subset R(v'', S'')$ ;

3) если  $R(v, S) \in P(\bar{W})$ , то  $\forall v' \in S(v) |v'| \leq |v|$ , т. е. координаты  $v'$  не превосходят по модулю координат  $v$  и все  $v' \in S(v)$  лежат в одном квадранте с  $v$ .

**Лемма 6.5.** Среди оптимальных решений задачи (6.16) имеется *правильное*.

**Доказательство.** Ясно, что если решение  $\bar{W}$  не тупиковое, то оно не оптимальное. Пусть в оптимальном решении  $W^*$  найдутся  $R(v', S'), R(v'', S'') \in P(W^*)$ ,  $v' \neq v''$  такие, что  $R(v', S') \subset R(v'', S'')$ , тогда решение

$$\tilde{W} = (\tilde{w}(v, R)) = \begin{cases} w^*(v', R') + w^*(v'', R'') & \text{для } v = v', S = S' \\ 0 & \text{для } v = v'', S = S'' \\ W^*(v, R) & \text{в остальных случаях} \end{cases}$$

очевидно также допустимо и оптимально. Повторяя подобную операцию с каждой парой разрезов с указанным свойством, построим через конечное число шагов оптимальное решение, удовлетворяющее требованию 2). Пусть  $W^*(v, R)$  такое, что  $R(v, S) \in P(W^*)$ , — компонента оптимального решения, для которой  $S$  содержит вершину  $v'$  такую, что  $|v'| \geq |v|$ . Полагая  $\tilde{w}(v, R(v, S \setminus \{v'\})) = w^*(v, R(v, S))$ ,  $\tilde{w}(v, R(v, S)) = 0$ , получим допустимое и оптимальное решение. Повторяя эту процедуру, построим оптимальное решение, удовлетворяющее требованию 3).

Назовем правильное решение  $\bar{W}$  *несдвигаемым*, если для любой положительной компоненты  $\bar{w}(v, R)$  из каждой вершины  $u \in S(v)$  имеется путь  $[u, v]$ , для всех дуг  $e$  которого имеет место соотношение

$$\bar{c}_e = c_e - \sum_{R(v, S) \ni e} \bar{w}(v, R) = 0. \quad (6.17)$$

Значение  $\bar{w}(v, R) = d$  интерпретируется как величина уменьшения длин всех дуг разреза  $R$ , который отделяет 0 от вершины  $v$ . Каждый кратчайший путь из 0 в  $v$  на плоскости с  $L_1$ -метрикой лежит в прямоугольнике  $\Pi(v)$ , определяемом пересечением квадранта, содержащего точку  $v$ , с множеством решений системы неравенств  $|x| \leq |x_v|$ ,  $|y| \leq |y_v|$ , где  $x_v, y_v$  — координаты точки  $v$ . Геометрически каждое  $\bar{w}(v, S)$  из несдвигаемого решения можно изобразить как часть  $\Pi(v)$ , состоящую из множества точек, отстоящих от  $v$  на расстояние, не превышающее  $d$ . При этом расстояние от произвольной точки  $u$  до  $v$  вычисляется с учетом ранее определенных значений других переменных: любой отрезок пути из 0 в  $v$ , лежащий целиком внутри построенных ранее частей, имеет нулевую длину. Пример геометрического изображения правильного несдвигаемого решения приведен на рис. 6.5. Внутри каждой части, соответствующей  $\bar{w}(v, S)$ , поставлен индекс порождающей вершины  $v$ .

*Лемма 6.6. Среди правильных оптимальных решений имеются несдвигаемые.*

*Доказательство.* Пусть  $W^*$  — правильное оптимальное решение, в котором найдется вершина  $u \in V_1$ , не удовлетворяющая определению. Тогда зафиксируем значения  $W^*(v, R)$  для всех  $v \neq u$  и применим алгоритм Дейкстры для построения системы разрезов, опреде-

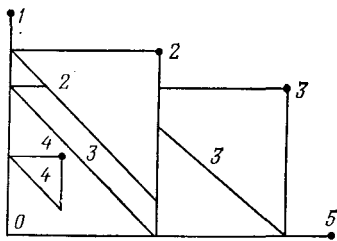


Рис. 6.5

ляющих кратчайший путь  $[0, u]$  в графе с весами дуг  $c_e = c_e - \sum_{R(v, S) \ni e} w^*(v, R)$ . При этом, в отличие от обычного, будем строить растущее дерево кратчайших путей, начиная не с вершины 0, а с вершины  $u$ . Легко понять, что либо разрезы, выбранные при этом алгоритмом Дейкстры, в объединении с решением  $W^*$  для всех  $v \neq u$

образуют несдвигаемое решение, либо найдется  $u' \in V_1$ ,  $u' \neq u$ , для которой можно повторить описанную процедуру. Таким образом, в конце концов будет получено несдвигаемое правильное оптимальное решение.

Перейдем теперь к изложению алгоритма построения такого решения. Для простоты изложения мы предполагаем далее, что задача невырождена, т. е. никакая пара точек из  $V_1$  не лежит на луче, порождающем решетку рассматриваемого графа (этого легко добиться обычными  $\varepsilon$ -смещениями точек из  $V_1$ ). В невырожденной ситуации существует единственное дерево Штейнера (с точностью до способа соединения его вершин орпутями, от которых нет ответвлений). Алгоритм состоит из трех этапов.

На первом этапе вводится упорядочение вершин из  $V_1$  по следующим правилам.

1. Точка  $u$  предшествует  $v$  ( $u \prec v$ ), если номер квадранта, содержащего  $u$ , меньше соответствующего номера для  $v$ . Точки, лежащие на осях координат, принадлежат квадранту с меньшим номером.

2. Для точек  $u, v$ , лежащих в квадрантах I, III,  $u \prec v$ , если  $|x_u| < |x_v|$ , а для квадрантов II, IV — если  $|y_u| < |y_v|$ . В силу невырожденности задачи эти неравенства всегда выполняются как строги.

Пусть  $(v_1, v_2, \dots, v_n)$  — упорядоченная последовательность точек из  $V_1$ . Соединим  $v_1$  с 0. Этот отрезок является обязательной частью дерева Штейнера  $T^*$ . Все точки, лежащие на разделяющей полуоси выше  $v_1$ , не могут быть включены в  $T^*$ . Поэтому они из рассмотрения исключаются. Если рассматривать задачу 2 непосредственно на плоскости (без построения графа), это требование эквивалентно раздвоению соответствующего отрезка полуоси. При этом один из полученных отрезков принадлежит первому квадранту, а другой — четвертому.

На втором этапе алгоритм строит последовательность ненулевых значений  $\omega^*(v_k, S)$ , просматривая точки  $v_k$  ( $k = 2, \dots, n$ ) в определенном на первом этапе порядке. Каждое  $\omega^*(v_k, S)$  для заданных  $v_k$  и  $S$  изображается на плоскости как множество точек, отстоящих от  $v_k$  на расстояние, не превышающее  $d = \omega^*(v_k, S)$ , и лежащих в прямоугольнике  $\Pi(v_k)$ . Расстояние вычисляется, как указано выше, с учетом определенных ранее значений  $\omega^*(v_k, S)$ . Процесс построения последовательности  $\omega^*(v_k, S)$  для данного  $k$  заканчивается, как только образуется путь нулевой длины из 0 в  $v_k$ . В процессе изменяются (в случае необходимости) значения  $\omega^*(v_i, S)$  для  $i \leq k$ .

По построенным значениям  $W^* = \{\omega^*(v_k, S)\}$  на третьем этапе строится  $T^*$  — оптимальное дерево Штейнера.

Приведем формальное описание алгоритма в геометрических терминах. Все его построения выполняются непосредственно на плоскости.

#### Алгоритм решения задачи 2.

Шаг 1. Упорядочить точки из  $V_1 = (v_1, v_2, \dots, v_n)$ .

Шаг 2. Провести операцию раздвоения части разделяющей полуоси. Положить  $\Phi = y_{v_i}$ ,  $k = 2$ .



Шаг 3. Построить систему правильных несдвигаемых разрезов и значений  $\bar{\omega}(v_k, S)$  до образования пути  $l_k$  из 0 в  $v_k$  длины 0. Положить  $d_k = \sum_S \bar{\omega}(v_k, S)$ .

Шаг 4. Если каждый разрез  $S$  со значением  $\bar{\omega}(v_i, S) > 0 \quad \forall i < k$  пересекает путь  $l_k$  не более одного раза, положить  $\Phi = \Phi + d_k$  и перейти к шагу 6.

Шаг 5. Найти  $\bar{\omega}(v_i, S)$  с максимальным значением  $i$ , для которого  $S$  пересекает  $l_k$  более одного раза. Соответствующее число пересечений обозначим  $s_k$ . Уменьшить  $\bar{\omega}(v_i, S)$  на минимальную величину  $\delta$ ,

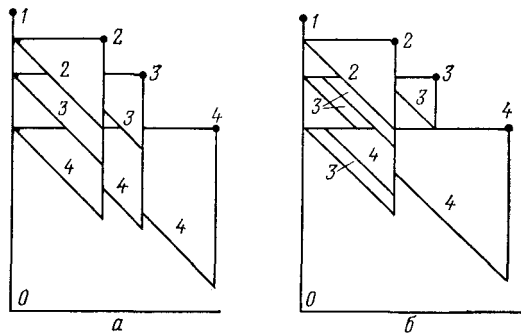


Рис. 6.6

где  $\delta$  выбирается так, что уменьшенное  $\bar{\omega}(v_i, S)$  пересекает  $l_k$  не более одного раза. Дополнить для  $v_k$  систему правильных несдвигаемых разрезов и значений  $\bar{\omega}(v_k, S)$  так, что значение  $d_k$  увеличилось на максимальную величину ( $\leq (s_k - 1)\delta$ ). Дополнить для  $v_i$  систему разрезов и значений  $\bar{\omega}(v_i, S)$  до появления пути  $l_i$  нулевой длины (при этом появится путь  $l_k$  нулевой длины). Перейти к шагу 4.

Шаг 6. Если  $k < n$ , положить  $k := k + 1$  и перейти к шагу 3.

Шаг 7. Положить  $T^n = [0, v_1]$ ,  $W^* = \bar{W}$

Шаг 8. Найти кратчайший путь  $l_k$  (в метрике  $L_1$ ) из  $T^k$  в  $v_k$ , величина лежащий в областях плоскости, порожденных разрезами  $S$  и величинами  $\omega^*(v_i, S) \quad \forall i \leq k$ .

Шаг 9. Положить  $T^{k-1} = T^k \cup l_k$ . Удалить области плоскости, порожденные всеми  $\omega^*(v_i, S)$  для данного  $k$ .

Шаг 10. Если  $k > 2$  (при  $y_{v_i} = 0$  — если  $k > 1$ ), положить  $k := k - 1$ . Перейти к шагу 8.

Шаг 11. Результирующее дерево выбрать в качестве  $T^*$ . Стоп.

Как показано ниже, длина  $T^*$  равна  $\Phi$ .

Пример действия алгоритма на шаге 5 проиллюстрирован на рис. 6.6. На нем показана ситуация до и после работы алгоритма на шаге 5.

Обоснование алгоритма базируется на следующем утверждении.

**Теорема 6.3.** Пусть  $W^*$  — правильное неподвижное решение. Тогда дерево  $T$ , построенное по правилам этапа 3, оптимально только в том случае, если не найдется разреза  $R(v, S)$  с  $w^n(v, S) > 0$ , который по крайней мере дважды пересекает  $T$ .

**Доказательство.** При построении нижней оценки величина каждого разреза суммируется ровно один раз, а при построении дерева — столько, сколько раз разрез пересекает его. Следовательно, значение нижней оценки совпадает с длиной  $T$  лишь в случае выполнения условий теоремы.

**Следствие.** Алгоритм строит оптимальное решение задачи.

**Доказательство.** Достаточно показать, что каждый разрез пересекает  $T$  точно один раз. По построению каждый разрез пересекает любое связывающее ордеровето (в том числе и  $T$ ) не менее одного раза. Пусть найдется разрез  $R(v_i, S)$ , пересекающий  $T$  более одного раза. Априори возможны два случая.

1.  $R(v_i, S)$  пересекает путь  $l_k$ , определяемый на шаге 8, более одного раза. В этом случае  $k > i$  и из невырожденности задачи путь  $l_k$  единственный. Указанная ситуация противоречит действиям на шаге 5.

2.  $R(v_i, S)$  пересекает два непересекающихся пути  $l_k$  и  $l_m$ . В этом случае  $k > i$  и  $m > i$ . Пусть для определенности  $k > m$ . Тогда при построении  $T^*$  разрез  $R(v_i, S)$  будет удален из рассмотрения при построении  $l_k$  и поэтому не может пересечь  $l_m$ . Поэтому путь из 0 в  $v_m$  либо не пересекает  $R(v_i, S)$ , либо имеет общую часть с путем  $l_k$ , проходящую через этот разрез.

Следствие доказано.

#### § 4. Задача Вебера в пространстве с прямоугольной метрикой

При решении ряда задач возникает необходимость размещения в пространстве с заданной метрикой или в вершинах графа группы взаимосвязанных между собой объектов по критерию минимизации суммарной работы по осуществлению требуемых коммуникаций как между размещаемыми, так и между ранее размещенными на данном множестве объектами. Естественным требованием при этом является допустимость размещения каждого объекта только в одной из заданных точек. Иногда требуется также, чтобы в любой из возможных точек размещалось в оптимальном решении задачи не более чем заданное количество точек. Задачи указанного класса в случае конечного числа мест возможного размещения формулируются обычно как задачи целочисленного программирования с квадратичной целевой функцией и транспортными ограничениями на булевы переменные, которые соответствуют вариантам возможного размещения заданного множества размещаемых точек. Квадратичное слагаемое целевой функции в таких задачах соответствует суммарному объему работы между всеми парами размещаемых точек, а линейная часть — между размещаемыми и ранее размещенными точками. Задачи рассматриваемого класса называются квадратичными задачами о назначении и как в об-

щем, так и в многочисленных частных случаях относятся к трудно-разрешимым. Однако, если отказаться от требования на размещение в каждой заданной точке ограниченного числа размещаемых точек, задача в ряде случаев становится полиномиально разрешимой.

К этому случаю относится так называемая задача Вебера в метрическом пространстве, которая формулируется следующим образом. В метрическом пространстве задано  $n$  точек  $A_1, A_2, \dots, A_n$ . В нем же требуется разместить  $m$  новых точек  $X_1, X_2, \dots, X_m$ . Каждой упорядоченной паре индексов  $(i, j)$  ( $i = 1, \dots, m, j = 1, \dots, n$ ) ставится в соответствие заданная величина потока  $c_{ij} \geq 0$  между точками  $X_i$  и  $A_j$ , а каждой упорядоченной паре  $(i, k)$  ( $i, k = 1, \dots, m, i < k$ ) — величина потока  $d_{ik} \geq 0$  между точками  $X_i$  и  $X_k$ . Требуется выбрать координаты точек  $X_i$  ( $i = 1, \dots, m$ ) так, чтобы достигался

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n c_{ij} r(X_i, A_j) + \sum_{i=1}^{m-1} \sum_{k=i+1}^m d_{ik} r(X_i, X_k) \right). \quad (6.18)$$

Здесь  $r(X, Y)$  — расстояние между  $X$  и  $Y$  в рассматриваемом метрическом пространстве.

В случае так называемых  $L_p$ -пространств, в которых метрика задается формулой

$$r_p(X, Y) = \left( \sum_{i=1}^N |x_i - y_i|^p \right)^{1/p},$$

где  $X = (x_1, \dots, x_N)$ , задача (6.18) относится к классу задач минимизации негладкой выпуклой функции, и для ее решения могут быть использованы методы негладкой оптимизации. В частности, при  $p = 2$  метрика является евклидовой, и в этом случае для решения задачи (6.18) разработаны, кроме общих, и специальные методы (см., например, [150]). В данном разделе рассматривается так называемая прямоугольная (или манхэттенская) метрика, соответствующая  $p = 1$ . В этом случае расстояние определяется по формуле

$$r(X, Y) = r_1(X, Y) = \sum_{k=1}^N |x_k - y_k|, \quad (6.19)$$

где  $N$  — размерность пространства,  $X = (x_1, \dots, x_N)$ ,  $Y = (y_1, \dots, y_N)$ . К задаче (6.18), (6.19) сводятся некоторые задачи наилучшего размещения новых станков в цехе, вокзалов и обслуживающих предприятий в городе и т. п. Эта задача исследовалась рядом авторов (см., например, [159, 176]). Цель данного раздела — показать, как решение задачи (6.18), (6.19) можно свести к решению последовательности задач о максимальном потоке в графе [73, 165].

Рассмотрим ряд свойств задачи (6.18), (6.19).

1. Задача (6.18), (6.19) распадается на  $N$  задач, каждая из которых является одномерной задачей Вебера. В дальнейшем будем рассматривать задачу (6.18), (6.19) в одномерном пространстве, не оговаривая это каждый раз и опуская индекс координаты у всех точек.

2. Задача (6.18), (6.19) является задачей минимизации кусочно линейной выпуклой функции.

3. В оптимальном решении задачи (6.18), (6.19) каждая точка  $X_i$  ( $i = 1, \dots, m$ ) совпадает с одной из точек  $A_j$  ( $j = 1, \dots, n$ ). Это свойство непосредственно следует из свойства 2.

Задача (6.18), (6.19) может быть сформулирована как задача линейного программирования следующим образом: найти

$$\min \left( \sum_{i=1}^m \sum_{j=1}^n c_{ij} s_{ij} + \sum_{i=1}^{m-1} \sum_{k=i+1}^m d_{ik} t_{ik} \right) \quad (6.20)$$

при ограничениях

$$-s_{ij} \leq X_i - A_j \leq s_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (6.21)$$

$$-t_{ik} \leq X_i - X_k \leq t_{ik}, \quad i = 1, \dots, m-1, \quad k = i+1, \dots, m, \quad (6.22)$$

где  $s_{ij}$ ,  $t_{ik}$  — вспомогательные переменные.

Рассмотрим задачу, двойственную к (6.20) — (6.22): найти

$$\max \sum_{j=1}^n A_j \sum_{i=1}^m (U_{ij} - u_{ij}) \quad (6.23)$$

при ограничениях

$$\sum_{j=1}^n (U_{ij} - u_{ij}) - \sum_{k=1}^{i-1} (V_{ki} - v_{ki}) + \sum_{k=i+1}^m (V_{ik} - v_{ik}) = 0, \quad i = 1, \dots, m, \quad (6.24)$$

$$U_{ij} + u_{ij} = c_{ij}, \quad U_{ij} \geq 0, \quad u_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad (6.25)$$

$$V_{ik} + v_{ik} = d_{ik}, \quad V_{ik} \geq 0, \quad v_{ik} \geq 0, \quad i = 1, \dots, m-1, \quad k = i+1, \dots, m, \quad (6.26)$$

где  $U_{ij}$ ,  $u_{ij}$ ,  $V_{ik}$ ,  $v_{ik}$  — двойственные переменные, соответствующие ограничениям (6.21), (6.22). Сделав замену переменных

$$\begin{aligned} z_{ij} &= U_{ij} - u_{ij} + c_{ij}, & Z_{ij} &= U_{ij} + u_{ij}, \\ w_{ik} &= V_{ik} - v_{ik}, & W_{ik} &= V_{ik} + v_{ik}, \\ i, k &= 1, \dots, m, \quad i < k, \quad j = 1, \dots, n, \end{aligned}$$

перепишем задачу (6.23) — (6.26) в следующем виде: найти

$$\max \left\{ F(Z, W) = \sum_{j=1}^n A_j \sum_{i=1}^m z_{ij} + c \right\} \quad (6.27)$$

при ограничениях

$$\sum_{j=1}^n z_{ij} - \sum_{k=1}^{i-1} w_{ki} + \sum_{k=i+1}^m w_{ik} = b_i, \quad i = 1, \dots, m, \quad (6.28)$$

$$0 \leq z_{ij} \leq 2c_{ij}, \quad -d_{ik} \leq w_{ik} \leq d_{ik}, \quad (6.29)$$

здесь  $c$  — константа,  $b_i = \sum_{j=1}^n c_{ij}$  ( $i = 1, \dots, m$ ).

Исследуем свойства задачи (6.27)—(6.29). Без потери общности можно предполагать, что все  $A_j$  ( $j = 1, \dots, n$ ) попарно различны, так как в противном случае точки с одинаковыми  $A_j$  можно заменить одной, модифицировав соответствующим образом матрицу коэффициентов  $c_{ij}$ . Будем полагать, что  $A_j$  упорядочены в порядке их убывания:  $A_1 > A_2 > \dots > A_n$ . Для простоты дальнейших рассуждений будем также предполагать, что все  $c_{ij} > 0$ . Это можно добиться с помощью традиционного  $\epsilon$ -преобразования, которое часто используется в теории линейного программирования. Обозначим через  $(Z^*, W^*) = ((z_{ij}^*), (w_{ik}^*))$  оптимальное решение задачи (6.27)—(6.29).

**Лемма 6.7.** Если  $z_{is}^* > 0$ , то  $z_{ir}^* = 2c_{ir}$ ,  $r = 1, \dots, s-1$ .

**Доказательство.** Пусть  $z_{ir}^* < 2c_{ir}$  для некоторого  $r < s$ . Тогда решение  $(\bar{Z}, \bar{W}^*)$ , где  $\bar{z}_{ij} = z_{ij}^*$  для  $(j \neq r, s) \wedge (l \neq i)$ ,  $\bar{z}_{ir} = z_{ir}^* - \delta$ ,  $\bar{z}_{is} = z_{is}^* + \delta$ ,  $\delta = \min \{z_{ir}^*, 2c_{is} - z_{is}^*\} > 0$ , является допустимым и увеличивает (6.27), что противоречит оптимальности  $(Z^*, W^*)$ .

Пусть  $(Z^1, W^1)$  и  $(Z^2, W^2)$  — допустимые решения (6.27)—(6.29). Положим  $Z^2 = Z^1 + \varphi^1$ ,  $W^2 = W^1 + \omega^1$  и подставим эти выражения в (6.27)—(6.29). Тогда  $\omega^1$  можно интерпретировать как допустимый поток в сети  $G(Z^1, W^1)$  на полном графе с  $m$  вершинами и пропускными способностями дуг  $(i, k)$ , равными  $d_{ik} - w_{ik}^1$ , и дуг  $(k, i)$ , равными  $d_{ik} + w_{ik}^1$  ( $i, k = 1, \dots, m, i < k$ ). Дивергенции в вершинах сети равны

$\sum_{j=1}^n \varphi_{ij}^1$ . Известно (см., например, [3, с. 11]), что любой поток представим в виде суммы элементарных потоков, т. е. потоков вдоль простых цепей из источника в сток и потоков по элементарным циклам. В описанной интерпретации вектор-решение  $W^2$  для произвольного  $(Z^2, W^2)$  можно рассматривать также как допустимый поток, если в качестве  $(Z^1, W^1)$  выбрать вектор  $(Z^1, 0)$  (такой всегда существует).

Поскольку изменение потоков вдоль элементарных циклов не меняет значения целевой функции (6.27), будем считать, что разложение  $W^1$  для любого решения  $(Z^1, W^1)$  не содержит их. Кроме того, в дальнейшем будем рассматривать лишь решения (не обязательно оптимальные), удовлетворяющие условиям леммы 6.7. Для допустимого решения  $(Z^1, W^1)$  положим

$$j_i^-(Z^1) = \min \{j: z_{ij}^1 < 2c_{ij}\}, \quad j_i^+(Z^1) = \max \{j: z_{ij}^1 > 0\}, \quad i = 1, \dots, m.$$

**Лемма 6.8.** Решение  $(Z^*, W^*)$  оптимально, если в сети  $G(Z^*, W^*)$  пропускная способность каждой дуги  $(i, k)$  такой, что

$$j_i^+(Z^*) > j_k^-(Z^*), \quad i, k = 1, \dots, m, \quad (6.30)$$

равна нулю.

**Доказательство.** Пусть  $(\bar{Z}, \bar{W})$  — произвольное допустимое решение. Тогда в разложении потока  $\omega^+ = \bar{W} - W^*$  каждый элементарный ненулевой поток будет соответствовать пути, для каждой дуги которого условия (6.30) не выполняются. Рассмотрим после-

довательность решений  $(Z^*, W^*), \dots, (\bar{Z}, \bar{W})$ , соответствующую произвольной последовательности элементарных потоков в  $W^*$ , где каждое решение  $(Z^s, W^s)$  ( $s = 1, 2, \dots$ ) определяется так:  $(Z^1, W^1) = (Z^*, W^*)$ ,  $W^{s+1} = W^s + \lambda^s$ ,  $Z^{k+1} = Z(W^{k+1})$  и  $Z^{k+1}$  удовлетворяет условиям леммы 6.7; здесь  $\lambda^k$  — элементарный поток. Пусть  $p$  — источник,  $q$  — сток,  $\mu$  — величина потока  $\lambda^k$ . Тогда

$$\sum_{j=1}^n z_{pj}^{k+1} < \sum_{j=1}^n z_{pj}^k, \quad \sum_{j=1}^n z_{qj}^{k+1} > \sum_{j=1}^n z_{qj}^k,$$

$$j_p^+(Z^k) \leq j_q^-(Z^k),$$

$$F(Z^{k+1}, W^{k+1}) - F(Z^k, W^k) = \sum_{j=1}^n A_j (z_{pj}^{k+1} - z_{pj}^k) + \sum_{j=1}^n A_j (z_{qj}^{k+1} - z_{qj}^k) \leq \mu (A_{j_p^+(Z^k)} - A_{j_q^-(Z^k)}) \leq 0, \quad k = 1, 2, \dots \quad (6.31)$$

Складывая неравенства (6.31) для всех соседних пар рассматриваемой последовательности решений, получим  $F(Z^*, W^*) - F(\bar{Z}, \bar{W}) \leq 0$ , откуда в силу произвольности  $(\bar{Z}, \bar{W})$  следует оптимальность  $(Z^*, W^*)$ .

Опишем алгоритм решения задачи (6.27)—(6.29), который требует выполнения  $R$  итераций.

Шаг 1. Положить  $W^1 = 0$ ,  $b_i^1 = 0$  ( $i = 1, \dots, m$ ),  $r = 1$ .

Шаг 2. Положить  $z_{ir} = \min \{b_i^r, 2c_{ir}\}$  ( $i = 1, \dots, m$ ).

Шаг 3. Если все  $\bar{z}_{ir} = 2c_{ir}$ , положить  $z_{ir}^* = z_{ir}$ ,  $b_i^{r+1} = b_i^r - z_{ir}^*$  ( $i = 1, \dots, m$ ),  $W^{r+1} = W^r$  и перейти к шагу 8.

Шаг 4. Построить сеть  $G^r$  с вершинами  $0, 1, \dots, m, m+1$ . Вершина 0 — источник,  $m+1$  — сток. Определить пропускные способности дуг  $d_{ik}^r$  по правилам

$$d_{0\ m+1}^r = 0,$$

$$d_{0i}^r = b_i^r - \bar{z}_{ir}, \quad d_{i\ m+1}^r = 2c_{ir} - \bar{z}_{ir}, \quad i = 1, \dots, m,$$

$$d_{ik}^r = \begin{cases} d_{ik}^r - \omega_{ik}^r, & \text{если } i < k, \\ d_{ki}^r + \omega_{ki}^r, & \text{если } i > k, \end{cases} \quad i, k = 1, \dots, m, \quad i \neq k.$$

Шаг 5. Построить максимальный поток  $\omega^r$  в  $G^r$ .

Шаг 6. Положить  $z_{ir}^* = \bar{z}_{ir} + \omega_{i\ m+1}^r$ ,  $b_i^{r+1} = b_i^r - z_{ir}^* - \omega_{0i}^r$ ,  $i = 1, \dots, m$ ,  $\omega_{ik}^{r+1} = \omega_{ik}^r + \omega_{ik}^r - \omega_{ki}^r$ ,  $i, k = 1, \dots, m$ ,  $i \neq k$ .

Шаг 7. Если не все  $b_i^{r+1} = 0$ , перейти к шагу 8. В противном случае положить  $z_{is}^* = 0$  ( $i = 1, \dots, m$ ,  $s = r+1, \dots, n$ ),  $W^* = W^{r+1}$  и закончить вычисления.

Шаг 8. Если  $r < R$ , положить  $r := r+1$  и перейти к шагу 2.

Шаг 9. Положить  $W^* = W^{R+1}$ ; Стоп.

Теорема 6.4. Алгоритм строит оптимальное решение задачи (6.27)—(6.29). Его трудоемкость —  $O(nm^3)$ .

Доказательство. Решение  $(Z^*, W^*)$  допустимо и удовлетворяет условиям леммы 6.7, 6.8, поэтому оно оптимально. Трудоемкость алгоритма построения максимального потока —  $O(m^3)$  [3]. Наиболее трудоемким в алгоритме является именно шаг 5, который выполняется не более  $n$  раз. Отсюда следует указанная оценка трудоемкости.

Зная оптимальное решение задачи (6.27)—(6.29), нетрудно построить оптимальное решение исходной задачи (6.18), (6.19). Вычислим

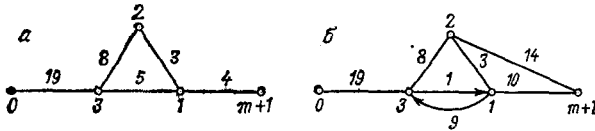


Рис. 6.7

$\bar{f}_i^+(Z^*)$  и положим  $X_i^* = A_{i\bar{f}_i^+(Z^*)}$  ( $i = 1, \dots, m$ ). Оптимальность  $X^* = (X_1^*, \dots, X_m^*)$  немедленно следует из теории двойственности в линейном программировании.

Таблица 6.1

$r, j$		1	2	3	4	5
I	$A_j$	10	9	7	2	0
	$c_{1j}$	16	2	5	6	3
	$c_{2j}$	9	6	7	4	4
	$c_{3j}$	2	3	1	8	15
II	$b_1^r$	32	0	0	0	
	$b_2^r$	30	12	0	0	
	$b_3^r$	29	25	15	4	
III	$\bar{z}_{1r}$	32	0	0	0	
	$z_{1r}^*$	32	4	1	0	0
	$z_{2r}^*$	18	12	0	0	
	$z_{2r}^*$	18	12	8	0	0
	$z_{3r}^*$	4	6	2	4	
IV	$\omega_{12}^r$		0	0		
	$\omega_{12}^r$		0	0		
	$\omega_{13}^r$		-4	-1		
	$\omega_{13}^r$		-4	-5		
	$\omega_{23}^r$		0	-8		
	$\omega_{23}^r$		0	-8		

Проиллюстрируем работу описанного алгоритма на примере. Исходные данные и процесс решения представлены в табл. 6.1. Здесь  $n = 5$ ,  $m = 3$ ,  $b_i = b_i^1$  ( $i = 1, 2, 3$ ),  $d_{12} = 3$ ,  $d_{13} = 5$ ,  $d_{23} = 8$ .

Знак «минус» в табл. 6.1 (IV) означает, что соответствующее положительное значение присвоено величине  $\omega_{ki}^2$  (или  $\omega_{ki}^r$ ) при  $k > i$ . Например,  $\omega_{31}^3 = 5$ . При этом  $\omega_{13}^3 = 0$ . На рис. 6.7 показаны сети  $G_2$  и  $G_3$  (рис. 6.7, а и 6.7, б). Дуги с нулевой пропускной способностью опущены. Связи без стрелок означают ребра;  $j_1^+(Z^*) = j_2^+(Z^*) = 3$ ,  $j_3^+(Z^*) = 4$ . Отсюда  $X_1^* = X_2^* = A_3 = 7$ ,  $X_3^* = A_4 = 2$  и значение (6.18) равно 327.



**ПАКЕТ ПРИКЛАДНЫХ ПРОГРАММ  
ДЛЯ РЕШЕНИЯ ЗАДАЧ  
ПРОИЗВОДСТВЕННО-ТРАНСПОРТНОГО ПЛАНИРОВАНИЯ  
БОЛЬШОЙ РАЗМЕРНОСТИ (ПЛАНЕР)**

---

Пакет ПЛАНЕР создан в Институте кибернетики имени В. М. Глушкова АН УССР для ЕС ЭВМ. Он функционирует под управлением операционных систем ОС/MFT и ОС/MVT и позволяет работать с магнитными дисками типа ЕС-5061 и ЕС-5050, используемыми под исходные данные и результаты работы пакета.

**1. Назначение пакета и состав математического обеспечения.** ППП ПЛАНЕР предназначен для решения специальных классов задач производственно-транспортного планирования большой размерности. Он может быть использован плановыми, проектными и научно-исследовательскими организациями при решении задач планирования производства, распределения и транспортировки продукции, а также размещения и реконструкции производства.

Отличительные особенности ППП ПЛАНЕР, по сравнению с рядом существующих пакетов для решения задач указанного типа:

- существенное расширение множества классов решаемых задач;
- реализация развитого системного обеспечения, позволяющего эффективно решать задачи в пакетном и диалоговом режимах, осуществлять в автоматизированном режиме модернизацию обеспечения пакета, работать с данными, обеспечивать надежность вычислений при решении задач большой размерности, эффективность отладки новых подключаемых к пакету программ;
- использование ряда новых эффективных математических методов решения задач отдельных типов.

Предлагаемые алгоритмы, ориентированные на решение конкретных классов задач и учитывающие их специфические особенности, позволяют существенно увеличить размеры эффективно решаемых задач.

Большинство задач, включенных в ППП ПЛАНЕР, обладают специфическими особенностями, связанными с наличием в матрице ограничений блоков транспортного или распределительного типа либо переменных верхних границ. Как показано в основном тексте, учет этих особенностей при решении задач большой размерности позволяет существенно повысить эффективность специальных методов по сравнению с общими.

В ППП ПЛАНЕР включены оптимизационные модули, реализующие в основном методы, изложенные в основном тексте и предназначенные для решения следующих задач:

- 1) динамическая распределительная задача (§ 3.4);

- 2) многопродуктовая транспортная задача на сети с ограниченными пропускными способностями дуг (§ 3.3);
- 3) трехиндексная распределительная задача с планарными суммами (§ 3.4);
- 4) задача распределения заказов между предприятиями по минимуму производственных и транспортных затрат (§ 3.6);
- 5) стохастическая транспортная задача с вероятностным спросом потребителей (§ 3.7);
- 6) стохастическая двухэтапная транспортная задача с вероятностным спросом (§ 3.7);
- 7) задача линейного программирования блочной структуры с общими ограничениями (§ 3.5);
- 8) задача размещения без ограничений на мощности производства (§ 5.1);
- 9) задача размещения с ограничениями на мощности производства (§ 5.4);
- 10) двухэтапная задача размещения производства (§ 5.3);
- 11) задача размещения и унификации (§ 5.3);
- 12) многомерная нелинейная задача о ранце [47, 48];
- 13) задача нахождения оптимального грузопотока на сети с сохранением баланса на дугах [47, 48].

В ПЛАНЕР включена также программа решения общей задачи выпуклого программирования (ПВП). Применяя ПВП, пользователь имеет возможность решать оптимизационные задачи, математические модели которых отличны от указанных выше. При этом для задач, имеющих блочную структуру, ПВП позволяет реализовать двухэтапную схему решения (§ 2.4).

Кроме указанных классов в пакет включены прикладные модули решения некоторых классических задач математического программирования (транспортная задача в матричной и сетевой форме, кратчайшие расстояния на сети и др.). Эти модули, будучи доступными пользователю, применяются как вспомогательные в других прикладных программах пакета.

Методы и программы решения транспортной задачи в сетевой и матричной формах разработаны в ЦЭМИ АН СССР М. Л. Бронштейном, К. В. Кимом и Б. В. Черкасским.

**2. Структура системного обеспечения пакета и функциональное назначение его основных частей.** Системное обеспечение ППП ПЛАНЕР имеет сложную структуру и является дальнейшим развитием математического обеспечения пакета ДИСПРО [46]. Основными языковыми, программными и информационными средствами данного пакета являются: входной непроцедурный язык, управляющая программа, набор прикладных модулей решения задач заданного класса, модель предметной области, базы данных, сервис.

Входной проблемно-ориентированный язык ППП ПЛАНЕР служит для взаимодействия пользователей с пакетом и предназначен для описания исходных данных решаемых задач, их параметров и результатов, а также для управления процессом решения задач в пакетном и диалоговом режимах. Язык состоит из операторов, которые имеют

следующий формат:

(имя оператора) (список параметров) (комментарий).

Различаются операторы: управления выполнением задания (НАЧАЛО, КОНЕЦ ШАГА, КОНЕЦ ЗАДАНИЯ, ПЕРЕЙТИ), описания задач и данных (РЕШИТЬ, ДАННЫЕ, СОХРАНИТЬ), манипулирования данными (ЗАПИСАТЬ, УДАЛИТЬ, МОДИФИЦИРОВАТЬ, ИЗМЕНИТЬ, ПЕЧАТЬ). На их основе составляется задание пакету, которое можно разбить на несколько шагов. Каждый шаг предполагает либо решение одной оптимизационной задачи, либо работу с данными. Такое разбиение позволяет пользователю управлять вычислительным процессом, иницируемым пакетом.

Ряд прикладных задач, решаемых с помощью ППП ПЛАНЕР, характеризуется не только большим объемом, но и достаточно сложной структурой исходной информации. При создании пакета одной из важнейших была проблема разработки такой организации исходной информации, которая позволила бы удобно и в единой форме представить исходные данные для широкого спектра задач пакета. Эта форма должна обеспечить возможность представления исходных данных каждой конкретной задачи в естественном виде, вытекающем из содержательной интерпретации ее данных.

Для решения задач большой размерности пользователь, как правило, генерирует исходные данные программно, используя при этом свои информационные базы данных. В связи с этим предложенная единая форма представления исходных данных прикладных программ пакета отражает естественную их организацию для каждой задачи, определяемую группировкой данных по содержательному смыслу. Это, на наш взгляд, позволяет достаточно просто разрабатывать программы, генерирующие исходные данные с помощью информационных баз пользователя.

Анализ прикладных задач оптимизации, для решения которых предназначен ППП ПЛАНЕР, показал, что достаточно универсальной и удобной формой организации данных является представление их в виде упорядоченного набора блоков данных, каждый из которых имеет древовидную (иерархическую) структуру. Структура исходных данных должна соответствовать структуре, определенным образом описанной и хранящейся в паспорте каждого прикладного модуля.

Такой подход позволяет в значительной степени упростить генерацию исходных данных, производить ввод и контроль неоднородной информации, подготовка которой может проводиться на уровне содержательной постановки задачи. При этом повышается качество контроля исходных данных, контроль и обработка информации производится без задания ее общего объема.

Исходные данные, представленные файлами последовательного доступа, созданными пользователем, для обработки и контроля помещаются в файлы прямого доступа (которые могут быть по требованию пользователя как временными, так и постоянными), где производится синтаксический и содержательный контроль данных, а при необходимости — корректировка в диалоговом режиме. Предусмотрены про-

смотр и коррекции символьной информации, находящейся в файлах прямого доступа. Программа, реализующая эти функции, выполняет следующие действия: чтение страницы текста, чтение по номеру записи, запись страницы текста, добавление записей.

Файл исходных данных состоит из подфайлов (блоков), каждый из которых содержит информацию об одной древовидной структуре.

Числовые данные представляются в символьном виде, где числа разделены запятыми. На объем информации не накладываются никакие ограничения; фактически он ограничен лишь величиной физической записи файла.

Синтаксический и содержательный контроль исходной информации производится по записям файла. При синтаксическом контроле определяются недопустимые символы, имеющие место в исходных данных, производится их локализация, выдача соответствующего диагностического сообщения и непосредственно локализованной ошибки на АЦПУ и экран дисплея с указанием недопустимых символов, что необходимо для оперативного исправления ошибок в процессе обработки в диалоговом режиме. При содержательном контроле проверяется соответствие структуры исходных данных ее описанию в паспорте прикладного модуля, а также принадлежность величины каждого числа требуемому диапазону. Такой контроль производится сразу же после синтаксического контроля одной из записей файла. Осуществив ввод некоторой числовой информации, по описанию структуры можно определить вид данных, которые должны вводиться в настоящий момент. Кроме того, определяются диапазоны величин, которым должны принадлежать величины вводимых числовых данных.

При обнаружении ошибок проводится определение их характера, локализация и выдача на АЦПУ или экран дисплея вместе с соответствующим диагностическим сообщением.

Осуществив синтаксический и семантический контроль, пакет формирует некоторую дополнительную информацию, необходимую для работы прикладных модулей, входящих в его состав. Эта информация и непосредственно исходные данные позволяют управляющей программе ППП ПЛАНЕР сформировать рабочие файлы для работы прикладных модулей, после чего управление передается прикладным программам. Результаты решения той или иной оптимизационной задачи по требованию пользователя могут быть выданы на АЦПУ в виде таблиц или записаны в файл прямого доступа в соответствии со структурой выходных данных, описанной в паспорте каждого прикладного модуля.

Пользователю предоставляются также возможность подключить свой модуль выдачи результатов и получить решение задачи в необходимой форме. Такими дополнительно подключенными модулями могут быть, например, подпрограмма вычисления значений функционала или градиента, необходимые для работы других модулей. Для подключения подпрограмм пользователя нужно поместить их в загрузочном (оттранслированном) виде в библиотеку загрузочных модулей и указать с терминала имя необходимой подпрограммы при соответствующем запросе пакета.

Диалоговый режим работы, предусмотренный в ППП ПЛАНЕР, осуществляется на основе диалоговых прикладных программ и дает возможность пользователю активно влиять на процесс ввода, трансляции и выполнения заданий, в том числе на процесс решения оптимизационных задач. При соответствующих запросах монитора, выдаваемых на экран дисплея, пользователь может исправить с дисплея ошибки, допущенные при вводе операторов, повторить ввод исходных данных, всего задания или его шагов, оперативно изменять шаги и последовательность их выполнения, задавать и изменять исходные данные задач и параметры методов их решения, выводить результаты решения на экран дисплея.

В пакете имеются средства создания контрольных точек и рестарта (повторного запуска). Такие точки дают возможность сохранять информацию о состоянии пакета в фиксированные моменты его работы и совершать рестарт из этих точек, что позволяет в определенной степени защитить пакет от системных сбоев ОС, в среде которой он функционирует.

Во время функционирования пакета могут возникнуть: постоянная или случайная машинная ошибка; нехватка выделенного машинного времени для пакета; ошибка, вызванная неправильными действиями оператора или параллельно выполняющимся заданием, и др. Следствием этого могут быть: аварийное окончание работы пакета; системный сбой; неправильные результаты; завершение работы пакета по истечении заказанного времени. Во избежание повторного запуска пакета с самого начала, что потребовало бы больших затрат машинного времени, пользователю предоставляется возможность продолжать работу пакета с некоторой промежуточной точки, предшествующей возникновению аварийной ситуации. Более того, можно указать, будет ли рестарт происходить автоматически в момент сбоя пакета, или его следует отсрочить. Автоматический рестарт предусмотрен лишь в тех случаях, когда ход завершения работы пакета при аварийном прерывании соответствует одному из допустимых кодов завершения, определенных во время генерации ОС, и, кроме того, рестарт должен быть разрешен оператором ЭВМ. Автоматический рестарт производится из последней контрольной точки, созданной до момента аварийного окончания работы пакета.

При повторном предъявлении задания на запуск пакета пользователю предоставляется возможность отсроченного рестарта, который может быть осуществлен с любой указанной им контрольной точки.

Более детальное описание пакета приведено в [47, 48] и технической документации, распространяемой Республиканским фондом алгоритмов и программ УССР.

1. Абрамов Н. Н., Малов В. Н. О надежности систем водоснабжения и путях ее обеспечения.— Изв. АН СССР. Энергетика и транспорт, 1976, №1, с. 161—170.
2. Александров И. А., Анциферов Е. Г., Булатов В. П. Методы центрированных сечений в выпуклом программировании.— Препринт; Иркутск, СЭИ СО АН СССР, 1983.— 33 с.
3. Адельсон-Вельский Г. М., Диниц Е. А., Карзанов А. В. Поточковые алгоритмы.— М.: Наука, 1975.— 118 с.
4. Баженов Л. Г. Об условиях сходимости метода минимизации почти дифференцируемых функций.— Кибернетика, 1972, № 4, с. 71—72.
5. Бакаев А. А., Михалевич В. С., Брановицкая С. В., Шор Н. З. Методика и опыт решения сетевых транспортных задач большого объема на ЭЦВМ.— В кн.: Математические методы и проблемы производства. М.: Наука, 1963, с. 247—257.
6. Беляева Л. В., Журбенко Н. Г., Шор Н. З. Об одной нелинейной транспортной задаче.— В кн.: Тр. семинара Науч. совета АН УССР по кибернетике «Теория оптим. решений». Киев: ИК АН УССР, 1976, с. 83—89.
7. Беляева Л. В., Журбенко Н. Г., Шор Н. З. О методе решения одного класса динамических распределительных задач.— Экономика и мат. методы, 1978, 14, № 1, с. 137—146.
8. Берж К. Теория графов и ее приложение.— М.: ИЛ, 1962, 320 с.
9. Береснев В. Л., Гимади Э. Х., Дементьев В. Т. Экстремальные задачи стандартизации.— Новосибирск: Наука, 1978.— 333 с.
10. Береснев В. Л. Алгоритм неявного перебора для задачи типа размещения и стандартизации.— В кн.: Управляемые системы. Вып. 12. Новосибирск: Ин. математики СО АН СССР, 1974, с. 24—34.
11. Бесконечные антагонистические игры / Под ред. Н. Н. Воробьева. — М.: Физматгиз, 1963.— 503 с.
12. Булавский В. А., Звягина Р. А., Яковлева Н. А. Численные методы линейного программирования.— М.: Наука, 1977.— 368 с.
13. Булавский В. А. Итеративный метод решения общей задачи линейного программирования.— В кн.: Численные методы оптимального планирования, вып. 1, эконом.-мат. сер. Новосибирск: СО АН СССР, 1962, с. 35—64.
14. Быковченко М. А. Оптимизация региональных схем водоснабжения в районной планировке.— В кн.: Вопросы совершенствования градостроительства и районной планировки. Киев: Будивельник, 1980, с. 33—39.
15. Вычислительные методы выбора оптимальных проектных решений / В. С. Михалевич, Н. З. Шор, Л. А. Галустова и др. — Киев: Наукова думка, 1977.— 178 с.
16. Гантмахер Ф. Р. Теория матриц.— М.: Гостехиздат, 1953. — 492 с.
17. Гершович В. И. Об опыте решения задачи распределения углей на кокование.— В кн.: Тр. семинара Науч. совета АН УССР по кибернетике «Теория оптим. решений». Киев, ИК АН УССР, 1977, с. 22—35.
18. Гершович В. И. Об одном методе отсечений, использующем линейные преобразования пространства. — В кн.: Теория оптимальных решений. Киев: ИК АН УССР, 1979, с. 15—23.

19. Гершович В. И. Об одном методе минимизации, использующем линейные преобразования пространства.— В кн.: Теория оптимальных решений. Киев: ИК АН УССР, 1980, с. 38—45.
20. Гершович В. И. Об одном алгоритме эллипсоидов.— В кн.: О некоторых алгоритмах негладкой оптимизации и дискретного программирования. Препринт 82-6 ИК АН УССР, 1981, с. 8—13.
21. Гершович В. И. Квадратичное сглаживание в итеративных декомпозиционных алгоритмах решения задач линейного программирования большой размерности.— В кн.: II Республиканский симпозиум по методам решения нелинейных уравнений и задач оптимизации. Таллин, 1981, с. 188—190.
22. Гершович В. И. О скорости сходимости алгоритмов эллипсоидов минимизации выпуклых функций.— В кн.: Программное обеспечение экстремальных задач и пакеты прикладных программ. Киев: ИК АН УССР, 1982, с. 37—42.
23. Гольштейн Е. Г., Третьяков Н. В. Градиентный метод минимизации и алгоритмы выпуклого программирования, связанные с модифицированными функциями Лагранжа.— Экономика и мат. методы, 1975, 11, № 4, с. 730—742.
24. Гольштейн Е. Г., Юдин Д. Б. Новые направления в линейном программировании.— М.: Сов. радио, 1966.— 524 с.
25. Гольштейн Е. Г., Юдин Д. Б. Задачи линейного программирования транспортного типа.— М.: Наука, 1969.— 346 с.
26. Гэри М. Р., Джонсон Д. С. Вычислительные машины и трудноразрешимые задачи.— М.: Мир, 1982.— 416 с.
27. Демьянов В. Ф., Малоземов В. Н. Введение в минимакс.— М.: Наука, 1972.— 368 с.
28. Демьянов В. Ф., Васильев Л. В. Недифференцируемая оптимизация.— М.: Наука, 1981.— 384 с.
29. Демьянов В. Ф., Рубинов А. М. О квазидифференцируемых функциях.— ДАН СССР, 1980, т. 250, № 1, с. 21—25.
30. Еремин И. И. О методе «штрафов» в выпуклом программировании.— Кибернетика, 1967, № 4, с. 63—67.
31. Еремин И. И., Астафьев Н. Н. Введение в теорию линейного и выпуклого программирования.— М.: Наука, 1976.— 192 с.
32. Еремин И. И. Двойственность для несобственных задач линейного и выпуклого программирования.— Свердловск, 1984 (Препринт ИММ УНЦ СССР).— 43 с.
33. Ермольев Ю. М. Методы решения нелинейных экстремальных задач.— Кибернетика, 1966, № 4, с. 1—17.
34. Ермольев Ю. М., Шор Н. З. Метод случайного поиска для задач двухэтапного стохастического программирования и его обобщение.— Кибернетика, 1968, № 1, с. 90—92.
35. Ермольев Ю. М. Методы стохастического программирования.— М.: Наука, 1976.— 240 с.
36. Журбенко Н. Г., Пинаев Е. Г., Шор Н. З., Юнг Г. Н. Выбор состава и распределение гражданских самолетов по авиалиниям.— Кибернетика, 1976, № 4, с. 138—141.
37. Зойтендейк Г. Методы возможных направлений.— М.: ИЛ, 1963.— 175 с.
38. Иоффе А. Д., Тихомиров В. М. Теория экстремальных задач.— М.: Наука, 1974.— 480 с.
39. Канторович Л. В. О методе наискорейшего спуска.— ДАН СССР, 1947, 56, вып. 3, с. 233—236.
40. Канторович Л. В. О перемещении масс.— ДАН СССР, 1942, 37, № 7—8, с. 227—229.
41. Кристофидес Н. Теория графов: Алгоритмический подход.— М.: Мир, 1978.— 482 с.
42. Лебедев С. С. Целочисленное программирование и множители Лагранжа.— Экономика и мат. методы, 1974, 10, № 3, с. 592—610.
43. Лебедев С. С., Ковалевская М. И. Множители Лагранжа в простейшей задаче размещения.— В кн.: Исследования по дискретной оптимизации. М.: Наука, 1976, с. 170—180.

44. Левин А. Ю. Об одном алгоритме минимизации выпуклых функций.— ДАН СССР, 1965, 160, № 6, с. 1244—1247.
45. Михалевич В. С., Сергиенко И. В., Шор Н. З. и др. Пакет программ ДИСПРО-3: назначение, классы решаемых задач, системное и алгоритмическое обеспечение.— Кибернетика, 1985, №1, с. 56—71.
46. Михалевич В. С., Сергиенко И. В., Лебедева Т. Т. и др. Пакет прикладных программ ДИСПРО, предназначенный для решения задач дискретного программирования.— Кибернетика, 1981, № 3, с. 117—137.
47. Михалевич В. С., Сергиенко И. В., Трубин В. А. и др. Пакет прикладных программ для решения задач производственно-транспортного планирования большой размерности.— Кибернетика, 1983, № 3, с. 57—71, 79.
48. Михалевич В. С., Сергиенко И. В., Трубин В. А. и др. Пакет прикладных программ для решения задач производственно-транспортного планирования большой размерности.— В кн.: Пакеты прикладных программ: Методы оптимизации. М.: Наука, 1984, с. 66—84.
49. Муртаф Б. Современное линейное программирование. — М.: Мир, 1984.— 224 с.
50. Меренков А. П. Применение ЭВМ для оптимизации разветвленных тепловых сетей.— Изв. АН СССР. Энергетика и транспорт, 1963, № 3, с. 531—538.
51. Мухитдинов Ф. Т., Трубин В. А. Перспективное развитие и размещение винодельческой промышленности УзССР.— В кн.: Экономико-математическое моделирование развития отраслей и транспорта. Киев : ИК АН УССР, 1978, с. 18—31.
52. Немировский А. С., Юдин Д. Б. Сложность задач и эффективность методов оптимизации. — М.: Наука, 1979.— 383 с.
53. Нурминский Е. А. Численные методы решения детерминированных и стохастических минимаксных задач.— Киев: Наукова думка, 1979.— 159 с.
54. Нурминский Е. А. О непрерывности  $\epsilon$ -субградиентных отображений.— Кибернетика, 1977, № 5, с. 148—149.
55. Поляк Б. Т. Один общий метод решения экстремальных задач.— ДАН СССР, 1967, 174, № 1, с. 33—36.
56. Поляк Б. Т. Минимизация негладких функционалов. — Журн. вычисл. математики и мат. физики, 1969, 9, № 3, с. 509—521.
57. Поляк Б. Т. Введение в оптимизацию.— М.: Наука, 1983.— 384 с.
58. Прим Р. К. Кратчайшие связывающие сети и некоторые обобщения.— В кн.: Кибернетический сборник, вып. 2. М., ИЛ, 1961, с. 95—107.
59. П р и м а к М. Е. О сходимости модифицированного метода чебышевских центров решения задачи выпуклого программирования.— Кибернетика, 1977, № 5, с. 100—102.
60. Пшеничный Б. Н., Данилин Ю. М. Численные методы в экстремальных задачах.— М.: Наука, 1975.— 319 с.
61. Пшеничный Б. Н. Необходимые условия экстремума. — М.: Наука, 1969.— 151 с.
62. Рокафеллар Р. Выпуклый анализ. — М.: Мир, 1973.— 470 с.
63. Скоков В. А. Замечание к методам минимизации, использующим операцию растяжения пространства.— Кибернетика, 1974, № 4, с. 115—117.
64. Сумароков С. В., Храмов А. В. Вопросы оптимального синтеза систем водоснабжения с учетом надежности.— В кн.: Вопросы надежности систем водоснабжения. М.: МИСИ, 1978, с. 36—44.
65. Сухарев М. Т., Ставровский Е. Ф. Оптимизация систем транспорта газа.— М.: Недра, 1975.— 277 с.
66. Теоретическое и экспериментальное исследование некоторых комбинаторных задач / Трубин В. А., Гильбурд М. М., Дерюгин В. Н. и др.— Киев, 1984.— 46 с. (Препринт АН УССР, Ин-т кибернетики им. В. М. Глушкова: 84—14.)
67. Трубин В. А., Юн Г. Н. Синтез сети авиалиний в процессе выбора оптимального типа пассажирских самолетов.— Управляющие системы и машины, 1972, № 1, с. 55—59.
68. Трубин В. А. Свойства и методы решения задач оптимального синтеза сетей.— Киев: Об-во «Знание» УССР, 1982.— 23 с.



69. Грубин В. А. О некоторых задачах типа размещения.— В кн.: Применение математических методов в экономических исследованиях и планировании. Киев: ИК АН УССР, 1969, № 1, с. 3—15.
70. Грубин В. А., Чумаков Б. М. Задача унификации и размещения технических средств.— В кн.: Теория оптимальных решений. Киев: ИК АН УССР, 1979, с. 69—75.
71. Грубин В. А., Гндоян А. К. Алгоритм и свойства задачи синтеза сетей с одним источником.— В кн.: Теория оптимальных решений. Киев: ИК АН УССР, 1980, с. 81—87.
72. Грубин В. А., Два класса задач размещения на древовидных сетях.— Кибернетика, 1983, № 4, с. 84—87.
73. Грубин В. А. Эффективный алгоритм для задачи Вебера с прямоугольной метрикой.— Кибернетика, 1978, № 6, с. 67—70.
74. Федоров В. В. Численные методы максимина.— М.: Наука, 1979.
75. Фиакко А., Мак-Кормик Г. Нелинейное программирование: Методы последовательной безусловной минимизации.— М.: Мир, 1972.— 240 с.
76. Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры.— М.: Физматгиз, 1960.— 656 с.
77. Хачиян Л. Г. Полиномиальный алгоритм в линейном программировании.— ДАН СССР, 1979, 244, с. 1093—1096.
78. Хачиян Л. Г. Полиномиальный алгоритм в линейном программировании.— Журн. вычисл. математики и матем. физики, 1980, 20, № 1, с. 51—58.
79. Ху Т. Целочисленное программирование и потоки в сетях.— М.: Мир, 1974.— 519 с.
80. Черенин В. П. Составление оптимального плана формирования однопутных поездов на ЭВМ.— Вестн. ВНИИЖТ, 1961, № 1, с. 47—54.
81. Черенин В. П., Хачатуров В. Р. Решение методом последовательных расчетов одного класса задач о размещении производства.— В кн.: Экономико-математические методы. Вып. 2. М.: Наука, 1965, с. 279—290.
82. Шор Н. З. Применение метода градиентного спуска для решения сетевой транспортной задачи.— В кн.: Материалы науч. семинара по теорет. и прикл. вопр. кибернетики и исследования операций: Науч. совет по кибернетике АН УССР. Киев, 1962, вып. 1, с. 9—17.
83. Шор Н. З. Применение обобщенного градиентного спуска в блочном программировании.— Кибернетика, 1967, № 3, с. 53—55.
84. Шор Н. З., Росина Н. И. Схема разложения задач линейного и выпуклого программирования и ее применение для решения задач планирования перевозок.— В кн.: Докл. I Всесоюз. конф. по оптимизации и моделированию трансп. сетей. Киев, 1967, с. 225—237.
85. Шор Н. З., Горбач Г. И. Решение задач распределительного типа методом обобщенного градиентного спуска.— В кн.: Тр. семинара Науч. совета АН УССР по кибернетике «Теория оптимальных решений». Киев, № 1, с. 59—71.
86. Шор Н. З. О скорости сходимости обобщенного градиентного спуска.— Кибернетика, 1968, № 3, с. 98—99.
87. Шор Н. З. О скорости сходимости метода обобщенного градиентного спуска с растяжением пространства.— Кибернетика, 1970, № 2, с. 80—85.
88. Шор Н. З. Использование операций растяжения пространства в задачах минимизации выпуклых функций.— Кибернетика, 1970, № 1, с. 6—12.
89. Шор Н. З., Журбенко Н. Г. Метод минимизации, использующий операцию растяжения пространства в направлении скорости двух последовательных градиентов.— Кибернетика, 1971, № 3, с. 51—59.
90. Шор Н. З., Галустова Л. А., Момот А. И. Применение математических методов при оптимальном проектировании Единой газоснабжающей системы с учетом динамики ее развития.— Кибернетика, 1978, № 1, с. 69—74.
91. Шор Н. З., Шабашова Л. П. О решении минимаксных задач методом обобщенного градиентного спуска с растяжением пространства.— Кибернетика, 1972, № 1, с. 82—88.
92. Шор Н. З. О методе минимизации почти дифференцируемых функций.— Кибернетика, 1972, № 4, с. 65—70.

93. Шор Н. З. Метод отсечения с растяжением пространства для решения задач выпуклого программирования.— Кибернетика, 1977, № 1, с. 94—95.
94. Юдин Д. Б. Математические методы управления в условиях неполной информации.— М.: Сов. радио, 1974.— 400 с.
95. Юдин Д. Б., Немировский А. С. Информационная сложность и эффективные методы решения выпуклых экстремальных задач.— Экономика и мат. методы, 1976, вып. 2, с. 357—369.
96. Шор Н. З. Новые направления в развитии методов негладкой оптимизации.— Кибернетика, 1977, № 6, с. 87—91.
97. Шор Н. З. Методы минимизации недифференцируемых функций и их приложения.— Киев: Наукова думка, 1979.— 199 с.
98. Шор Н. З., Гершович В. И. Об одном семействе алгоритмов для решения задач выпуклого программирования.— Кибернетика, 1979, № 4, с. 62—67.
99. Шор Н. З., Гершович В. И. Об одной модификации алгоритмов минимизации градиентного типа с растяжением пространства для решения задач большой размерности.— Кибернетика, 1981, № 5, с. 67—70.
100. Юдин Д. Б., Гольштейн Е. Т. Линейное программирование: Теория, методы, приложения.— М.: Наука, 1969.— 699 с.
101. Agmon S. The relaxation method for linear inequalities.— *Canad. J. Math.*, 1954, 6, N 2, p. 382—392.
102. Akgül M. Topics in relaxation and ellipsoidal methods.— Pitman Publishing Inc., 1984.— 322 p.
103. Bachem A., Grötschel M., Korte B. (eds). *Mathematical Programming: The State of Art*, Bonn, 1982.— Springer-Verlag, 1983.— 655 p.
104. Balinski M. L., Wolfe P. (eds). *Nondifferentiable optimization*. Math. Programming Study, 3.— North-Holland, Amsterdam, 1975.— 178 p.
105. Balinski M. L., Gomory R. E. A primal method for the assignment and transportation problems.— *Management Sci.* 1964, 10, N 5, p. 578—594.
106. Barr R. S., Glover F., Klingman D. The alternating basis algorithm for assignment problems.— *Math. Progr.*, 1977, 13, N 1, p. 1—3.
107. Barton R. R. Nondifferentiable optimization methods for a traffic assignment decomposition model.— Report 81 — EM — 018, RCA Labs., Princeton, N. J., 1981.
108. Berge C. Balanced matrices.— *Math. Programming*, 1972, 2, N 1, p. 19—39.
109. Bertsekas D., Mitter S. A descent numerical method for optimization problems with nondifferentiable cost functionals.— *SIAM J. Control*, 1973, 11, p. 637—652.
110. Bertsekas D. Necessary and sufficient conditions for a penalty method to be exact.— *Math. Programming*, 1975, 9, N 1, p. 87—99.
111. Bertsekas D. Nondifferentiable optimization via approximation.— In [103], p. 1—25.
112. Best M. A. A method to accelerate the rate of convergence of a class of optimization algorithms.— *Math. Programming*, 1975, 9, N 2, p. 139—160.
113. Bilde O., Krarup J. Sharp lower bounds and efficient algorithms for the simple plant location problem.— *Annals of discrete mathematics*, 1. Studies in integer programming. North-Holland, 1977, p. 79—97. \*
114. Bland R. B., Goldfarb D., Todd M. J. The ellipsoid method: a survey.— *Operations Res.*, 1981, 29, N 5, p. 1039—1091.
115. Chandrasekaran. The weighted euclidean 1-center problem.— *Operations Res. Letters*, 1982, 1, N 2, p. 111—112.
116. Clarke F. H. Generalized gradients and applications.— *Transactions AMS*, 1975, 205, p. 247—262.
117. Cunningham W. H. Theoretical properties of the network simplex method.— *Mathematics of Operations Res.*, 1979, 4, N 2, p. 196—208.
118. Dantzig G., Wolfe P. The decomposition principles for linear programs.— *Oper. Res.*, 1960, 8, N 1, p. 101—111.
119. Dijkstra E. Two problems in connection with graphs.— *Numer. Math.*, 1959, 1, N 2, p. 262—271.

120. Ecker J. G., Kupferschmid A. B. An implementation and generalization of Shor's ellipsoid algorithm for convex programming.— *Math. Programming*, 1983, 27, N 1, p. 83—106.
121. Edmonds J. Paths, trees and flowers.— *Canad. J. Math.*, 1965, 17, p. 449—467.
122. Edmonds J. Optimum branchings.— *J. Res. Bur. Standards*, 1967, 71B, p. 233—240.
123. Edmonds J. Maximum matching and a polyhedron with 0—1 vertices.— *J. Res. Bur. Standards*, 1965, 69E, p. 125—130.
124. Edmonds J., Fulkerson D. R. Bottleneck extrema.— *J. Combinatorial Theory*, 1970, 8, N 8, p. 299—306.
125. Edmonds J. Matroids and the greedy algorithm.— *Math. Programming*, 1971, 1, N 1, p. 127—136.
126. Edmonds J., Karp R. M. Theoretical improvements in algorithmic efficiency for network flow problem.— *J. ACM*, 1972, 19, N 2, p. 248—264.
127. Edmonds J., Giles R. A mini-max relation for submodular functions on graphs.— *Annals of Discrete Mathematics*, 1977, 1, N 2, p. 185—204.
128. Edmonds J. Matroids intersection.— *Annals of Discrete Mathematics*, 1979, 4, N 1, p. 39—49.
129. Efromyson M. E., Ray T. L. A branch-bound algorithm for plant location.— *Operations Res.*, 1969, 17, N 1, p. 81—111.
130. Fisher M. L., Northup W. D., Shapiro J. F. Using duality to solve discrete optimization problems.— In [104], p. 56—94.
131. Fisher M. L. Lagrangian relaxation method for solving integer programming problems.— *Management Sci.*, 1981, 27, N 1, p. 1—18.
132. Fulkerson D. R. Blocking and antiblocking pairs of polyhedra.— *Math. Programming*, 1971, 1, N 2, p. 168—194.
133. Fulkerson D. R., Weinberger D. B. Blocking pairs of polyhedra arising from networks flows.— *J. Combinatorial Theory*, 1975, 18, N 2, p. 265—283.
134. Fulkerson D. R. Packing rooted directed cuts in a weighted directed graph.— *Math. Programming*, 1974, N 1, p. 1—13.
135. Gerstenhaber M. Solution of large scale transportation problems.— In: *Combinatorial Analysis* / Eds. R. Bellman, M. Hall. Amer. Math. Soc., 1960, p. 251—260.
136. Goffin J. L. On convergence rates of subgradient optimization methods.— *Math. Programming*, 1977, 13, p. 329—347.
137. Goffin J. L. Convergence rates of the ellipsoid method on general convex functions.— *Mathematics of Operations Res.*, 1983, 8, N 2, p. 235—250.
138. Goffin J. L. Variable metric relaxation methods. Part I: A conceptual algorithm.— *Technical Rep. SOL 81—16*, Stanford University, 1981.
139. Goldfarb D., Sit W. Y. Worst case behavior of the steepest edge simplex method.— *Discrete Applied Math.*, 1979, 1, N 3, p. 277—285.
140. Goldfarb D., Todd M. J. Modifications and implementation of the Shor—Khachian algorithm for linear programming.— *Math. Programming*, 1982, 23, N 1, p. 1—19.
141. Grötschel M., Lovász L., Shrijver A. The ellipsoid method and its consequences in combinatorial optimization.— *Combinatorica*, 1981, 1, N 2, p. 169—197.
142. Held M., Karp R. M. The travelling salesman problem and minimum spanning trees. Part II.— *Math. Programming*, 1971, 1, N 1, p. 6—25.
143. Held M., Wolfe P., Crowder H. Validation of subgradient optimization.— *Math. Programming*, 1974, 6, p. 62—88.
144. Karmarkar N. A new polynomial-time algorithm for linear programming.— *Combinatorica*, 1984, 4, N 4, p. 373—395.
145. Karp R. M. On the computational complexity of combinatorial problems.— *Networks*, 1975, 5, N 1, p. 45—68.
146. Kelley J. E. The cutting plane method for solving convex programs.— *SIAM J. Appl. Math.*, 1960, 8, N 4, p. 703—712.
147. Kiwiel K. C. Methods of descent for nondifferentiable optimization.— *Lecture Notes in mathematics*, Springer — Verlag, Berlin, 1985, 1133, 362 p.

148. Korte B., Schrader R. A note on convergence proofs of Shor—Khachian methods.— In: Lecture Notes on Control and Information Sciences, N 30, Springer, N. Y., 1980, p. 51—57.
149. Kruskal J. B. On the shortest spanning subtree of a graph and the travelling salesman problem.— Proc. Amer. Math. Soc., 1965, 7, N 1, p. 48—50.
150. Kuhn H. W. A note on Fermat's problem.— Math. Programming, 1973, 4, N 1, p. 98—107.
151. Lawler E. G., Marter C. U. Computing maximal polymatroidal network flows.— Math. of Operations Res., 1982, 7, N 3, p. 334—347.
152. Lawler E. G., Martel C. U. Flow network formulations of polymatroid optimization problems.— Annals of Discrete Mathematics, 1982, 16, N 2, p. 189—200.
153. Lemaréchal C. An algorithm for minimizing convex functions. In: Proceedings of IFIP Congress, North-Holland, Amsterdam, 1974, p. 552—556.
154. Lemaréchal C. An extension of Davidon methods to nondifferentiable problems. In [104], p. 95—100.
155. Lemaréchal C. Bundle methods in nonsmooth optimization.— In: Nonsmooth Optimization / Eds. C. Lemaréchal, R. Mifflin. Pergamon Press, Oxford, 1978, p. 79—102.
156. Lemaréchal C. Numerical experiments in nonsmooth optimization. In: Progress in nondifferentiable optimization / Ed. E. A. Nurminski. CP—82—88, International Institute for Applied System Analysis: Laxenburg, Austria, 1982, p. 61—84.
157. Lovász L. The matroid matching problem. In: Algebraic methods in graph theory / Eds. L. Lovász, V. Sós. North-Holland, 1981, p. 495—517.
158. Lovász L. Submodular functions and convexity.— In [103], p. 235—257.
159. Love R. F., Verex L. An application of a facilities location model in the pretreated concrete industry.— Interfaces, 1976, 6, N 1, p. 45—49.
160. Mangasarian O. L. Iterative solution of linear program.— SIAM J. Numer. Anal., 1981, 18, N 4, p. 606—614.
161. Mifflin R. Semismooth and semiconvex functions in constrained optimization.— SIAM J. Control Optim., 1977, 15, N 6, p. 959—972.
162. Minox M., Serreault J. Y. Subgradient optimization and large scale programming: an application to optimum multicommodity network synthesis with security constraints.— Operations Research, 1981, 15, N 2, p. 185—203.
163. Nemhauser J. L., Wolsey L. A., Fisher M. L. An analysis of approximations for maximizing submodular set functions. I.— Math. Programming, 1978, 14, N 3, p. 265—294.
164. Padberg M. W., Rao M. Odd minimum cut sets and b-matchings.— Mathematics of Operations Res., 1982, 7, N 1, p. 67—80.
165. Picard J. C., Ratliff H. D. A cut approach to the rectilinear distance facility location problem.— Operations Res., 1978, 26, N 3, p. 422—433.
166. Pulleyblank W. R. Polyhedral combinatorics.— In: [103], p. 312—345.
167. Rockafellar R. T. Generalized subgradients in mathematical programming.— In: [103], p. 368—390.
168. Röck H. Scaling techniques for minimal cost network flow.— In: Discrete structures and algorithms. Berlin. Hanser, 1980, p. 181—191.
169. Schrader R. Ellipsoid methods.— In: Modern applied mathematics / Ed. B. Korte. North-Holland, N. Y., 1982, p. 265—311.
170. Schrijver A. Mini-max results in combinatorial optimization.— In: [103], p. 439—500.
171. Shor N. Z. Generalized gradient methods of nondifferentiable optimization employing space dilatation operations.— In: [103], p. 501—529.
172. Smale S. The problem of the average speed of the simplex method.— In: [103], p. 530—539.
173. Tarjan R. E. Finding optimum branching.— Networks, 1977, 7, N 1, p. 25—35.
174. Tarjan R. E. Complexity of combinatorial algorithms.— SIAM Review, 1978, 20, N 4, p. 457—491.
175. Tind J. Blocking and antiblocking polyhedra.— Annals of discrete mathematics, 1979, 4, p. 159—174.

176. W e s o l o w s k y G . O . , L o v e R . F . The optimal location of new facilities using rectilinear distances.— *Operations Res.*, 1971, 19, N 1, p. 124—130.
177. W o l f e P . A method of conjugate subgradient for minimizing nondifferentiable functions.— In: [104], p. 145—173.
178. W o l f e P . A bibliography for the ellipsoid algorithm.— N. Y.: IBM Research Center, 1980.— 6 p.
179. Y a k i n M . Z . On the convergence rate of a space extension algorithm for non-differentiable optimization.— Univ. Houston, CBA, WP, 69, 1981.
180. Z a d e h N . A bad network problem for the simplex method and other minimum cost flow algorithms.— *Math. Programming*, 1973, 5, N 3, p. 255—266.
181. Z a d e h N . What is the worst case behaviour on the simplex?— Department of Operations Res., Stanford University, T. R. 27, 1980.

*Владимир Сергеевич Михалевич  
Владимир Алексеевич Трубин  
Наум Зуселевич Шор*

**ОПТИМИЗАЦИОННЫЕ ЗАДАЧИ  
ПРОИЗВОДСТВЕННО-ТРАНСПОРТНОГО ПЛАНИРОВАНИЯ**

Модели, методы, алгоритмы

Серия «Экономико-математическая библиотека»

---

Редактор *А. Д. Вайнштейн*  
Художественный редактор *Т. Н. Кольченко*  
Технический редактор *И. Ш. Аксельрод*  
Корректоры *Е. Ю. Рычагова, Т. С. Вайсберг*

ИБ № 12381

Сдано в набор 26.11.85. Подписано к печати 11.08.86. Т-16816. Формат 60×90<sup>1/16</sup>. Бумага тип. № 1. Гарнитура литературная. Печать высокая. Усл. печ. л. 16,5. Усл. кр.-отт. 16,5. Уч.-изд. л. 18,1. Тираж 4000 экз. Заказ № 5-904. Цена 3 р.

Ордена Трудового Красного Знамени издательство «Наука»  
Главная редакция физико-математической литературы  
117071 Москва В-71, Ленинский проспект, 15

Киевская книжная типография научной книги.  
252004 Киев 4, ул. Репина, 4

ИЗДАТЕЛЬСТВО «НАУКА»  
ГЛАВНАЯ РЕДАКЦИЯ  
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ  
117071 Москва В-71, Ленинский проспект, 15

---

ГОТОВИТСЯ К ПЕЧАТИ:

Михалевич В. С., Гупал А. М., Норкин В. И. **Методы невыпуклой оптимизации.**

В прикладных задачах часто приходится иметь дело с недифференцируемыми функциями и с невыпуклыми областями. Разработка методов оптимизации для таких задач представляет большую трудность. В книге рассмотрены конечномерные задачи невыпуклой негладкой оптимизации и численные методы их решения.

Для специалистов в области прикладной математики, информатики, экономики, кибернетики.

ИЗДАТЕЛЬСТВО «НАУКА»  
ГЛАВНАЯ РЕДАКЦИЯ  
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ  
117071 Москва В-71, Ленинский проспект, 15

---

**ГОТОВИТСЯ К ПЕЧАТИ:**

**Алексеев О. Г. Комплексное применение методов дискретной оптимизации.**

Содержит систематизированное изложение задач дискретного программирования. Основное содержание составляет комплексный подход использования методов динамического программирования и метода ветвей и границ. Рассматриваются сужение области поиска путем преобразования ограничений и применение двойственной задачи для оценки границы решения, а также метод динамического программирования со скользящей последовательностью. Приводится оценка эффективности предлагаемых методов по результатам вычислительных экспериментов на ЭВМ.

Для специалистов в области прикладной математики, кибернетики, физики, экономики, а также для студентов вузов.



ИЗДАТЕЛЬСТВО «НАУКА»  
ГЛАВНАЯ РЕДАКЦИЯ  
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ  
117071 Москва В-71, Ленинский проспект, 15

---

**ГОТОВИТСЯ К ПЕЧАТИ:**

**Червинский Р. А. Методы синтеза систем в целевых программах.**

Изучается одна из центральных и наиболее сложных проблем программно-целевого планирования — проблема синтеза сложных систем. Методы синтеза сложных систем рассматриваются как составная часть обоснования и реализации сложных систем. Затрагиваются вопросы планирования и обоснования решений при развитии системы, описаны методы принятия решений при непосредственном управлении реализацией целевой программы.

Для специалистов в области кибернетики, связанных с разработкой программ.